

Database Actions

Using Guide for Oracle Cloud



G31944-10
December 2025



Database Actions Using Guide for Oracle Cloud,

G31944-10

Copyright © 2020, 2025, Oracle and/or its affiliates.

Primary Author: Celin Cherian

Contributing Authors: Manisha Mati, Shirley Thomas

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	i
Documentation Accessibility	i
Product Accessibility	i
Related Documents	i
Conventions	ii
Third-Party License Information	ii

1 Changes in This Release

2 About Oracle Database Actions

Accessing Database Actions	1
Enabling User Access to Database Actions	2
About the Database Actions User Interface	2
Using the Omnisearch Bar	4
The Home Page - Launchpad	6
Enabling Detailed Request Error Messages for a Specific Schema	9

Part I Development

3 The SQL and Data Modeler Pages

The Overview Page	1
The SQL Page	2
Object Navigator and Files	3
Executing SQL Statements in the Code Editor	6
About Session State in Database Actions	9
Keyboard Shortcuts	9
Support for Multilingual Engine	12
Viewing the SQL Output	14
Using the Explain Plan Diagram	17

Loading Data	20
Loading Data from a Local File to a New Table	20
Loading Data from a Local File to an Existing Table	22
Loading Data from Multiple Local Files into Multiple Tables	25
Format Specifications for JSON and XML Files	27
The Data Modeler Page	32
Navigating Diagrams and Objects	32
About the Data Modeler Editor	34
About Viewing JSON Data Guides	36
Creating and Editing Database Objects	41
The Table Properties Dialog	42
Columns Pane	44
Primary Key Pane	47
Unique Keys Pane	47
Indexes Pane	48
Foreign Keys Pane	48
Table Constraints Pane	49
Comments Pane	49
Storage Pane	49
External Table Properties Pane	49
Materialized View Pane	52
DDL Pane	54
Output Pane	54
The Index Properties Dialog	54
The Sequence Properties Dialog	55
The View Properties Dialog	56
The Synonym Properties Dialog	57
The Materialized View Log Properties Dialog	58
Implied Foreign Keys	60

4 The REST Pages

About RESTful Web Services	1
RESTful Services Terminology	2
About the Overview Page	3
About the AutoREST Page	3
Enabling REST Access for a Database Object	4
Disabling REST Access for a Database Object	5
Generating cURL Requests for a REST-Enabled Database Object	5
About the REST Search Toolbar	5
Creating RESTful Web Services	6
Managing Resource Modules	7

Creating a Resource Module	8
Importing a Resource Module	9
Editing a Resource Module	9
Deleting a Resource Module	9
Publishing/Unpublishing a Resource Module	9
Exporting a Resource Module	10
Viewing the Module in OpenAPI View	10
Managing Resource Templates	10
Creating a Resource Template	11
Editing a Resource Template	12
Deleting a Resource Template	12
Managing Resource Handlers	12
Creating a Resource Handler	13
Editing a Resource Handler	14
Deleting a Resource Handler	14
Viewing Resource Handler Details and Managing Parameters	14
Using Pre-Authenticated Requests to Access ORDS RESTful Services	20
Example: Inserting a Record using a POST Handler	21
Securing RESTful Web Services	22
Managing Roles	23
Creating a Role	24
Editing a Role	24
Deleting a Role	24
Viewing Assigned Privileges	24
Managing Privileges	24
Creating a Privilege	25
Editing a Privilege	25
Deleting a Privilege	26
Managing OAuth Clients	26
Creating an OAuth Client	27
Editing an OAuth Client	28
Deleting an OAuth Client	28
Exporting an OAuth Client	28
Managing Secrets	29
Examples	29
Creating an OAuth Client Using the Client Credentials Grant Type	29
Creating an OAuth Client Using the Auth Code Grant Type	31

5 The JSON Page

About the JSON User Interface	2
Managing JSON Collections	4

Creating a Collection	4
About Adding or Editing a JSON Document	5
About Database Differences in JSON Documents	6
About Querying Documents in a Collection	7
Using the In-Context Autocomplete Feature in the JSON Editor	10
Creating Indexes for JSON Collections	11
Creating Relational Views of JSON Documents	13
Viewing the JSON Data Guide Diagram for a Collection	14

6 The Charts and Dashboards Page

Overview	1
Creating or Editing a Chart	2
Example: Creating and Editing a Chart	3
Creating or Editing a Dashboard	6

7 The Scheduling Pages

Overview	2
Jobs	2
Summary	3
Running	4
History	4
Forecast	5
History (Gantt Chart)	5
Notifications	6
Create or Edit Job	7
Chains	9
Create or Edit Chain	12
Add or Edit Step	12
Add or Edit Rule	13
Programs	13
Create or Edit Program	14
Schedules	15
Create or Edit Schedule	16
Objects	16
Create Job Class	18
Create File Watcher	18
Create Window	19
Create Window Group	20

8 The Liquibase Page

Generate a Deployment	1
About the Liquibase User Interface	2

Part II Monitoring

9 The Monitoring Pages

The Monitoring Overview Page	1
The Performance Hub Page	2
The Alerts Page	3
The Sessions Page	4
The Storage Page	4
The Parameters Page	5
The Database Dashboard Page	6
Database Dashboard Overview	6
Database Dashboard Activity	12
The AWR Page	13

Part III Data Tools

10 The Data Studio Overview Page

11 Python Data Studio API for Oracle Autonomous Database

Working with Python Data Studio API for Autonomous Database	1
Ingest Functions	3
Analytic View Functions	12
Insight Functions	21
Miscellaneous functions	25
Data Share Functions	32
Recipient Data Share Functions	38
Provider Data Share Functions	42
Catalog Functions	46

12 The Data Load Page

Data Studio Settings	4
Slack Integration with Data Studio	10

Create AI Profile	10
Managing Connections	15
Create Credentials	17
Create Oracle Cloud Infrastructure Native Credentials	28
Create Amazon Web Services (AWS) Credentials	30
Manage Resource Principal with DBMS_CLOUD	31
Register Data Catalog	34
Loading Data	37
Use OCI Language Service Capabilities in Data Studio	38
Overview of Sentiment Analysis, Key Phrase Extraction and Language Detection	39
Parameters to Analyze Data	41
Perform Sentiment Analysis	41
Perform Key Phrase Extraction	44
Perform Language Detection	47
Perform Text Translation	50
Use OCI Document Understanding to extract Tables from Images	53
Overview of Table Extraction	53
Perform Table Extraction from Image	54
Use GeoJSON in Data Load	57
Add Spatial Expression for GeoJSON columns	58
Parameters for Spatial Data Analysis	59
Perform Spatial Analysis	59
Output Data Generated from Spatial Analysis	62
Load Apache Iceberg Tables	62
Loading Data from File System	66
Create Directories in Database Actions	67
Loading Data from AI Source	69
Loading Data from Share	70
Create Live Feed from Data Load	73
Loading Data From Local Files	77
Loading Data from Other Databases	86
Load Data from Oracle and Non-Oracle Databases using Database Links	94
Loading Data from Cloud Storage	98
Manage Cloud Storage Links for Data Load Jobs	99
Prepare the Data Load Job	100
Add Files or Folders for the Data Load Job	100
Enter Details for the Data Load Job	101
Run the Data Load Job	112
View Details About the Data Load Job After It Is Run	112
View the Table Resulting from the Data Load Job	113
Linking Data	113
Linking to File System	114

Linking to Share	114
Linking to Data Catalog	114
Linking to Other Databases	114
Linking to Objects in Cloud Storage	116
Manage Cloud Storage Links for Data Link Jobs	117
Prepare the Data Link Job	117
Add Files or Folders for the Data Link Job	117
Enter Details for the Data Link Job	118
Run the Data Link Job	122
View Details About the Data Link Job After It Is Run	122
View the Table Resulting from the Data Link Job	123
Feeding Data	123
Creating a Notification-Based Live Table Feed	129
Creating a Notification-Based Live Table Feed using Amazon Simple Storage Service (S3)	132
Creating a Notification-Based Live Table Feed using Microsoft Azure	137
Send Email using Live Feed Tool	142
DBMS_LIVE_FEED Package Reference	144
Summary of DBMS_LIVE_FEED Subprograms	145

13 The Catalog Tool

About the Catalog Page	3
Browse and Search Catalogs	6
Sorting and Filtering Entities	7
Filter Entities	8
Display Entity Search	9
Catalog Settings	10
Manage Catalogs	12
Inline Data Load of Delta Share from the Catalog Tool	30
Query Remote Catalogs and Databases	33
Conceptual Diagram of Querying Remote Catalogs	34
Concepts related to querying remote catalogs and databases	35
About Querying Remote Catalogs and Databases	36
Examples: List the Mounted Catalogs	38
Examples: List the Tables in a Catalog, Including Their Owning Schemas	38
Examples: Query Data from a Table in a Catalog	38
Examples: Query a Table from a Delta Share and Analyze the View	39
Viewing Entity Details	40
Autonomous Database Entity Details	47
Registering Cloud Links to Access Data	50
Export Data to Cloud	51

Gathering Statistics for Tables	53
Editing Tables	53
Columns Pane	55
Primary Key Pane	56
Unique Keys Pane	56
Indexes Pane	57
Foreign Keys Pane	57
Table Constraints Pane	58
Comments Pane	58
Storage Pane	58
External Table Properties Pane	58
Materialized View Pane	60
DDL Pane	62
Output Pane	62
Manage Catalogs with DBMS_CATALOG	62
DBMS_CATALOG Package	71
Summary of DBMS_CATALOG	73
MOUNT_DB_LINK Procedure	77
MOUNT_SHARES Procedure	78
MOUNT_SHARE Procedure	79
MOUNT_VIRTUAL_CATALOG Procedure	80
MOUNT_DATA_CATALOG Procedure	81
MOUNT_ICEBERG Procedure	82
UNMOUNT Procedure	84
GET_CATALOG_PROPERTY Procedure	84
GET_CATALOG_PROPERTY_CLOB Procedure	85
UPDATE_CATALOG_PROPERTY Procedure	85
UPDATE_CATALOG_PROPERTY_CLOB Procedure	86
UPDATE_TABLE_PROPERTY_CLOB Procedure	86
GET_SCHEMA_PROPERTY Procedure	87
GET_SCHEMA_PROPERTY_CLOB Procedure	87
UPDATE_SCHEMA_PROPERTY Procedure	88
GET_TABLE_PROPERTY Procedure	88
GET_TABLE_PROPERTY_CLOB Procedure	89
UPDATE_TABLE_PROPERTY Procedure	89
UPDATE_TABLE_PROPERTY_CLOB Procedure	90
ADD_REQUIRED_CREDENTIAL Procedure	90
REMOVE_REQUIRED_CREDENTIAL Procedure	91
RENAME_REQUIRED_CREDENTIAL Procedure	91
UPDATE_REQUIRED_CREDENTIAL_PROPERTY Procedure	92
GET_REQUIRED_CREDENTIAL_PROPERTY Procedure	92
SET_LOCAL_CREDENTIAL Procedure	93

GET_LOCAL_CREDENTIAL_MAP (Overloaded) Procedure	93
CREATE_CLOUD_STORAGE_LINK Procedure	94
DROP_CLOUD_STORAGE_LINK Procedure	95
CREATE_EXTERNAL_TABLE Procedure	95
DROP_SCHEMA Procedure	96
DROP_TABLE Procedure	96
CREATE_SCHEMA Procedure	97
GET_TABLES Procedure	97
GET_SCHEMAS Procedure	98
GET_OBJECTS Procedure	99
GENERATE_TABLE_SELECT (Overloaded) Procedure	99
OPEN_TABLE_CURSOR Procedure	100
FLUSH_CATALOG_CACHE Procedure	101
PREFILL_CATALOG_CACHE Procedure	101
CREATE_SYNCHRONIZED_SCHEMAS Procedure	102
DROP_SYNCHRONIZED_SCHEMAS Procedure	102
UPDATE_SYNCHRONIZED_SCHEMA_PROPERTY Procedure	103
GRANT_TO_RECIPIENT Procedure	103
REVOKE_FROM_RECIPIENT Procedure	103

14 The Data Transforms Page

Access Oracle Data Transforms From Data Studio	2
Data Transforms Notes	3
Work with Connections	3
Supported Connection Types	6
Create Custom Connectors	126
Create a Data Transforms Connection for Remote Data Load	127
Create an Apache Iceberg Connection	128
Create a Delta Share Connection	131
Create an Oracle Business Intelligence Cloud Connector Connection	134
Create and use an Oracle Cloud Infrastructure Generative AI Connection	136
Create an Oracle Enterprise Resource Planning Cloud Connection	138
Create an Oracle Financials Cloud Connection	140
Create and Use an Oracle NetSuite Connection	141
Create an Oracle Object Storage Connection	150
Create a REST Server Connection	153
The Data Load Page	155
Create and Run Data Loads	158
Run a Data Load	160
Work with Data Entities	164
Import Data Entities	165

Create Data Entities	166
Create Data Entities within the Data Flow editor	167
View Statistics of Data Entities	169
Work with Projects	170
About Data Flows	171
Create a Data Flow	171
About Data Flow Editor	173
Supported Database Functions	174
Add Components	176
Use Text Embedding Vector in a Data Flow	177
Component Properties	178
Map Data Columns	179
Validate and Execute a Data Flow	179
Schedule Data Flows or Workflows	180
Monitor Status of Data Loads, Data Flows, and Workflows	182
Introduction to Workflows	182
Create a New Workflow	183
Define a Data Studio Data Load in a Work Flow	186
Create and Use Variables	187
Create a Variable	187
Use Variables in a Data Flow	188
Use Variables in a Workflow	189
Use Machine Learning (ML) Models	190
Create an ML Model Data Entity in the Data Flow editor	190
ML Model Data Entity Properties	192
Use ML Model in a Data Flow	193
Create and Manage Jobs	193
Export and Import Objects	195
Export Objects	195
Import Objects	197
Reference	198
Enable Access to Private Data Sources from Autonomous Database	198
Troubleshoot Mismatch with Server Cert DN Error	199
Increase the Memory of ODI Agent	200

15 The Data Insights Page

About Insights	1
Generate Insights and View Reports	2

16 The Data Analysis Tool

Searching and obtaining information about Analytic Views	5
Creating Analytic Views	6
Working with Analyses	17
Viewing Analyses	17
Workflow to build Analyses	17
Format Numeric Columns in the Data Analysis Tool	17
Creating Analyses	18
Saving Analyses	19
Creating Reports	19
Working with Reports	23
Creating Reports on a Query	28
Creating Reports on an Analytic View	29
Run Natural Language Query in the Data Analysis Tool	30
Generate SQL Queries From Natural Language on Analytic Views	31
Generate SQL Queries from Natural Language on Tables	32
Generate Natural Language explanation from SQL Query using AI Explain	35
Vector Search in the Data Analysis Tool	37
Using Calculation Templates	42
Cumulative Aggregates	43
Prior and Future Period	44
Period to Date	45
Parallel Period	46
Moving Aggregates	46
Share	47
Rank	48
Access the Spreadsheet Add-in by OCI IAM Domains	49
Role Based Access Control IAM Connections	62
Oracle Autonomous Database for Excel	78
Install the add-in from Microsoft AppSource	80
Install the add-in from Database Actions	83
Install the add-in on Mac	83
Install the add-in on Windows	84
Uninstall the add-in	86
Using Oracle Autonomous Database for Excel	88
Connection management	89
Import a Connection	90
Add a connection	92
Share a connection	92
Log Settings	93
Run Direct SQL queries in an Excel worksheet	94

Read and Access Data Using Table Hyperlinks in Excel Sheet	97
Reporting and Analysis in Excel	101
Query an Analytic View in an Excel worksheet	106
Data Analysis in Excel Sheet	110
Natural Language in Excel Sheet	117
FAQs for Troubleshooting errors with Excel Add-in	120
Oracle Autonomous Database for Excel Privacy Policy Details	124
Oracle Autonomous Database for Google Sheets	125
Download Connection File	126
Install the add-on from Google Workspace Marketplace	127
Connecting to Autonomous Database	130
Generate Client ID and Client Secret using UI	131
Authorize Google Sheets to use Autonomous Database	138
Natural Language in Google Sheets	140
Data Analysis in Google Sheets	146
Run Direct SQL Queries	151
Read and Access Data Using Table Hyperlinks in Google Sheets	154
About Autonomous Database menu	157
Share or Publish	157
Oracle Autonomous Database for Google Sheets Terms of Service	157
Oracle Autonomous Database for Google Sheets Privacy Policy Details	158
Oracle Autonomous Database for Google Sheets Support	159

17 The Data Marketplace Tool

Access Data Marketplace Tool	2
Publish Listings to Marketplace	3
Consume Dataset	7
DBMS_DATAMART Package Reference	10
Summary of DBMS_DATAMART Subprograms	11

18 The Table AI Assist Tool

Prepare Data Using Table AI Assist Tool	1
Add a Column	12
Replace a Column	12
Remove a Column	13
Rename a Column	13
View Statistical Details about the Table you Load after you Create a Recipe	13
View Table Entity Details	14
Conversational AI for the Table AI Assist Tool	14

19 The Jobs Feature

How to Access the Jobs Feature?	1
Conceptual Diagram of the Jobs Feature	2
Create a Job	7
Manage and Monitor the Job	16
DBMS_DATA_TOOLS_JOBS Package Reference	17
Summary of DBMS_DATA_TOOLS_JOB Subprograms	18

20 The Data Share Tool

Overview of the Data Share Tool	1
Access and Enable the Data Share Tool	4
Provide Share	6
Provide Share Overview	7
Provide Versioned Share	11
Provide Live Share	15
View Share Entity Details	18
Create Share Recipient	19
View Share Recipient Entity Details	21
Consume Share	23
Consume Share Overview	23
Consume Versioned Share	25
Consume Live Share	30
View Share Provider Entity Details	31
Manage Shares with DBMS_SHARE	32
Data Share Limitations on Autonomous Database	32
Live Share Feature Slow Performance	33
FAQs on the Data Share Tool	41

Part IV Administration

21 The Database Users Page

Creating or Editing a User	2
Enabling REST	3
Deleting a User	4

22 The APEX Workspaces Page

Create a Workspace	1
--------------------	---

23 The Data Pump Page

Importing Data using Oracle Data Pump	2
Requirements	2
Importing Data	8

A Supported SQL*Plus and SQLcl Commands in SQL Worksheet

Supported SQL*Plus Commands	A-1
Supported SQLcl Commands	A-2
SODA Commands	A-3

Preface

This document provides information about Database Actions, a web-based interface that provides development and administration features for Oracle Database.

- [Audience](#)
- [Documentation Accessibility](#)
- [Product Accessibility](#)
- [Related Documents](#)
- [Conventions](#)
- [Third-Party License Information](#)

Audience

This online help is intended for those using Database Actions in Oracle Autonomous Database.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Product Accessibility

Database Actions provides features to support accessibility. See *Oracle Database Actions Accessibility Guide*.

Related Documents

Oracle REST Data Services Installation and Configuration Guide

Oracle REST Data Services Developer's Guide

[Oracle Autonomous Database](#)

Oracle SQL Developer User's Guide

To download release notes, installation documentation, white papers, or other collateral for SQL Developer, go to the Oracle Technology Network (OTN) at

<http://www.oracle.com/technetwork/>

For the PL/SQL page on OTN, see <http://www.oracle.com/technetwork/database/features/plsql/>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Third-Party License Information

See Third-Party License Information in *Oracle REST Data Services Developer's Guide*

1

Changes in This Release

This section lists the changes in this release.

The **Create AI Profile** wizard enables you to connect your Autonomous AI Database to AI providers and select models and credentials. It also supports advanced options like Retrieval Augmented Generation (RAG), vector indexes, and table metadata so you can securely manage how AI accesses your tables.

See [Create AI Profile](#) for more information.

2

About Oracle Database Actions

Oracle Database Actions is a web-based interface that uses Oracle REST Data Services to provide development, data studio, administration and monitoring features for Oracle Autonomous Database.

The main features include executing your SQL statements and scripts, creating Data Modeler diagrams, developing RESTful web services, managing JSON collections, and using the Data Load, Catalog, Data Insights, Business Models, and Data Transforms tools to load data from local and remote sources, view data in your tables and views, view objects in your data dictionary, and organize, analyze, and transform your data.

📘 Note

- Some features are only available if you sign in as a user with database administration rights. For such features, the following statement appears at the start of the feature description:
Available only if you signed in as a database user with administrator rights.
- Database Actions cannot be launched from a standby database in an Autonomous Data Guard configuration.

Database Actions is also available for download and deployment in your own on-premises Oracle Database or in customer-managed Oracle Database cloud services. For more information, see [Oracle Database Actions for On-Premises Oracle Database](#).

- [Accessing Database Actions](#)
- [About the Database Actions User Interface](#)
This section describes the Database Actions user interface.
- [The Home Page - Launchpad](#)
- [Enabling Detailed Request Error Messages for a Specific Schema](#)

📘 See Also

- [Creating or Editing a User](#) for creating users and assigning roles.

Accessing Database Actions

Database Actions runs in Oracle REST Data Services and access to it is provided through schema-based authentication. To access Database Actions, you must sign in as a database user whose schema has been enabled for Database Actions.

In Oracle Autonomous Database databases, the ADMIN user is pre-enabled. To enable another database user's schema, see [Enabling User Access to Database Actions](#).

- [Enabling User Access to Database Actions](#)

Enabling User Access to Database Actions

To enable a database user to access Database Actions, run the following code as the ADMIN user:

```
BEGIN
  ords_admin.enable_schema(
    p_enabled => TRUE,
    p_schema => 'schema-name',
    p_url_mapping_type => 'BASE_PATH',
    p_url_mapping_pattern => 'schema-alias',
    p_auto_rest_auth => NULL
  );
  commit;
END;
```

where:

- **schema-name** is the database schema name in all-uppercase.
- **schema-alias** is an alias for the schema name that will appear in the URL the user will use to access Database Actions. Oracle recommends that you do not use the schema name itself as a security measure to keep the schema name from being exposed.

After enabling user access, in the Autonomous Database Details page, click **Database Actions**. The Database Actions Launchpad page appears.

About the Database Actions User Interface

This section describes the Database Actions user interface.

The Database Actions user interface has three components:

- The [Header](#) at the top
- The page body, whose content varies depending on which page you are viewing
- The [Status Bar](#) at the bottom

Header

The header contains the Selector icon, a Search field, the help icon, and the user drop-down list.



- **Selector Icon**

Click the Selector icon  to see the main navigation menu slide into view. Click **Oracle Database Actions** in the header to go to the Launchpad page.

- **Search field**

To enter a search term, click in the Search field or use the shortcut key **Ctrl+K** (**Command+K** for Apple computers). For more information, see [Using the Omnisearch Bar](#).

- **Help Icon**

Click the help icon to open the contextual or online help for the page you are viewing.

- **User Drop-Down List**

The user drop-down list shows the database user you are signed in as, and provides the following items when you open it:

- **Preferences**

The options are:

General

- * **Language:** Select one of the following languages for the user interface: English, German, Spanish, French, Italian, Japanese, Korean, Portuguese, and Chinese
- * **Timezone:** Select **UTC** or **Local time zone** from the drop-down list.
- * **Theme:** Options are Dark, Light or Same as browser. Select **Dark** for changing the display to a dark background (Dark mode).

Code Editor

Provides global settings that apply to the code editor, such as theme, indentation, font family and line numbers.

Worksheet

- * **SQL History:** Controls whether the history of commands executed in the code editor is enabled in the browser or not.

Scripts

Provides options to set the initial state of commands.

- **Log:** Opens a dialog that shows the list of HTTP calls made during your session.
- **About:** Opens a dialog providing version information for the database and other components as well as copyright and licensing information.
- **Sign Out:** Signs you out of your database session.

Status Bar

The status bar contains icons that link to log files. The three icons (Errors, Warnings, Processes) are filters that have been applied to the log file.



Errors, Warnings: Displays an Errors or Warnings dialog, which lists log entries from unsuccessful REST calls or from any other problem in the application.

Processes: Displays a Processes dialog, which logs REST calls that are either finished or ongoing.

Log notification link: Displays a Log dialog, containing log entries of the following types: Errors, Warnings, Processes, SQL History and SQL Result.

- [Using the Omnisearch Bar](#)

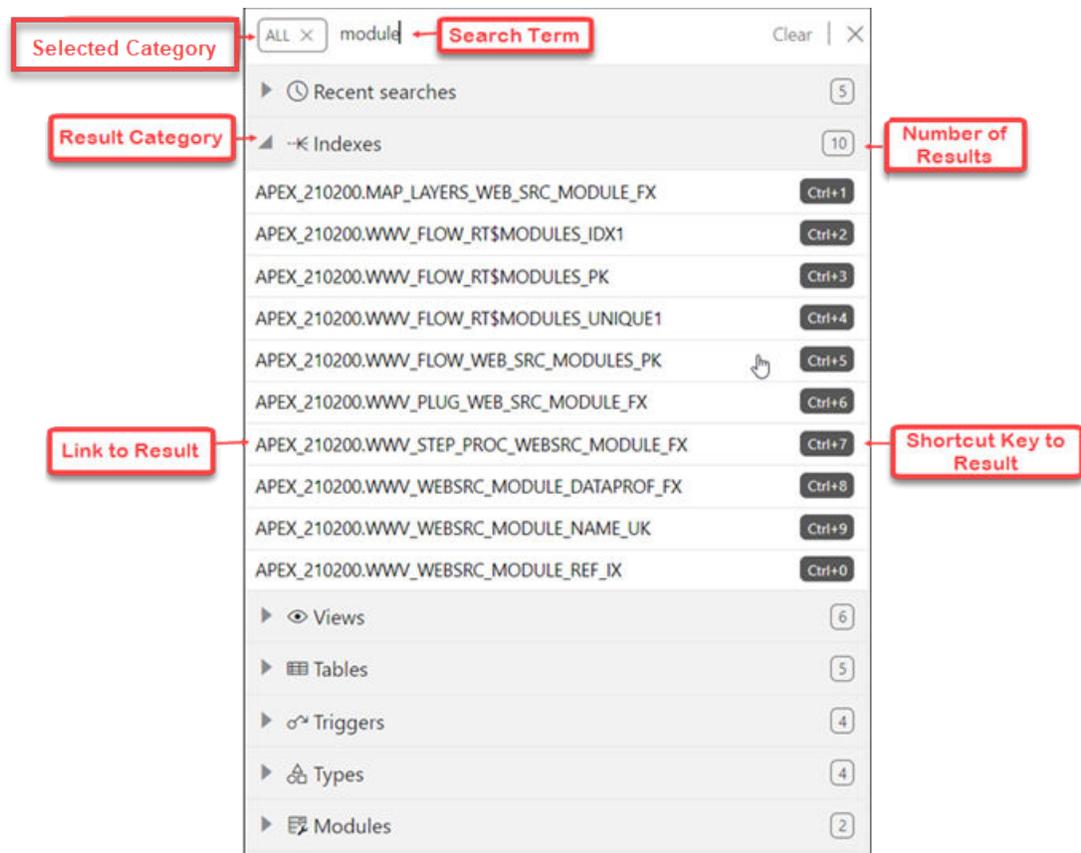
You can access the Omnisearch bar (Search field) in the header from any page in Database Actions.

Using the Omnisearch Bar

You can access the Omnisearch bar (Search field) in the header from any page in Database Actions.

To enter a search term, click in the search field located at the top right of the header, or use the shortcut key **Ctrl+K** (**Command+K** for Mac OS systems).

In the Omnisearch bar, you can filter the search entry by selecting a category (such as tables, views, templates, and so on) from the displayed list.



If you do select a category, you will then need to enter the exact search term.

If you do not select a category, **ALL** is selected by default. After you enter the search term, a search is performed across all categories. In some cases, the search term is searched across multiple attributes of a category. The following table lists the attributes that are searched for each category:

Category	Attribute
Tables	Name
Views	Name
Indexes	Name
Packages	Name

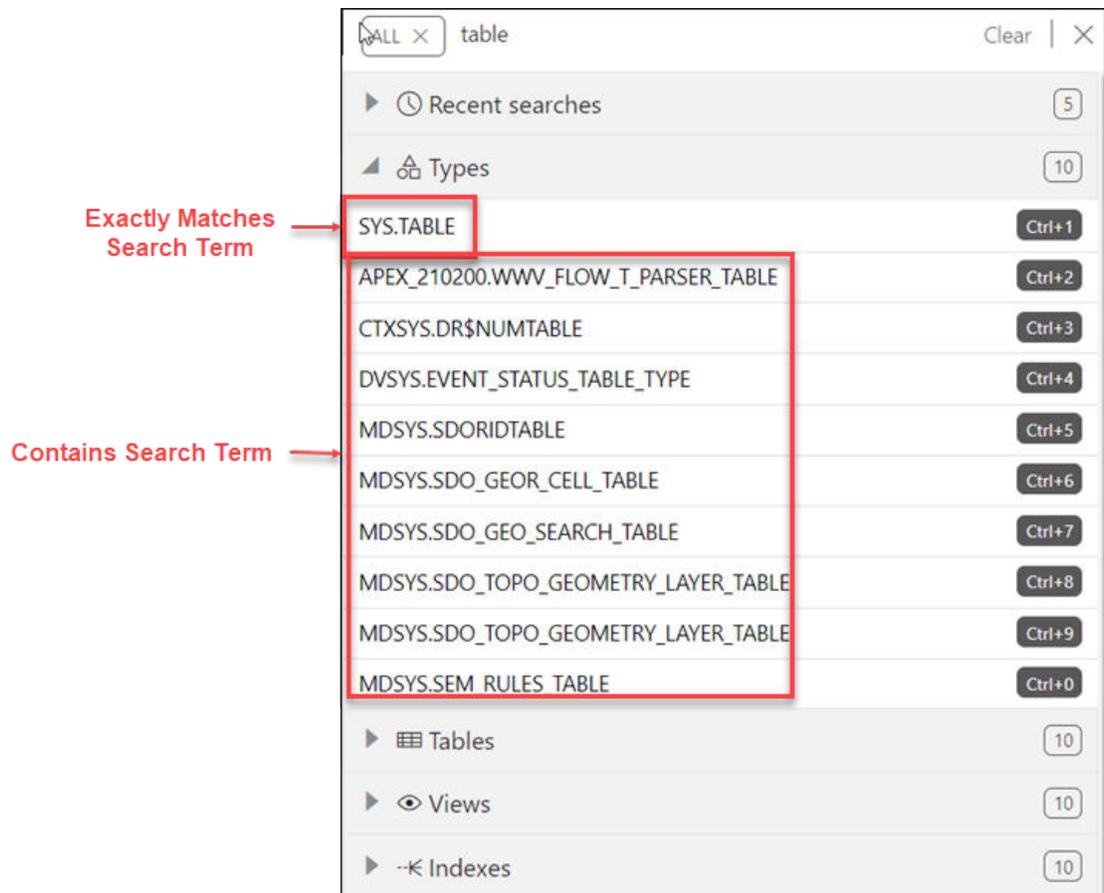
Category	Attribute
Functions	Name
Procedures	Name
Triggers	Name
Types	Name
Sequences	Name
Charts	Name, Comments, URI_Prefix
Dashboards	Name, Comments, URI_Prefix
Modules	Name, Comments, URI_Prefix
Templates	Comments, URI_Prefix
Handlers	Comments
Roles	Name
Privileges	Label, Name, Comments, Description
OAuth Clients	Name, Description
Database Users	Username, Alias (alias is used depending on user permissions)
APEX Workspaces	Workspace, Workspace display name

Displaying Results for the Search Term

In the results displayed, categories are sorted based on the following criteria:

1. Number of exact matches
2. Most number of matched items

The results within a category are displayed as two groups. The first group consists of items that exactly match the search term. The second group consists of items that contain the search term but do not start with it. Within each group, the items are sorted alphabetically.



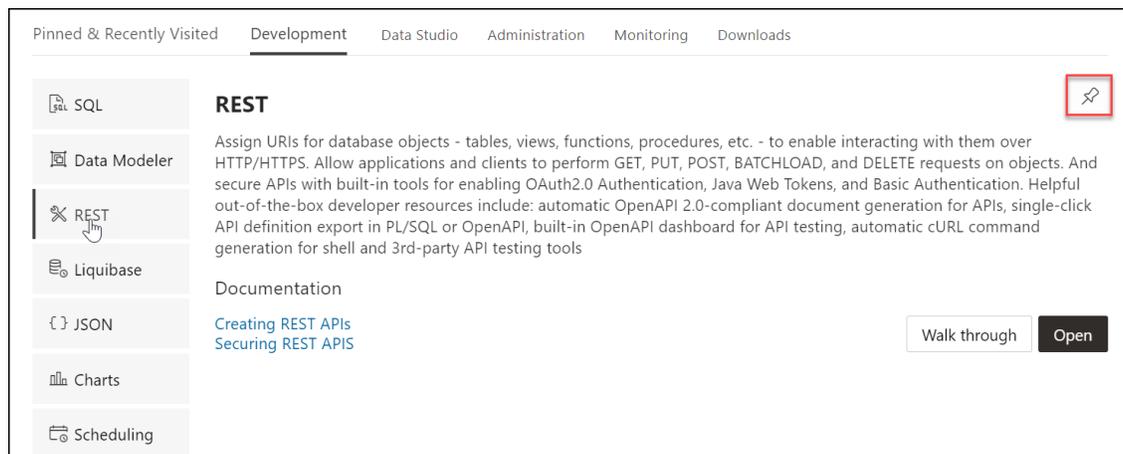
You can quickly access previous search terms using the **Recent Searches** list.

The Home Page - Launchpad

The Launchpad page for Database Actions can now dynamically display categories and features based on whether you are logged in to an on-premises, Autonomous Database or OCI environment. For on-premises, the main categories displayed are Pinned and Recently Visited, Development, Administration and Monitoring. For Autonomous Database, additional categories displayed are Data Studio and Downloads. Descriptions for features belonging to these categories are provided below.

The first time you log in, the Development tab is displayed by default. All features that pertain to the Development category are displayed vertically on the left side of the screen. When you hover over a feature name, its description is displayed on the right along with links to specific documentation pages. A new feature is highlighted with a blue label.

In Pinned & Recently Visited, Pinned lists all features that you have pinned (saved). To pin a feature, click the Pin icon displayed on the top right of the feature description. There is no limit on the number of features that can be pinned. **Recently Visited** lists the last seven features that you have accessed.



To navigate to a specific feature such as the REST pages, click **REST** on the left or click **Open** in the feature description part.

The following is a description of each feature displayed on the home page.

Development

- **SQL:** Enter and execute SQL and PL/SQL commands, and create database objects. See [The SQL Page](#)
- **Data Modeler:** Create diagrams from existing database schemas, generate DDL statements, and create reports. See [The Data Modeler Page](#)
- **APEX:** Link to the Oracle Application Express sign-in page. Application Express is a rapid web application development platform for the Oracle database. See Oracle Application Express documentation
- **REST:** Develop RESTful web services and ensure secure access. See [The REST Pages](#)
- **Liquibase:** View changelogs for your schema. See [The Liquibase Page](#)
- **Charts:** Create charts and dashboards containing multiple charts using SQL queries. See [The Charts and Dashboards Page](#)
- **JSON:** Manage and query JSON collections. JSON is available only if you are signed in as a database user with the SODA_APP role. See [The JSON Page](#)
- **Scheduling:** Provide details of scheduled jobs, chains, programs and schedules. See [The Scheduling Pages](#)
- **Oracle Machine Learning:** Link to the Oracle Machine Learning sign-in page. See Creating Dashboards, Reports and Notebooks
- **Graph Studio:** Link to the Autonomous Database sign-in page. See [Using Oracle Graph with Autonomous Database](#)

Data Studio

- **Data Load:** Load or access data from local files or remote databases. See [The Data Load Page](#).
- **Catalog:** Understand data dependencies and the impact of changes. See [The Catalog Tool](#).
- **Data Transforms:** Design graphical data transformations for data integration. See [The Data Transforms Page](#).

- **Data Analysis:** Creates Analytic Views and analyze data with add-ins. See [The Data Analysis Tool](#).
- **Data Insights:** Discover anomalies, outliers, and hidden patterns in your data. See [The Data Insights Page](#).
- **Data Share:** Provides Data Shares and Consumes Data Shares with other databases and non-database tools. See [The Data Share Tool](#).

Administration

Administration is available only if you are signed in as a database user with administrator rights.

- **Database Users:** Perform user management tasks such as create, edit, and REST enable users. See [The Database Users Page](#)
- **Data Pump:** Monitor Data Pump jobs initiated through the available Database API endpoints, the DBMS_DATAPUMP package, or the SQL Developer Data Pump Export and Import wizards. See [The Data Pump Page](#)
- **APEX Workspaces:** Create and manage APEX workspaces. See [Creating Applications with Oracle APEX](#)
- **Download Client Credentials:** Download and use the wallet file to securely connect your existing tools and applications to Autonomous Database. See [Download Database Connection Information](#)
- **Set Resource Management Rules:** Set resource management rules to allocate CPU/IO shares to consumer groups and to cancel SQL statements based on their runtime and amount of IO. See [Manage Runaway SQL Statements on Autonomous Database and Manage CPU/IO Shares on Autonomous Database](#)

Monitoring

Monitoring is available only if you are signed in as a database user with administrator rights.

- Monitor database activity and performance using various tools. See [The Monitoring Pages](#)
- **Performance Hub:** Shows the performance data for the specified time period. See [The Performance Hub Page](#)
- **Database Monitor:** Provides information about the performance of an Autonomous Database instance. See [The Database Dashboard Page](#)

Downloads

- **Download Oracle Instant Client:** Link to the Oracle Instant Client page on OTN. See [Import Data Using Oracle Data Pump on Autonomous Database](#)
- **Download SODA Drivers:** Link to the Oracle JSON Document Database page on OTN. See [Simple Oracle Document Access \(SODA\)](#)
- **Download Microsoft Excel/Google Sheets Add-In:** Opens a Download screen with Microsoft Excel and Google Sheets tabs. Click the Download icon in the Microsoft Excel tab to download the Oracle Autonomous Database Add-in for Excel. Click the Download icon in the Google Sheets tab to download the Oracle Autonomous Database Add-in for Google Sheets. The individual tabs consists of instructions to install the respective add-ins for the database. See [Oracle Autonomous Database Add-in for Excel](#) and [Oracle Autonomous Database add-on for Google Sheets](#) for more details on this.

Related Services

- **RESTful Services and SODA:** Oracle REST Data Services (ORDS) provides HTTPS interfaces for working with the contents of your Oracle Database in one or more REST enabled schemas.. See [Developing RESTful Services in Autonomous Database](#)
- **Oracle Machine Learning RESTful Services:** Oracle Machine Learning provides REST APIs for OML4Py Embedded Python Execution and OML Services.
- **Oracle Database API for MongoDB:** Oracle Database API for MongoDB enables MongoDB compatible clients and drivers to connect directly to Autonomous Database. See [Using Oracle Database API for MongoDB](#)

Enabling Detailed Request Error Messages for a Specific Schema

For development or debugging purposes, you can enable detailed request error messages to appear for a specific schema in an Autonomous Database.

To set this up, run the following code block:

Note

You need the `ORDS_ADMINISTRATOR_ROLE` to run this code block.

```
begin
ords_admin.set_property(
  p_schema => 'HR',
  p_key => 'debug.printDebugToScreen',
  p_value => 'true'
);
end;
/
```

Subsequently, any user request that produces an error response will include a detailed message, including a stack trace.

After debugging, turn the schema-level configuration off by setting the parameter to `false`.

Note

This setting must not be enabled on production systems due to the risk of sensitive information being revealed to an attacker.

Part I

Development

This part provides information about the following topics:

Topics:

- [The SQL and Data Modeler Pages](#)
- [The REST Pages](#)
- [The Liquibase Page](#)
- [The Charts and Dashboards Page](#)
- [The JSON Page](#)
- [The Scheduling Pages](#)
- [The SQL and Data Modeler Pages](#)
Use the SQL page to enter and execute SQL and PL/SQL statements and create database objects. The Data Modeler page enables you to create diagrams from existing schemas, retrieve data dictionary information, generate DDL statements, and export diagrams.
- [The REST Pages](#)
The REST pages enable you to develop and protect ORDS (Oracle REST Data Services) based RESTful web services for your database.
- [The JSON Page](#)
- [The Charts and Dashboards Page](#)
Charts enable you to create charts from the database. The chart is constructed using the input SQL command. Dashboards enable you to group charts together to create reports.
- [The Scheduling Pages](#)
- [The Liquibase Page](#)
The Liquibase page displays information of all database deployments made in the current schema. A deployment consists of changesets, which is a list of sequential changes to be applied to the database.

3

The SQL and Data Modeler Pages

Use the SQL page to enter and execute SQL and PL/SQL statements and create database objects. The Data Modeler page enables you to create diagrams from existing schemas, retrieve data dictionary information, generate DDL statements, and export diagrams.

Topics

- [The Overview Page](#)
- [The SQL Page](#)
- [The Data Modeler Page](#)
- [Creating and Editing Database Objects](#)
- [The Overview Page](#)
The Overview page contains widgets that provide a general overview of the activity in the SQL and Data Modeler pages.
- [The SQL Page](#)
The SQL page enables you to enter and execute SQL and PL/SQL statements, and create database objects.
- [The Data Modeler Page](#)
The Data Modeler page provides an integrated version of Oracle SQL Developer Data Modeler with basic reporting features. You can create diagrams from existing schemas, retrieve data dictionary information, generate DDL statements, and export diagrams.
- [Creating and Editing Database Objects](#)
You can create and edit objects using Create and Edit Object wizards available from the Navigator tab in the SQL and Data Modeler pages.

The Overview Page

The Overview page contains widgets that provide a general overview of the activity in the SQL and Data Modeler pages.

To navigate to the Overview page, click **Selector**  and then under Development, select **Overview**.

- **My Worksheets:** Displays your saved worksheets. You can click the name of the worksheet to open it in the Worksheet page.
- **My Diagrams:** Displays the Data Modeler diagrams that have been saved. You can click the name of the diagram to open it in the Data Modeler page.
- **Recently Modified Objects:** Displays a timeline of the created, modified, and dropped objects in the database. You can zoom in and out using the + and – icons. You can also move horizontally by dragging the cursor to the right or left.
- **Invalid Objects:** Displays the invalid objects in your schema.
- **Table Stats Freshness:** Displays the time period since the tables were last analyzed.

You can right-click the header in Invalid Objects, Table Stats Freshness, My Worksheets, or My Diagrams to manage or sort columns:

- **Columns:** Enables you to select columns to show or hide.
- **Sort:** Displays a dialog box for selecting columns to sort by. For each column, you can specify ascending or descending order, and you can specify that null values be displayed first.

Right-click the body of the display table to count rows or to view records:

- **Count Rows:** Displays the number of rows in the table.
- **Single Record View:** Enables you to view data for a table or view, one record at a time.
- **Copy:** Copies data from a cell or a row or a range of rows. To copy from more than one row, select the rows you want to copy, right-click by pressing the SHIFT or CTRL key, and select **Copy**.

The SQL Page

The SQL page enables you to enter and execute SQL and PL/SQL statements, and create database objects.

To navigate to the SQL page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **SQL**.
- Click **Selector**  to display the navigation menu. Under Development, select **SQL**.

You can use SQL and PL/SQL statements in the worksheet to create a table, insert data, create and edit a trigger, select data from a table, and save that data to a file. Some other features are syntax highlighting and error detection.

The SQL page consists of the left pane for navigating worksheets and objects, the editor for executing SQL statements, and the output pane for viewing the results. These panes are described in the following sections:

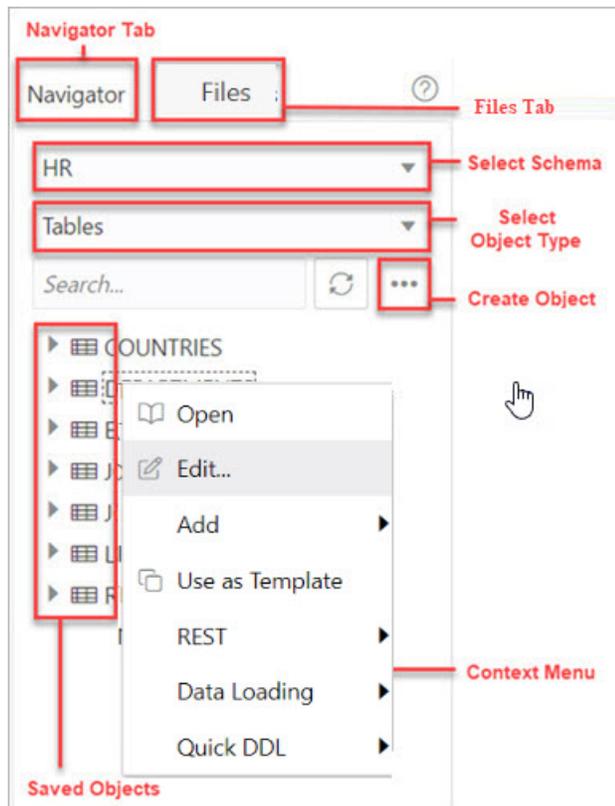
- [Object Navigator and Files](#)
- [Executing SQL Statements in the Code Editor](#)
- [Viewing the SQL Output](#)
- [Loading Data](#)
- [Object Navigator and Files](#)
The Navigator tab in the left pane displays saved objects for the selected schema. The Files tab enables you to view and open files saved in your browser or local system.
- [Executing SQL Statements in the Code Editor](#)
The code editor in the SQL page enables you to run SQL statements, PL/SQL scripts, and JavaScript code. The main features include in-context code completion, syntax highlighting, and error debugging.
- [Viewing the SQL Output](#)
The lower right pane in SQL displays the output of the operation executed in the SQL editor.
- [Loading Data](#)
In the SQL page, you can load data from one or more local files into one or more tables.

Object Navigator and Files

The Navigator tab in the left pane displays saved objects for the selected schema. The Files tab enables you to view and open files saved in your browser or local system.

The following figure shows the various elements in the left pane.

Figure 3-1 Left Pane in SQL



Navigator Tab

Displays saved objects for the selected schema.

- **Schema and Object Type selector:** Use the drop-down lists to select the schema and filter the results by object type.
- **Search:** Searches the contents of a saved worksheet or search for objects in the Navigator tab by name. The search functionality is not case-sensitive, retrieves all matching entries, and does not require the use of wild card characters.
- **Context menu:** Options in the context menu are:

- **Open** to browse properties and data relevant to the object type (tables and views).

The Data pane displays the data for a table, view, or materialized view.

To edit an entry, double-click a cell to make edits. You can also click  and enter the value. When you make an edit, the border of the gutter cell in that row changes to blue.

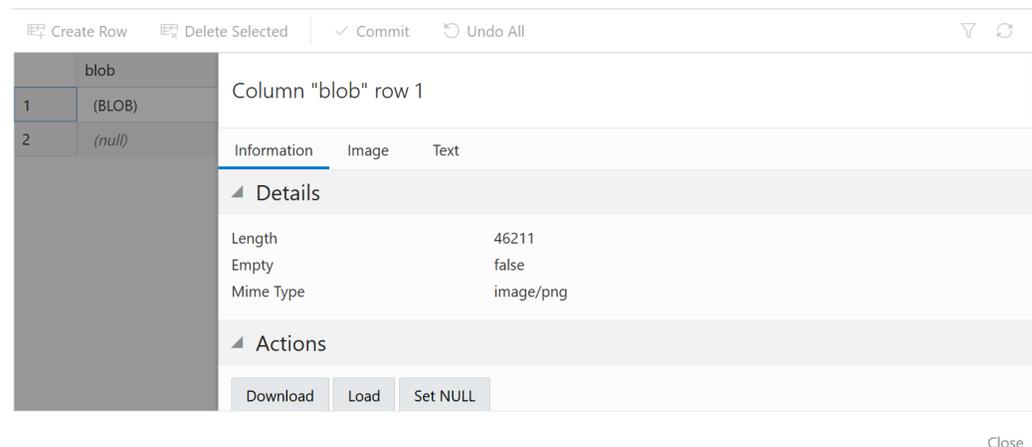
The icons available in the Data pane are:

- * **Create Row:** Insert a new row into the database table. When you insert a row using the **Create Row** icon, the row is committed into the database.
- * **Delete Selected:** Mark the selected row for deletion. When you mark a row for deletion, the border of the row changes to red.
- * **Commit:** Commit all changes made to the database.
- * **Undo All:** Revert all changes that are marked for commit.

Use the **Filter** icon at the top right corner to filter the column data. You can also right-click a cell to access the context menu to count rows, view a single record, export or copy the cell text to clipboard.

To view a Binary Large Object data type (BLOB), click the pencil icon for a BLOB data type. In the View Value dialog:

- * The **Image** tab displays the loaded image, if the loaded BLOB type is an image.
- * The **Text** tab displays the text file, if the loaded BLOB type is text .
- * The **Information** tab displays the details and allows you to perform the following actions:
 - * **Download:** To download the image/text file of BLOB data type .
 - * **Load:** To insert an image/text of BLOB data type.
 - * **Set NULL:** To delete the object and set the value as NULL.



- **Edit** edits the properties of an existing object.
- **Add** creates an object based on the object type selected.
- **Use as Template** creates an object by using the properties of an existing object as the template.
- **REST**
 - * **Enable** enables REST access for the database object. See [Enabling REST Access for a Database Object](#)

A REST enabled object is indicated by a **REST Enabled** icon  in the Navigator tab.

 - * **Disable** disables REST access for the database object after it is enabled. See [Disabling REST Access for a Database Object](#)

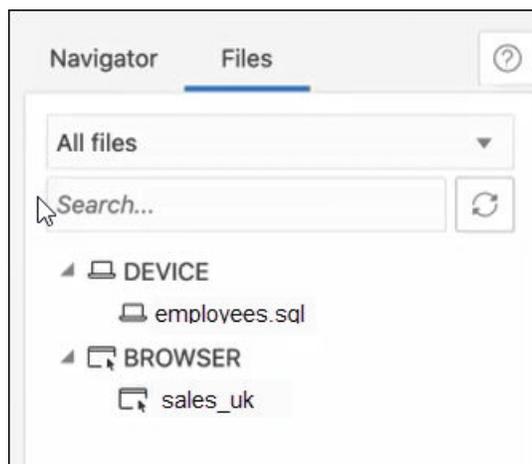
- * **cURL Command** generates a cURL call for a selected HTTP method for the database object. See [Generating cURL Requests for a REST-Enabled Database Object](#)
- **Data Loading** loads data from local files into a table.
- **Quick DDL** generates Data Definition Language statements for the object.
- **Refresh** : Refreshes the objects or worksheets listed in the left pane.
- **Object Submenu** : Opens the Create Object dialog to create a new object based on the type selected in the drop-down list.
- **Help** : Provides contextual help documentation.

Files Tab

Enables you to open files from your browser or local device.

Note

The DEVICE category in the left pane is displayed only when using a Chromium-based browser in a secure context (HTTPS).



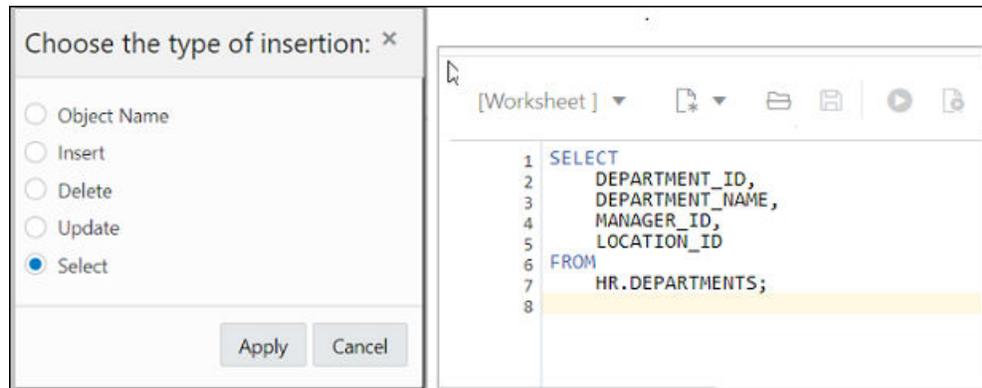
All files: Use the drop-down list to filter files by browser or device.

The context menu options for a file are **Open** and **Delete**. In Device, the corresponding option for deleting a file is **Forget**. In this case, the file is not deleted, instead the reference to the file is removed.

Drag and Drop Objects and Files into the Worksheet

You can drag objects from the left pane and drop them into the worksheet editor in the right pane.

- If you drag and drop a table or view, you are prompted to select one of the following SQL statements: Insert, Update, Select, or Delete. For example, if you choose Select, a Select statement is constructed with all columns in the table or view. You can then edit the statement, for example, modifying the column list or adding a WHERE clause.

Figure 3-2 Insert Select Query

If you choose Object Name, the name of the object prefixed by the schema name is added to the worksheet.

- If you drag and drop a function or procedure, you can choose to insert the name or the PL/SQL code of the function or procedure in the worksheet. If you select the PL/SQL code, you can enter the parameters before inserting the code into the worksheet.

Executing SQL Statements in the Code Editor

The code editor in the SQL page enables you to run SQL statements, PL/SQL scripts, and JavaScript code. The main features include in-context code completion, syntax highlighting, and error debugging.

You can enter SQL and PL/SQL statements to specify actions such as creating a table, inserting data, selecting data, or deleting data from a table. SQL keywords are automatically highlighted. For multiple statements, you must terminate:

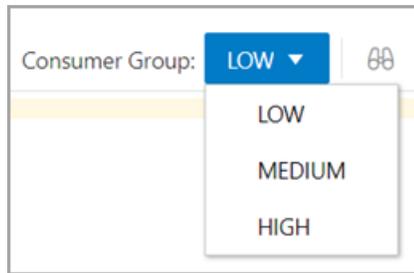
- Each non-PL/SQL statement with either a semicolon (;) or a slash (/) on a new line
- Each PL/SQL statement with a slash (/) on a new line

The **PL/SQL editor** is triggered when opening the following object types: Functions, Procedures, Packages and Types. This editor helps you detect errors in your PL/SQL code during compilation. The output includes details such as the specific line and column where the error is detected, along with a link to go to the relevant position in the code block.

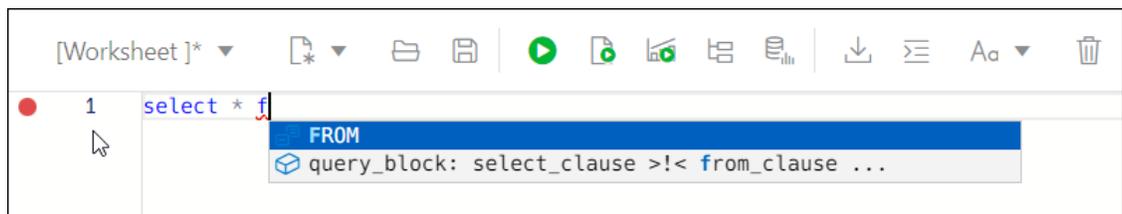
The **JavaScript worksheet mode** supports the Multilingual Engine syntax in Oracle Database release 21c. For more details, see [Support for Multilingual Engine](#).

For SQL*Plus and SQLcl statements supported in the worksheet, see [Supported SQL*Plus and SQLcl Commands in SQL Worksheet](#).

Use the Consumer Group drop-down list to select the consumer group to run your SQL or PL/SQL code. The values in the drop-down list match the database services available when connecting to the database. This feature is available only if you have the EXECUTE privilege on the CS_SESSION package.

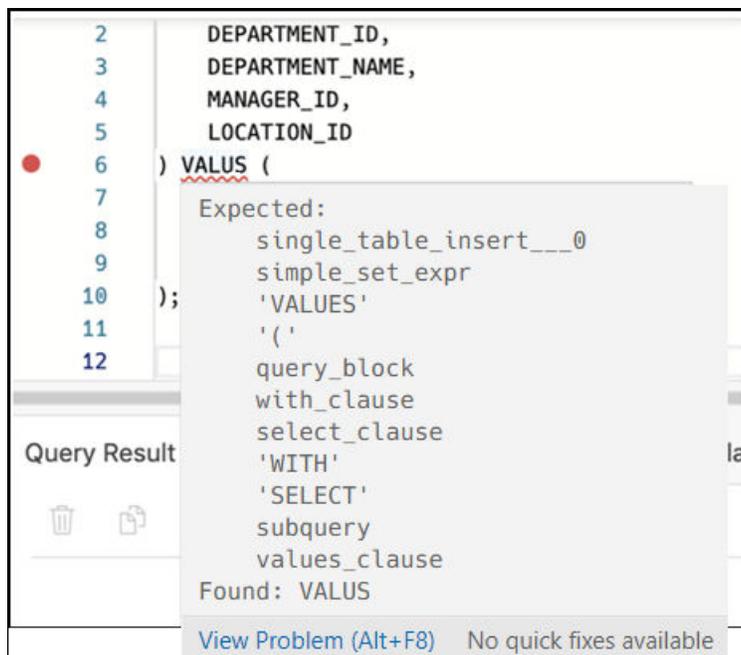


If you press **Ctrl+Space**, the editor provides you with a list of possible completions at the insertion point that you can use to autocomplete code that you are editing. This list is based on the code context at the insertion point.



The editor offers a comprehensive list of commands available through the Command Palette. To open the Command Palette, press **Ctrl+Shift+P**. For a list of keyboard shortcut keys, see [Keyboard Shortcuts](#).

An error in the code is signified by a red dot in the left gutter and a squiggle line beneath the specific text. When you hover over it, you see a pop-up displaying possible fixes for resolving the error.



You can set code editor preferences using the **Preferences** option available in the top-right user drop-down list. Some of the code editor options that you can customize are Theme (Light, Dark and High contrast dark), Font size and family, Tab size, Word wrap, Ruler, Line numbers and so on.

The SQL toolbar contains icons for the following operations:

- **Worksheet** drop-down list
 - **Open**: Opens a file from the browser or device.
 - **Open Recent**: Displays the recently accessed files. If there are more than five files in the recent list, then a **More** link is displayed for viewing the additional files.
 - **Save As**: Saves a file to the browser or device.
- **New File**
 - Select **Worksheet** to create a worksheet.
 - Select **PL/SQL** to create a PL/SQL object type. The editor switches to a PL/SQL mode.
 - Select **JavaScript** to create a JavaScript file. The editor switches to a JavaScript mode and **(JS)** is added to the file name.
- **Open** opens a file from your browser or device. To open a file from your device, in the Open File slider, click **Open File** and browse to select the file, or drag and drop the file into the slider.
- **Run Statement**
 - **Run Statement** executes the selected statements or the statement at the mouse pointer in the worksheet editor. The SQL statements can include bind variables and substitution variables of type VARCHAR2 (although in most cases, VARCHAR2 is automatically converted internally to NUMBER if necessary). A dialog box is displayed for entering variable values.
 - **Run Statement Without Pagination** executes the selected statements without wrapping it inside a `ROW_NUMBER() OVER` analytic function to implement paging. This means the query is run 'as is' in the worksheet, and only fetches the first 256 rows.
- **Run Script**
 - **Run as SQL Script** executes all statements in the worksheet editor using the Script Runner. The SQL statements can include bind variables (but not substitution variables) of type VARCHAR2 (although in most cases, VARCHAR2 is automatically converted internally to NUMBER if necessary). A dialog box is displayed for entering bind variable values.
 - **Run as JavaScript** executes the code as a JavaScript file. This option is used only with JavaScript code. If the JavaScript code is added to a PL/SQL block, then select "Run as SQL Script" to execute the script.
- **Compile** (for PL/SQL toolbar) performs a PL/SQL compilation of the subprogram.
- **Create Chart** creates a chart for the corresponding SQL statement entered in the editor. In contrast, you can partially highlight a subquery and create a chart. A slider window is displayed for entering the chart parameters. For a description of the fields, see [Creating or Editing a Chart](#).
If the SQL statement is syntactically incorrect or incomplete, an error/warning notification is displayed.

- **Explain Plan** generates the execution plan for the statement (internally executing the EXPLAIN PLAN statement). The execution plan is automatically displayed in the Explain Plan tab in the worksheet output pane. See [Viewing the SQL Output](#)
- **Autotrace** runs the statement and collects run time statistics and the actual execution plan. The Autotrace output is displayed in the Autotrace tab in the worksheet output pane. Currently, there are no preferences available.
- **Download Editor Content** downloads the content of the worksheet as a SQL file to the local system.
- **Format** formats the SQL statement in the editor, such as capitalizing the names of statements, clauses, keywords, and adding line breaks and indentation.
- **Clear** removes the statements from the editor.
- **Tour** provides a guided tour of the worksheet highlighting salient features and providing information that is useful if you are new to the interface.
- **Help** provides context-related help and provides a link to the help documentation.
- **Open in Fullscreen** opens the editor in full screen mode.
- [About Session State in Database Actions](#)
In Database Actions, a connection to the database is stateless.
- [Keyboard Shortcuts](#)
This section lists the keyboard shortcuts for various commands in the SQL page.
- [Support for Multilingual Engine](#)

About Session State in Database Actions

In Database Actions, a connection to the database is stateless.

In a stateless environment, each HTTPS request from a client maps to a new database session. Therefore, a session begins and ends with every SQL statement or script execution.

As the session state is not maintained, session attributes do not persist and commands such as ROLLBACK and COMMIT do not apply. If a SQL statement or script executes successfully, an implicit commit is performed. If it executes with an error, an implicit rollback is performed.

Therefore, when needed, include the ROLLBACK and COMMIT commands or session attributes in the PL/SQL code block that is sent to the database for a session.

The only configuration commands that persist during a session in Database Actions are:

- SET DEF[INE] <ON|OFF|*prefix_character*>
- SET ESC[APE] <ON|OFF|*escape_character*>
- SET TIMI[NG] <ON|OFF>

Keyboard Shortcuts

This section lists the keyboard shortcuts for various commands in the SQL page.

Table 3-1 Keyboard Shortcuts

Windows	MacOS	Description
Ctrl + Enter	Cmd + Enter	Runs the code as query.

Table 3-1 (Cont.) Keyboard Shortcuts

Windows	MacOS	Description
Alt + Tab	Option + Tab	Focus next element.
Ctrl+ Esc / Escape	Cmd + Esc / Escape	Remove focus from editor.
Ctrl + Down Arrow	Cmd + Down Arrow	Moves to the next SQL code from history.
Ctrl + Up Arrow	Cmd + Up Arrow	Moves to the previous SQL code from history.
Ctrl + S	Cmd + S	Saves the current worksheet.
Ctrl + O	Cmd + O	Opens the worksheet browser dialog.
Ctrl + I	Cmd + I	Downloads the content of the editor.
F1	Fn + F1	Opens the help topic.
Shift + Esc	Shift + Esc	Focus previous element.
F5	Fn + F5	Runs code as script.
F6	Fn + F6	Shows Autotrace.
F10	Fn + F10	Shows Explain Plan.
F11	Fn + F11	Creates a chart.
Ctrl + B	Cmd + B	Opens the "Convert Case" drop-down list.
Ctrl + F7	Cmd + Fn + F7	Formats code in the editor.
Ctrl + Space	Ctrl + Space	Autocompletes code (shows hints).

The following table lists the keyboard shortcuts for commands in the Command Palette.

Table 3-2 Command Palette Keyboard Shortcuts

Windows	MacOS	Description
Ctrl + Alt + Up	Cmd + Option + Up	Add Cursor Above
Ctrl + Alt + Down	Cmd + Option + Down	Add Cursor Below
Shift + Alt + I	Shift + Option + I	Add Cursors to Line Ends
Ctrl + K Ctrl + C	Cmd + K Cmd + C	Add Line Comment
Ctrl + D	Cmd + D	Add Selection To Next Find Match
Shift + Alt + Down	Shift + Option + Down	Copy Line Down
Shift + Alt + Up	Shift + Option + Up	Copy Line Up
Ctrl + U	Cmd + U	Cursor Undo
-	Cmd + Backspace	Delete All Left
-	Ctrl + K	Delete All Right
Ctrl + Shift + K	Shift + Cmd + K	Delete Line
Shift + Alt + Right	Shift + Ctrl + Cmd + Right	Expand Selection
Ctrl + F	Cmd + F	Find
Enter	Enter	Find Next
Ctrl + F3	Cmd + Fn + F3	Find Next Selection
Shift + Enter	Shift + Enter	Find Previous
Ctrl + Shift + F3	Shift + Cmd + Fn + F3	Find Previous Selection
-	Cmd + E	Find With Selection

Table 3-2 (Cont.) Command Palette Keyboard Shortcuts

Windows	MacOS	Description
Ctrl + Shift + [Option + Cmd + [Fold
Ctrl + K Ctrl + 0	Cmd + K Cmd + 0	Fold All
Ctrl + K Ctrl + /	Cmd + K Cmd + /	Fold All Block Comments
Ctrl + K Ctrl + 8	Cmd + K Cmd + 8	Fold All Regions
Ctrl + K Ctrl + 1	Cmd + K Cmd + 1	Fold Level 1
Ctrl + K Ctrl + 2	Cmd + K Cmd + 2	Fold Level 2
Ctrl + K Ctrl + 3	Cmd + K Cmd + 3	Fold Level 3
Ctrl + K Ctrl + 4	Cmd + K Cmd + 4	Fold Level 4
Ctrl + K Ctrl + 5	Cmd + K Cmd + 5	Fold Level 5
Ctrl + K Ctrl + 6	Cmd + K Cmd + 6	Fold Level 6
Ctrl + K Ctrl + 7	Cmd + K Cmd + 7	Fold Level 7
Ctrl + K Ctrl + [Cmd + K Cmd + [Fold Recursively
Ctrl + Shift + \	Shift + Cmd + \	Go to Bracket
Ctrl + G	Ctrl + G	Go to Line...
Alt + F8	Option + Fn + F8	Go to Next Problem(Error, Warning, Info)
F8	Fn + F8	Go to Next Problem in Files (Error, Warning, Info)
F7	Fn + F7	Go to Next Symbol Highlight
Shift + Alt + F8	Shift + Option + Fn + F8	Go to Previous Problem (Error, Warning, Info)
Shift + F8	Shift + Fn + F8	Go to Previous Problem in Files (Error, Warning, Info)
Shift + F7	Shift + Fn + F7	Go to Previous Symbol Highlight
Ctrl +]	Cmd +]	Indent Line
Ctrl + Shift + Enter	Shift + Cmd + Enter	Insert Line Above
-	Ctrl + J	Join Lines
Ctrl + K Ctrl + D	Cmd + K Cmd + D	Move Last Selection To Next Find Match
Alt + Down	Option + Down	Move Line Down
Alt + Up	Option + Up	Move Line up
F1 (All browsers)	Fn + F1	Open Command palette
Ctrl + Shift + P (Google Chrome only)		
Ctrl + [Cmd + [Outdent Line
Ctrl + K Ctrl + U	Cmd + K Cmd + U	Remove Line Comment
Ctrl + H	Option + Cmd + F	Replace
Ctrl + Shift + .	Shift + Cmd + .	Replace with Next Value
Ctrl + Shift + ,	Shift + Cmd + ,	Replace with Previous Value
Ctrl + Shift + L	Shift + Cmd + L	Select All Occurrences of Find Match
Alt + F1	Option + Fn + F1	Show Accessibility Help
Shift + F10	Shift + Fn + F10	Show Editor Context Menu
Ctrl + K Ctrl + I	Cmd + K Cmd + I	Show Hover
Shift + Alt + Left	Shift + Ctrl + Cmd + Left	Shrink Selection
Shift + Alt + A	Shift + Option + A	Toggle Block Comment

Table 3-2 (Cont.) Command Palette Keyboard Shortcuts

Windows	MacOS	Description
Ctrl + K Ctrl + L	Cmd + K Cmd + L	Toggle Fold
Ctrl + /	Cmd + /	Toggle Line Comment
Ctrl + M	Shift + Ctrl + M	Toggle Tab Key Moves Focus
-	Shift + Ctrl + T	Transpose Letters
Ctrl + Space	Ctrl + Space	Trigger Suggest
Ctrl + K Ctrl + X	Cmd + K Cmd + X	Trim Trailing Whitespace
Ctrl + Shift +]	Option + Cmd +]	Unfold
Ctrl + K Ctrl + J	Cmd + K Cmd + J	Unfold All
Ctrl + K Ctrl + 9	Cmd + K Cmd + 9	Unfold All regions
Ctrl + K Ctrl +]	Cmd + K Cmd +]	Unfold Recursively

Support for Multilingual Engine

Database Actions provides support for Multilingual Engine (MLE) by enabling you to run JavaScript code in the worksheet.

Prerequisites

For the availability of MLE features in Database Actions, you need the:

- `DBMS_MLE` package in Oracle Database Release 21c and later versions. For more information, see `DBMS_MLE` in *Oracle AI Database PL/SQL Packages and Types Reference*.
- `EXECUTE DYNAMIC MLE` and `EXECUTE ON JAVASCRIPT` privileges assigned to you.

You can work with JavaScript code in the worksheet in the following ways:

- Create a JavaScript worksheet
- Execute JavaScript code in a standard worksheet
- Execute JavaScript code as a PL/SQL block

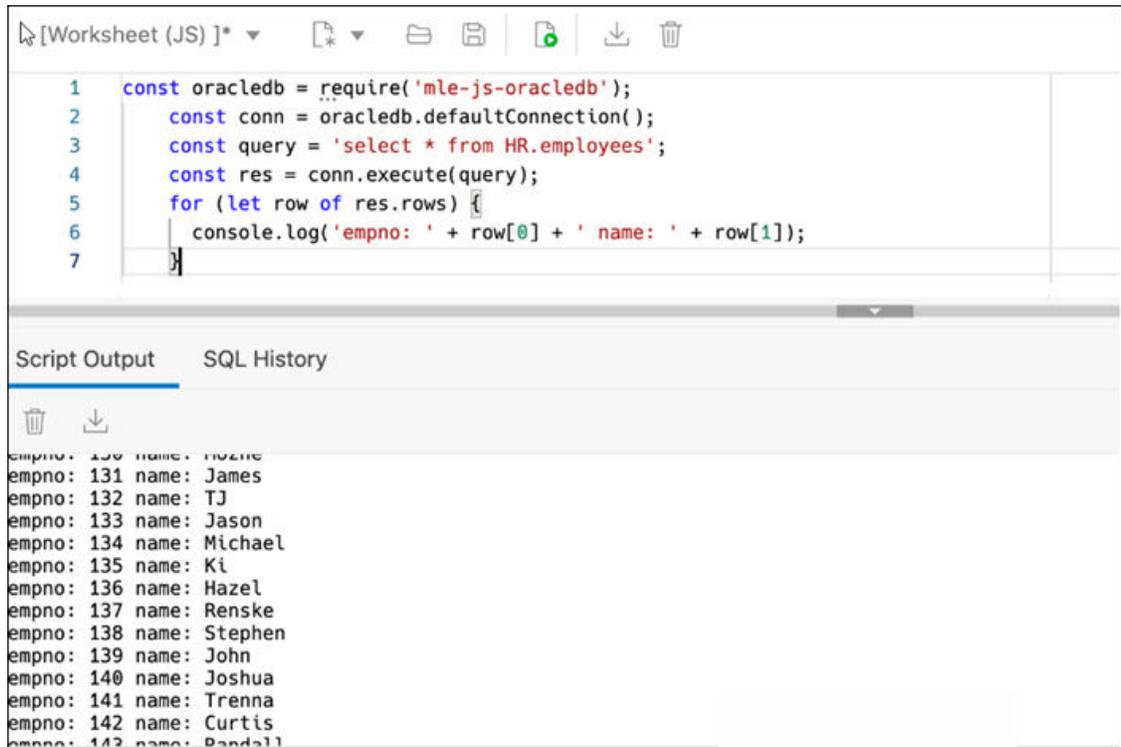
Create a JavaScript Worksheet

You can open the worksheet in JavaScript mode. You will also see the toolbar icons and output tabs change to reflect the JavaScript mode.

To create and save a JavaScript worksheet:

1. In the toolbar, expand the **New File** icon and select **JavaScript**.
2. When you enter code in the worksheet, the JavaScript code is automatically highlighted.
3. Execute the JavaScript code in the worksheet using **Run Script**.
4. Click **Save**.

In the Files pane, **(JS)** is added to the JavaScript file name. This enables you to quickly identify the JavaScript file.



The screenshot shows a web-based SQL worksheet interface. At the top, there is a toolbar with icons for file operations and a dropdown menu showing "[Worksheet (JS)]*". Below the toolbar is a code editor with the following JavaScript code:

```
1  const oracledb = require('mle-js-oracledb');
2  const conn = oracledb.defaultConnection();
3  const query = 'select * from HR.employees';
4  const res = conn.execute(query);
5  for (let row of res.rows) {
6    console.log('empno: ' + row[0] + ' name: ' + row[1]);
7  }
```

Below the code editor, there are two tabs: "Script Output" (which is selected) and "SQL History". The "Script Output" tab displays the following log output:

```
empno: 130 name: Neena
empno: 131 name: James
empno: 132 name: TJ
empno: 133 name: Jason
empno: 134 name: Michael
empno: 135 name: Ki
empno: 136 name: Hazel
empno: 137 name: Renske
empno: 138 name: Stephen
empno: 139 name: John
empno: 140 name: Joshua
empno: 141 name: Tenna
empno: 142 name: Curtis
empno: 143 name: Randall
```

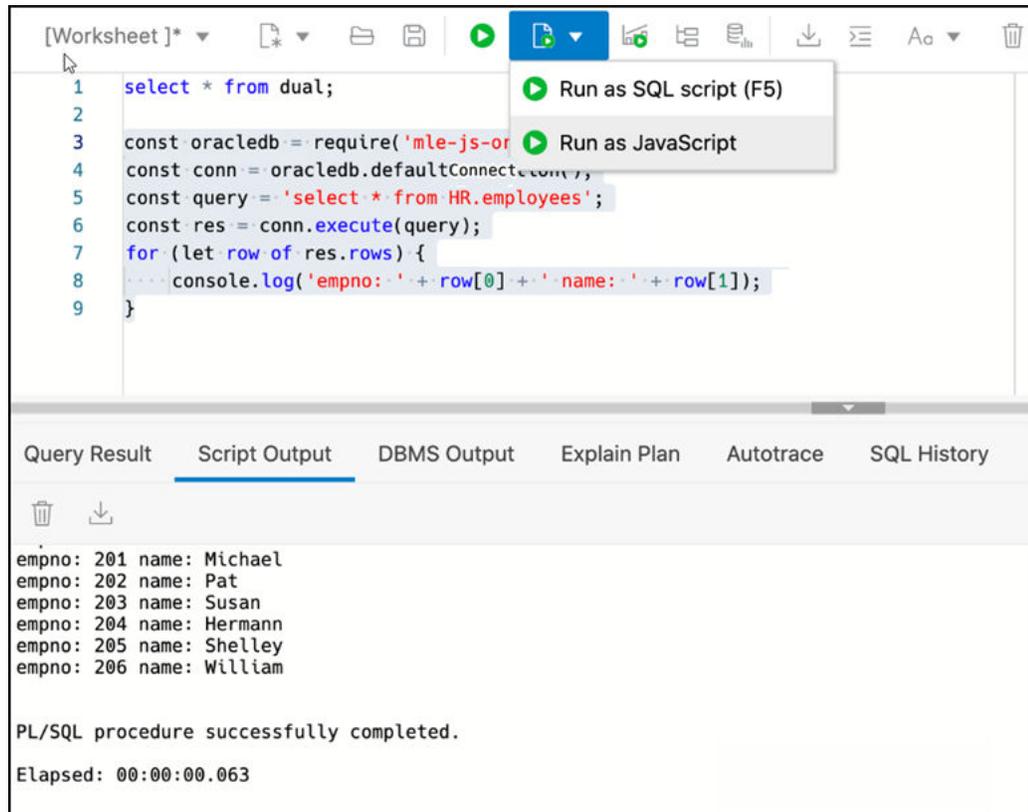
Execute JavaScript code in a standard worksheet

In a standard worksheet (when the worksheet is not in JavaScript mode):

1. Select the JavaScript code to execute.
2. In the worksheet toolbar, expand the **Run Script** icon and select **Run as JavaScript**.

Note

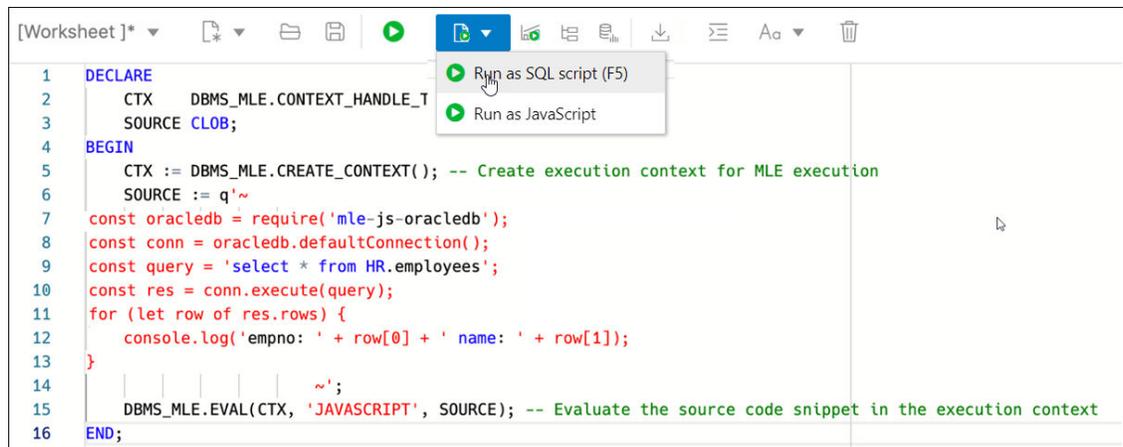
Selecting **Run as SQL script** results in an error.



Execute JavaScript code as a PL/SQL block

You can use a PL/SQL code block to execute JavaScript code.

From the worksheet toolbar, expand the **Run Script** icon and select **Run as SQL script (F5)**.



Viewing the SQL Output

The lower right pane in SQL displays the output of the operation executed in the SQL editor.

The following figure shows the output pane in the SQL page.

Figure 3-3 Output Pane

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and formatting. The worksheet contains the following SQL query:

```

1 SELECT
2   DEPARTMENT_ID,
3   DEPARTMENT_NAME,
4   MANAGER_ID,
5   LOCATION_ID
6 FROM
7   HR.DEPARTMENTS;
8

```

The output pane is active, showing the results of the query in a table format. The table has five columns: an index column, department_id, department_name, manager_id, and location_id. The results are as follows:

	department_id	department_name	manager_id	location_id
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400

The output pane also shows tabs for Query Result, Script Output, DBMS Output, Explain Plan, Autotrace, SQL History, and Data Loading. The Query Result tab is selected, and the execution time is shown as 0.019 seconds.

The output pane has the following tabs:

- **Query Result:** Displays the results of the most recent Run Statement operation in a display table.
- **Script Output:** Displays the text output from your statements executed as a script using the script engine.
- **DBMS Output:** Displays the output of DBMS_OUTPUT package statements.
- **Explain Plan:** Displays the plan for your query using the Explain Plan command. The default view is the diagram view. For more information, see [Using the Explain Plan Diagram](#).
- **Autotrace:** Displays the session statistics and execution plan from v\$sql_plan when executing a SQL statement using the Autotrace feature. Displays the output if you clicked the Autotrace icon.
- **SQL History:** Displays the SQL statements and scripts that you have executed. To re-enter a previously executed query in the worksheet, double-click the query in the history list. You can search for specific statements by clicking the Search icon. The Search functionality is case-sensitive, retrieves all entries that contain the search text, and does not require wildcard characters.

Note

The executed statements are saved in the database, and are therefore available across all devices. For a read-only database, the SQL statements are saved on your browser.

- **Data Loading:** Displays a report of the total rows loaded and failed for all visible tables (including tables from other schemas).

The icons in this pane are:

- **Clear output:** Clears the output.
- **Show info:** Displays the SQL statement for which the output is displayed.
- **Open in new tab:** Opens the query result or explain plan in a new window.
- **Share Query Results:** This is only applicable to the Query Result and is available for use only when a query has been successfully executed, generating result data. If no result data is available, then this option is disabled for use.

It enables you to share query results with others using a shareable URL to the query results. On clicking the Share button, a URL is generated. By simply copying it to your clipboard, you can easily share the results with non-authenticated users. Therefore, anyone you provide this URL to can access it without authentication (like a user name/password), and for the time limit you can define using **Expiration date**. If you don't specify expiration date and time, then it will have a default expiration date as the current date and a default minimum expiration time of one hour. Although a default minimum expiration duration is set, there is no maximum limit on how long the expiration duration can be set. This means that you can set the expiration duration as long as you like as per your specific business requirements.

Note

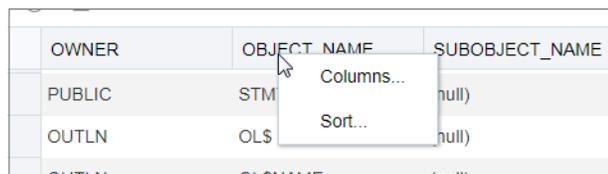
Query Results that you share using this option are based on a specific query that is executed at a particular point in time. If the underlying objects or views referenced in the query are modified later, the query result data may change or vary.

- **Download:** This is applicable only for Query Result. Enables you to download the query result to your local computer in CSV, JSON, XML, or TEXT (.tsv) format.

In the Query Result tab, in the display table, the context menu (right-click) for the row header consists of the following:

- **Columns** enables you to select columns to hide.
- **Sort** displays a dialog box for selecting columns to sort by. For each column, you can specify ascending or descending order, and you can specify that null values be displayed first.

Figure 3-4 Context Menu for Row Header



OWNER	OBJECT_NAME	SUBOBJECT_NAME
PUBLIC	STM	(null)
OUTLN	OL\$	(null)
OUTLN	OLNAME	(null)

The context menu for the rest of the display table consists of the following commands:

- **Count Rows** displays the number of rows in the result set for your query.
- **Single Record View** enables you to view data for a table or view, one record at a time.

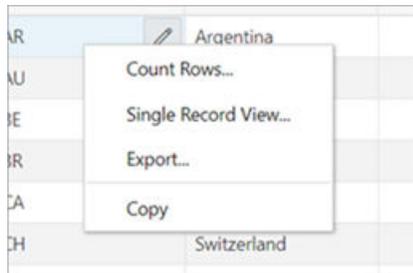
- **Export** generates the file for download based on the format selected, which can be XML, CSV (comma-separated values including a header row for column identifiers), Insert , Delimited, Fixed, HTML, JSON, or TEXT.
 - **Format:** Select the format to export from the drop-down list.
 - **Line Terminator:** Identifies the terminator for each line. The line terminator is not included in the data exported. If the preview page shows the data in one single row, the correct terminator is not specified.
 - **Header:** Controls whether the first row is a header row or the first row of data.
 - **Left and Right Enclosure:** Enclosures are used for character data and are optional. Enclosures are not included in the data exported.

Note

If a popup blocker is enabled, it will prevent the file from downloading.

- **Copy** copies data from a cell or a row or a range of rows.

Figure 3-5 Context Menu



- [Using the Explain Plan Diagram](#)

Using the Explain Plan Diagram

The Explain Plain diagram view is a graphical representation of the contents of `PLAN_TABLE`, which is the default table for results of the `EXPLAIN PLAN` statement. The hierarchical nature of the steps in the execution plan is depicted in the diagram.

By default, three levels of steps are visible in the diagram. You can use the **+/-** signs at the bottom of each step (available when the step has children) to expand or collapse. To view all

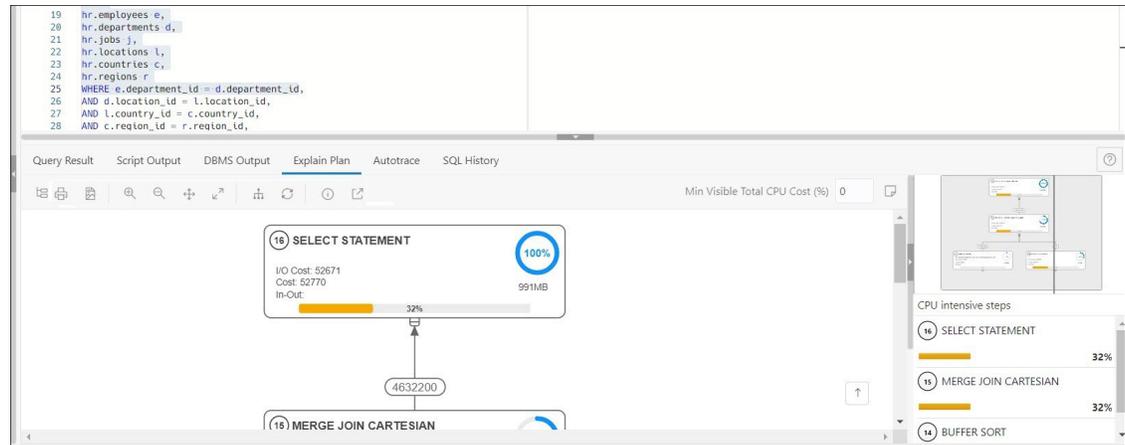
steps in the diagram, use  **Expand All** in the toolbar.

The diagram also provides the following details:

- Cardinality (number on the arrow to the parent step), which is the number of rows processed
- Operation and options applied in that step
- Execution order, which is the sequential number in the order of execution
- Access predicates CPU cost in percentage (orange bar)

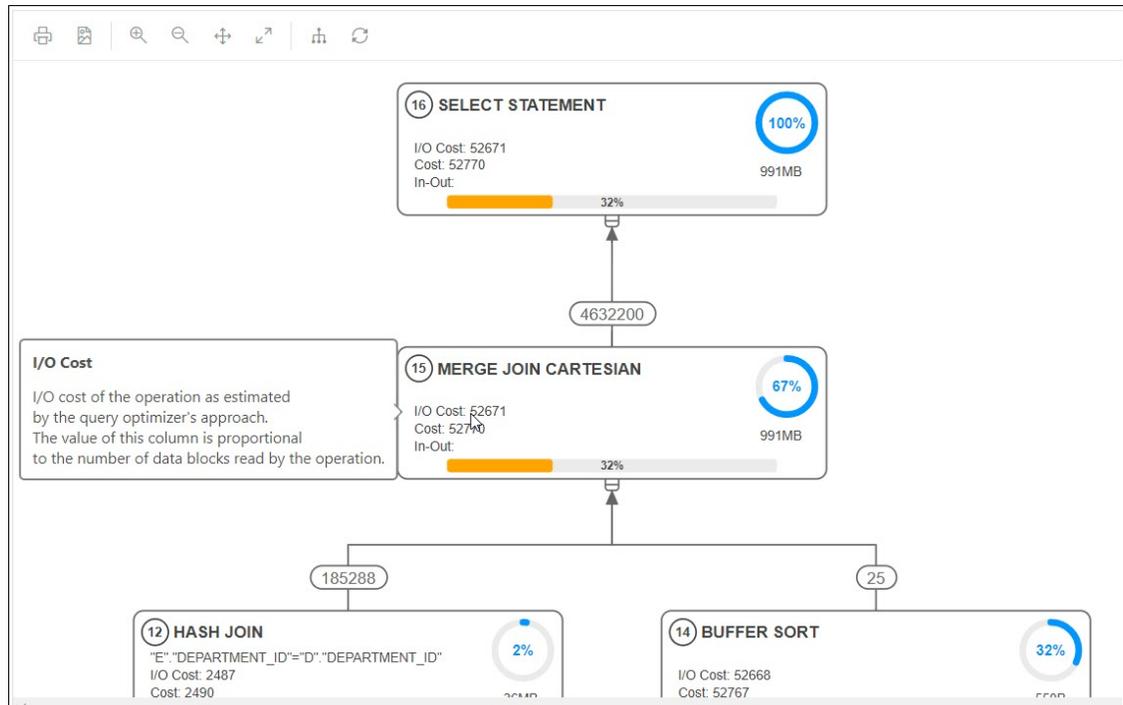
- Total CPU cost for the step in percentage (blue circle)
- Estimated I/O Cost, Bytes processed and Cost metrics

You can see a brief description pop-up when you hover over any of these statistics in a step.



The icons in the toolbar are:

- **Advanced View/Diagram View:** Switches between Advanced and Diagram views. The Advanced View displays data from `PLAN_TABLE` in a mixed tabular/tree view. The Diagram View displays the data in a flowchart format.
- **Print Diagram:** Prints the diagram.
- **Save to SVG:** Saves the diagram to file in SVG format.
- **Zoom In, Zoom Out:** If a step is selected in the diagram, clicking the Zoom In icon ensures that it remains at the center of the screen.
- **Fit Screen:** Fits the entire diagram in the visible area.
- **Actual Size:** Sets the zoom factor to 1.
- **Expand All:** Displays all steps in the diagram.
- **Reset Diagram:** Resets the diagram to the initial status, that is, only three levels of steps are displayed.
- **Show Info:** Shows the `SELECT` statement used by the Explain Plan functionality.
- **Open in New Tab:** Opens the diagram view in a new tab for better viewing and navigation. The diagram is limited to the initial `SELECT` statement.
- **Min Visible Total CPU Cost(%):** Defines the threshold to filter steps with total CPU cost less than the the provided value.
Enter a value between 0 and 100. There is no filtering for 0.
- **Plan Notes:** Displays the Explain Plan notes.



Properties

Double-click or press **Enter** on a selected step to open the Properties slider, which provides more information about that step. See `PLAN_TABLE` in *Oracle Database Reference* for a description of each property.

The Properties slider shows:

- All information for that step extracted from `PLAN_TABLE` in a tabular format. Nulls are excluded.
You can select **JSON** to view the properties in JSON format.
- Information from `OTHER_XML` column of `PLAN_TABLE`.
The information is displayed in JSON format.

Navigation

- Press the **Tab** key to move through the steps in the execution order. The selected step has a blue border around it.

To move in the reverse direction, press the **Shift + Tab** keys.

If no step is selected, pressing the **Tab** key selects the step with execution number 1.

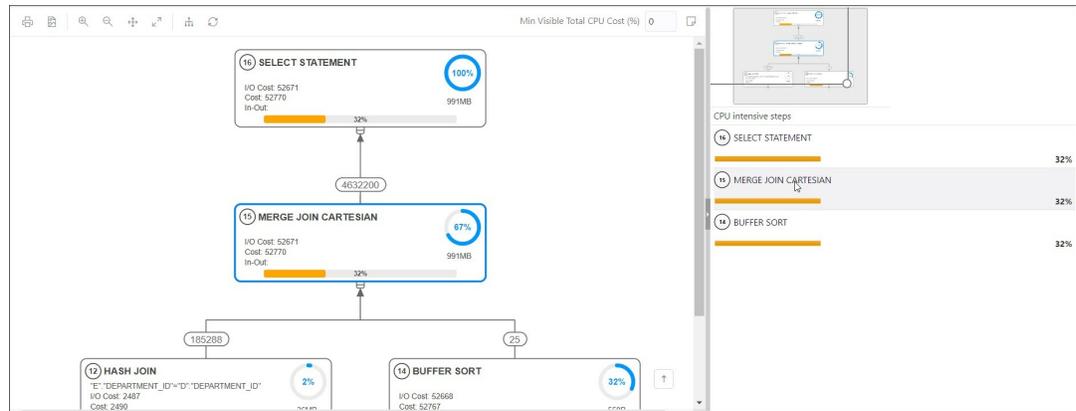
- Depending on the zoom level, use horizontal and vertical scrollbars to view different parts of diagram.

Click the left mouse button and hold it to pan the diagram around up and down.



Use the icon at the bottom right to scroll to the top of the diagram.

- The Diagram Navigator is at the top right corner and represents a smaller copy of the diagram on a grey background. The rectangle border allows zoom-in and zoom-out operations and moves to show different parts of the diagram.



The diagram navigator shows a list with steps having more than 1% CPU cost in descending order. Click a step in the list to navigate to the same step in the diagram, enabling you to see it in the context of the other steps.

Loading Data

In the SQL page, you can load data from one or more local files into one or more tables.

The file formats that you can load are CSV, XLS, XLSX, TSV, TXT, XML, JSON, and Newline Delimited JSON (NDJSON). For XML, JSON, and NDJSON files, see [Format Specifications for JSON and XML Files](#).

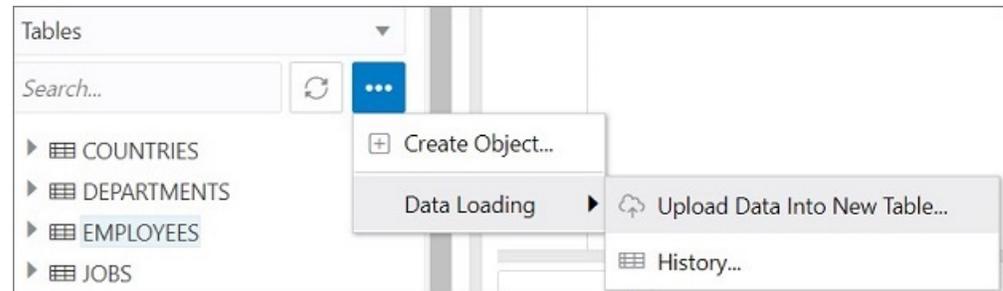
- [Loading Data from a Local File to a New Table](#)
- [Loading Data from a Local File to an Existing Table](#)
- [Loading Data from Multiple Local Files into Multiple Tables](#)
- [Loading Data from a Local File to a New Table](#)
- [Loading Data from a Local File to an Existing Table](#)
- [Loading Data from Multiple Local Files into Multiple Tables](#)
- [Format Specifications for JSON and XML Files](#)

Data has to be stored in a particular format for JSON and XML files to load them successfully into a table.

Loading Data from a Local File to a New Table

To load data from a local file to a new table:

1. You can start in one of the following ways:
 - In the Navigator tab, in the left pane, click **Object submenu** **...**, select **Data Loading**, and then select **Upload Data into New Table**.

Figure 3-6 Upload Data to New Table Option

- In the Navigator tab, drag and drop the local file into the left pane. When you drag a file into the pane, the following message is displayed `Drop the file here to start.`

The Upload Data into New Table is displayed. A preview of the data is displayed in a grid format.

2. Click **Show/Hide options**  to display options that you can modify for data preview:
 - **Column names:** Select **Get from file** to display column headers in the first row.
 - **Encoding:** An option to select the encoding type is visible when the loaded file is in plain text format (CSV, TSV, or TXT). The default encoding type is UTF-8.
 - **Text enclosure** and **Field delimiter:** These options are visible only when the selected file is in plain text format (CSV, TSV, or TXT). Select or enter the character used in the source file for text enclosure and field delimiter.
 - **Rows to skip:** Enter or use the up and down arrows to select the number of rows to skip.
 - **Preview size:** Enter or use the up and down arrows to select the number of rows to preview.
 - **Limit rows to upload:** If you select this option, you need to specify the rows to load. Use the up and down arrows to select the number of rows to load.

To remove the options selected and the data preview, click **Clear**.

After selecting the required options, click **Apply**, and then click **Next**.

3. In Table Definition, do the following:
 - In the **Table Name** field, enter a name for the target table.
 - Select the check box at the beginning of a row to add the column to the target table.
 - Select or enter values for column attributes such as Column Name, Column Type, Precision, Scale, Default, Primary Key and Nullable.
 - The Format Mask column appears for date, timestamp and numeric types of data. For date and timestamp types, you must select a value from the drop-down list or type the value in the Format Mask field. For numeric types, the format mask is optional.

For a date and timestamp column, you need to supply a compatible format mask that describes the data being uploaded. For example, if the date data looks like `12-FEB-2021 12.21.30`, you need to supply a date mask of `DD-MON-YYYY HH.MI.SS`. The format mask is automatically determined based on the data in the file. You need to review the suggested format mask and if needed, modify it by entering the format directly into the target cell.

Figure 3-7 Table Definition Step in Upload Data into New Table

Upload Data Into New Table

1 Data preview 2 Table definition 3 Review

File: employees.csv Schema: PDBDBA Table name: EMPLOYEES

<input checked="" type="checkbox"/>	Column Name	Column Type	Length/Precision	Scale	Default	Primary Key	Nullable	Format mask
<input checked="" type="checkbox"/>	EMPLOYEE_NO	NUMBER				<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	NAME	VARCHAR2	4000			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	EMAIL	VARCHAR2	4000			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	PHONE_	NUMBER				<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	HIRE_DATE	TIMESTAMP WITH TIM...	4000			<input type="checkbox"/>	<input checked="" type="checkbox"/>	DD-MM-RRRR
<input checked="" type="checkbox"/>	LAST_NAME	VARCHAR2	4000			<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Click **Next**.

- Review the generated DDL code based on the selections made in the previous screens. The mapping of the source to target columns are also displayed.

Click **Finish**. After the data is successfully loaded, the new table is displayed in the Navigator tab.

- For a detailed report of the total rows loaded and failed, do one of the following:
 - Right-click the table in the Navigator tab, select **Data Loading**, and then select **History**. This displays the report for a specific table.
 - In the Navigator tab, select Object submenu *******, select **Data Loading**, and then select **History**. This displays the report for all tables in the schema that is selected in the Navigator tab.
 - In the worksheet output pane, select the **Data Loading** tab. This displays the report for all visible tables (including tables from other schemas).

A summary of the data loaded is displayed in the History dialog. If any data failed to load, you can view the number of failed rows in the Failed Rows column. Click the failed rows column to open a dialog showing the failed rows.

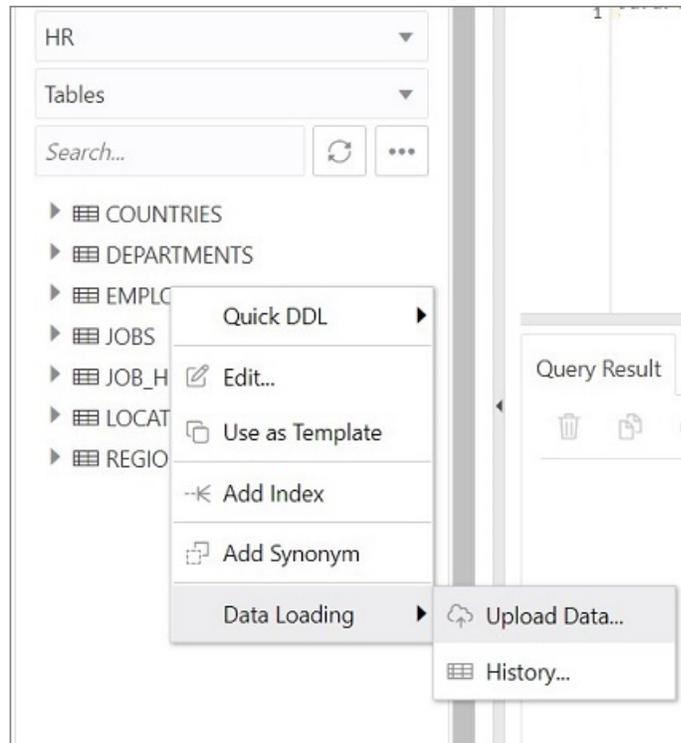
In the History dialog, you can also search for files loaded by schema name, table name, or file name. To remove the loaded files, click Remove all history .

You can also load data from a file to a new table using the steps in [Loading Data from Multiple Local Files into Multiple Tables](#).

Loading Data from a Local File to an Existing Table

To load data from a local file to an existing table:

- In the Navigator tab, in the left pane, right-click the table that you want to load data into, select **Data Loading**, and then select **Upload Data**.

Figure 3-8 Upload Data Option for an Existing Table

The Import data dialog is displayed.

2. Drag and drop the file from your system into the dialog, or click **Select Files** to browse for the file and open it.

A preview of the data is displayed in a grid format.

3. Click **Show/Hide options**  to display options that you can modify for data preview:
 - **Column names:** Select **Get from file** to display column headers in the first row.
 - **Encoding:** An option to select the encoding type is visible when the loaded file is in plain text format (CSV, TSV, or TXT). The default encoding type is UTF-8.
 - **Text enclosure** and **Field delimiter:** These options are visible only when the selected file is in plain text format (CSV, TSV, or TXT). Select or enter the character used in the source file for text enclosure and field delimiter.
 - **Rows to skip:** Enter or use the up and down arrows to select the number of rows to skip.
 - **Rows to load:** Enter or use the up and down arrows to select the number of rows to load.
 - **Preview size:** Enter or use the up and down arrows to select the number of rows to preview.

To remove the options selected and the data preview, click **Clear**.

After selecting the required options, click **Apply**, and then click **Next**.

4. In Data mapping, match the data in the file to the appropriate columns in the target table. By default, the matching is done using column name.

Figure 3-9 Data Mapping

Source column	Target column	Format	Row 1	Row 2	Row 3
Employee No	EMPLOYEE_ID		100	101	102
Name	- Select -		Steven	Rob	Kristy
Email	EMAIL		s@gmail.com	r@gmail.com	k@gmail.com
Phone	- Select -		9876541244	8765412345	1234567898
Hire Date	HIRE_DATE		17-06-1987	18-08-2000	18-02-2004
Last Name	LAST_NAME	YYYY-MM-DD	CC	EE	FF

To modify, click **Show/Hide options** . In Match columns by:

- Select **Name** to match columns based on the name of the column in the target table.
- Select **Position** if you want to match columns based on the position of the column in the target table.
- Select **None** to remove the current selections and to select the target column for each source column from the drop-down list.

Note

Based on the data in the file, attempts are made to automatically retrieve the correct format mask of date-based columns. If this is incorrect, you can change the suggested format by entering it directly into the target cell.

If there are any issues to resolve, you see a notification such as 1 pending actions on the top right of the dialog.

Click **Next**.

5. A summary of the previous screens is displayed. Click **Finish**.

The data will start uploading to the target table. After it is completed, an entry is added to the Log with the status of the operation. To view the Log, click the timestamp notification at the bottom of the page. If the operation is successful, a **Data Import Completed** notification is displayed.

6. For a detailed summary of the upload process, right-click the table in the Navigator tab, select **Data Loading**, and then select **History**. A summary of the data loaded is displayed in the Data Loading History dialog.

If any data failed to load, you can view the number of rows in the Failed Rows column. Click the column to open a dialog showing the failed rows.

In the Data Loading History dialog, you can also search for files loaded by schema name, table name, or file name. To remove the loaded files, click Remove all history .

You can also load data from a file to an existing table using the steps in [Loading Data from Multiple Local Files into Multiple Tables](#).

Loading Data from Multiple Local Files into Multiple Tables

To load data into multiple tables from multiple files:

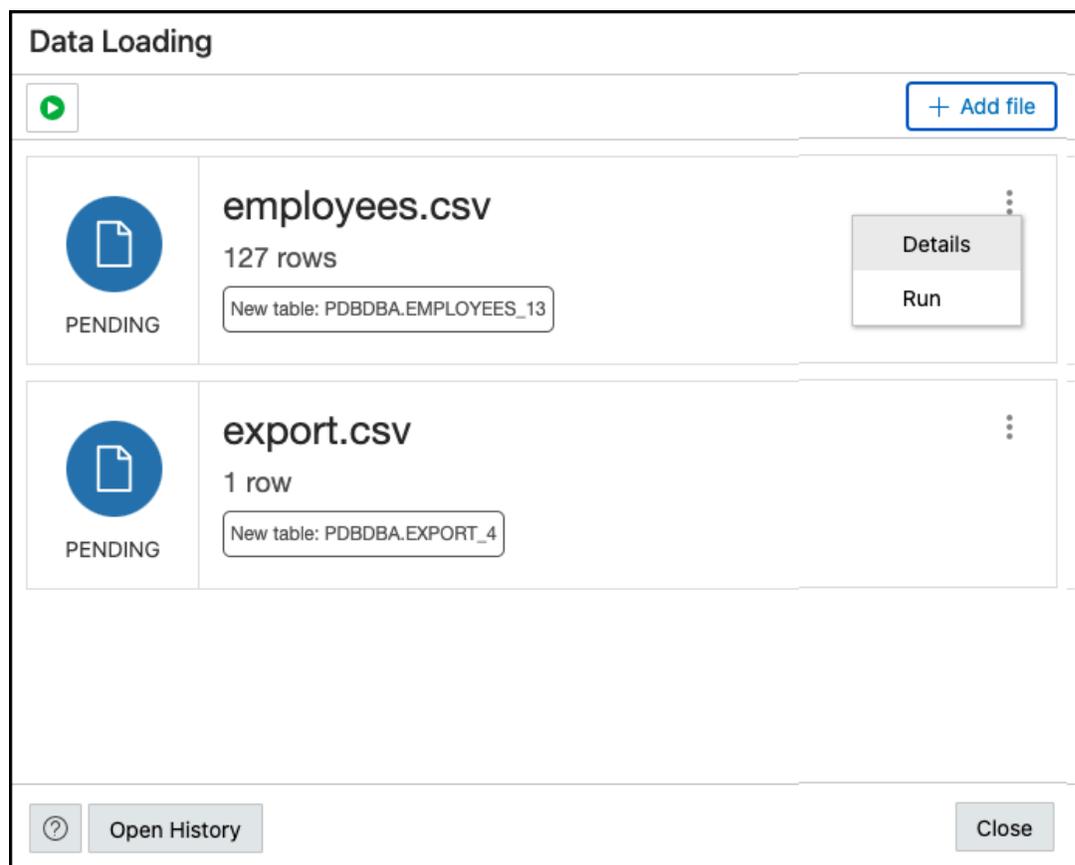
1. In the SQL page, on the top right, click **Data Load**.

The Data Loading slider appears.

2. Click **Add File** to browse for one or more files and add them concurrently. Alternatively, you can drag and drop the files into the Data Loading slider.

The added files are displayed as cards.

3. To open a file, click the name of the file or click  at the top right corner of the card and select **Details**.



In the Details page, a preview of the file is displayed. Click **Next**. To go to the previous page, click **All Files**.

Note

A preview of the entire file is shown the first time the file is loaded. Subsequently if any changes are made to the settings, the preview is set at a maximum of 10 rows. For a description of the settings, see step 2 in [Loading Data from a Local File to a New Table](#).

- In the Target Details page, the **Actions** field contains the following two options:
 - Create New Table** to create a table. By default, this option is selected and the **Table Name** field is prefilled.
 - Append to Existing Table** to add the file to an existing table. The data loader attempts to match the file name to an existing table name. Expand **Table Rows Preview** to preview the existing table.

For a description of the mapping options, see step 3 in [Loading Data from a Local File to a New Table](#) and step 4 in [Loading Data from a Local File to an Existing Table](#).

Click **All Files** to return to the initial Data Loading page.

- To load data into a table, at the top right of the card, click  and select **Run**.

To load data into multiple tables simultaneously, click  **Run All**.

The files that are successfully uploaded are displayed with a green colour icon and a check mark and the status `UPLOADED`.

Some other possible statuses are:

- `UPLOADED WITH WARNING` in yellow.
- `NOT UPLOADED BECAUSE OF ERRORS` in red.
- `UPLOADED WITH ERRORS` where you can see the number of failed rows.
- `CONTAINS WARNING(S)` indicates that something in the settings is wrong. This file will not be loaded until the warnings are resolved.

Data Loading	
UPLOADED	Target table: PDBDBA.EMPLOYEES2
 UPLOADED	export.csv 1 row New table: PDBDBA.EXPORT_4
 UPLOADE...	numeric types 1.csv 5 total row(s), 1 row(s) failed New table: PDBDBA.NUMERIC_TYPES_1_2
 NOT UPL...	employees.csv 127 rows New table: ANONYMOUS.EMPLOYEES_13
 PENDING	export.csv 1 row New table: PDBDBA.EXPORT_5
 CONTAIN...	custom_char.tsv 1 row Target table: PDBDBA.CUSTOM_CHAR

- After a file is loaded, go to the Details slider to see the third step named **Results**.

The Results step shows the total number of rows loaded along with the number of failed rows.

To view all previous files that have been loaded along with their loading statuses, click **Open History**. You can filter by schema and table name.

Format Specifications for JSON and XML Files

Data has to be stored in a particular format for JSON and XML files to load them successfully into a table.

The format specifications are described in the following sections.

- [JSON Files](#)

- [XML Files](#)

JSON Files

For JSON files, the conversion for primitive types to table columns is supported only for top-level data. Nested objects are saved as JSON strings such as VARCHAR2 (JSON) or CLOB (JSON).

Note

JSON check constraints are available only for Oracle Database 12c and later releases.

Consider the following JSON file as an example:

```
[
  {
    "ItemNumber": 1,
    "Description": "One Magic Christmas",
    "Part": {
      "UnitPrice": 19.95,
      "UPCCode": 13131092899
    },
    "Quantity": 9,
    "Total": 179.55
  },
  {
    "ItemNumber": 2,
    "Description": "Lethal Weapon",
    "Part": {
      "UnitPrice": 17.95,
      "UPCCode": 85391628927
    },
    "Quantity": 5,
    "Total": 89.75
  }
]
```

Load the JSON file using "Upload Data" in the SQL page, and it is converted to the following table with two rows. `part` is a nested object that is assigned the column type CLOB (JSON) during data mapping.

	itemnumber	description	part	quantity	total
1	1	One Magic Christmas	{"UnitPrice":19.95,"UPCCode":131310...	9	179.55
2	2	Lethal Weapon	{"UnitPrice":17.95,"UPCCode":853916...	5	89.75

XML Files

This section lists the specifications for loading XML files.

- Attributes will have their own columns

If the XML data is structured as:

```
<?xml version="1.0"?>
<catalog>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <publisher>John Doe</publisher>
  </book>
</catalog>
```

The generated columns in the table are `id`, `author`, `title`, `genre`, and `publisher`.

- Two or more levels of nesting are needed to parse the data

In the following example, the data that needs to be parsed will not be located because it has only one level of nesting (`catalog`).

```
<?xml version="1.0"?>
<catalog>
  <author>Ralls, Kim</author>
  <title>Midnight Rain</title>
  <genre>Fantasy</genre>
  <publisher>John Doe</publisher>
</catalog>
```

However, the following examples will work:

```
<?xml version="1.0"?>
<catalog>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <publisher>John Doe</publisher>
  </book>
</catalog>
```

or

```
<?xml version="1.0"?>
<catalog>
  <bookstore>
    <book id="bk102">
      <author>Ralls, Kim</author>
      <title>Midnight Rain</title>
      <genre>Fantasy</genre>
      <publisher>John Doe</publisher>
    </book>
  </bookstore>
</catalog>
```

```

    </bookstore>
</catalog>

```

or

```

<?xml version="1.0"?>
<catalog>
  <bookstore>
    <shelf>
      <book id="bk102">
        <author>Ralls, Kim</author>
        <title>Midnight Rain</title>
        <genre>Fantasy</genre>
        <publisher>John Doe</publisher>
      </book>
    </shelf>
  </bookstore>
</catalog>

```

- Special characters such as hyphen (-) and period (.) in tag names are replaced by underscore (_) in the column name

XML tag names can contain hyphens and periods. Since the parser is converting XML to JSON, these characters become invalid object keys.

```

<?xml version="1.0"?>
<catalog>
  <book id="bk102">
    <author-name>Ralls, Kim</author-name>
    <title.1>Midnight Rain</title.1>
    <genre>Fantasy</genre>
    <publisher>John Doe</publisher>
  </book>
</catalog>

```

The generated columns are `id`, `author_name`, `title_1`, `genre`, and `publisher`.

- First-level only-text tags are ignored

```

<?xml version="1.0"?>
<catalog>
  <library> New Age Library </library>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <publisher>John Doe</publisher>
  </book>
</catalog>

```

The `<library>` tag is ignored and only the content of the `<book>` tag is taken into account. The generated columns are `id`, `author`, `title`, `genre`, and `publisher`.

- First-level repetitive data is interpreted as an array of values

```
<?xml version="1.0" encoding="UTF-8"?>
<items id="orders">
  <item_number>1</item_number>
  <description>One Magic Christmas</description>
  <part>
    <unit_price>19.95</unit_price>
    <upccode>13131092899</upccode>
  </part>
  <quantity>9</quantity>
  <total>179.55</total>
  <item_number>2</item_number>
  <description>Lethal Weapon</description>
  <part>
    <unit_price>17.95</unit_price>
    <upccode>85391628927</upccode>
  </part>
  <quantity>5</quantity>
  <total>89.75</total>
</items>
```

The generated columns include `item_number`, `description`, `part`, and each column will have only one row with the following values respectively ([1, 2], ["One Magic Christmas", "Lethal Weapon"], [{"unit_price":19.95, "upccode":13131092899}, {"unit_price":17.95, "upccode":85391628927}]) and so on for the remaining columns.

- Tags containing values and attributes are converted to object

```
<?xml version="1.0"?>
<catalog>
  <book id="bk102">
    <author country="ca">Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <publisher>John Doe</publisher>
  </book>
</catalog>
```

The `<author>` tag is converted to a column and will have an object as value that is structured as:

```
{
  "_": "Ralls, Kim",
  "country": "ca"
}
```

Note that the value of the tag has underscore ("`_`") as key and the attributes as "`attribute_name`": "`attribute_value`".

The Data Modeler Page

The Data Modeler page provides an integrated version of Oracle SQL Developer Data Modeler with basic reporting features. You can create diagrams from existing schemas, retrieve data dictionary information, generate DDL statements, and export diagrams.

To navigate to the Data Modeler page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **Data Modeler**.
- Click **Selector**  to display the navigation menu. Under Development, select **Data Modeler**.

The Data Modeler page consists of the left pane for navigating objects and diagrams, the editor pane for working with relational diagrams, and the right pane for viewing the properties of the object selected in the diagram. These panes are described in the following sections:

- [Navigating Diagrams and Objects](#)
- [About the Data Modeler Editor](#)
- [Navigating Diagrams and Objects](#)
The **Diagrams** tab lists the Data Modeler diagrams that have been saved.
- [About the Data Modeler Editor](#)
You can create and work with relational diagrams in the editor pane.

Related Topics

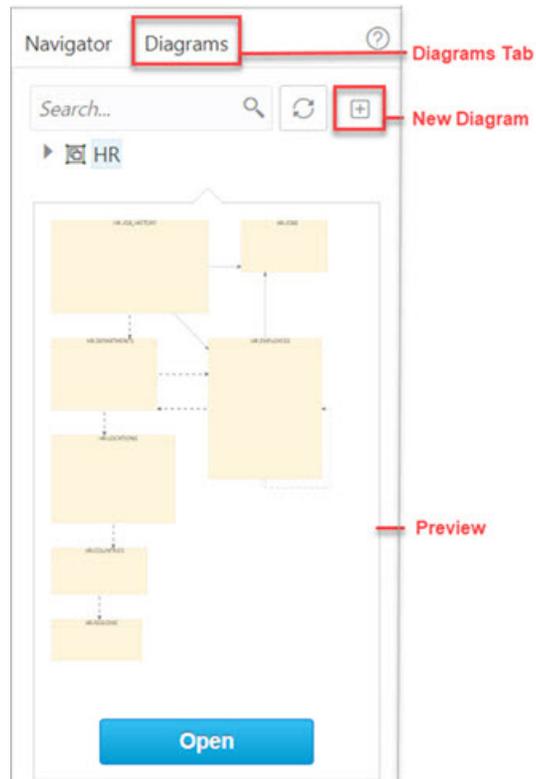
- *Oracle SQL Developer Data Modeler User's Guide*

Navigating Diagrams and Objects

The **Diagrams** tab lists the Data Modeler diagrams that have been saved.

When you right-click a diagram, you have options to open, save, delete, and view properties. [Figure 3-10](#) shows a thumbnail preview that is displayed when you click the name of a diagram in the left pane.

Figure 3-10 Diagram Preview in Data Modeler



- To create a new diagram, in the Diagrams tab, click New Diagram .
- To open an existing diagram, in the Diagrams tab, right-click the diagram and click **Open**.

Note

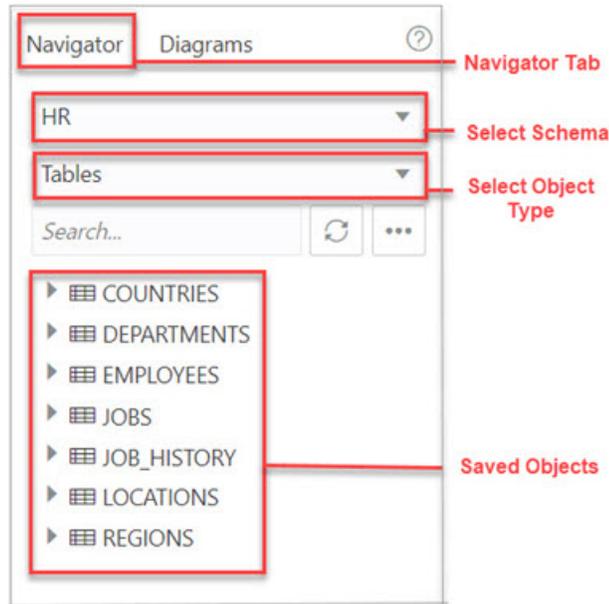
Diagrams are stored in the database, in the Database Actions user schema. When you use Data Modeler for the first time, a `OSDDMW_DIAGRAMS` table is created in your schema, to store the diagrams.

Oracle SQL Developer Data Modeler can import from or export to the `OSDDMW_DIAGRAMS` table if a connection is provided. For more information, see [Sharing Diagrams with SQL Developer Data Modeler](#).

The **Navigator** tab lists the objects that are available for each schema. You can select the schema and object type from the drop-down lists. For a particular schema, if a tables object is selected, the left pane displays all the tables that belong to the particular schema. You can expand a table to view its columns. [Figure 3-11](#) indicates the important elements in the Navigator tab.

In a schema, you can search for objects in the Navigator tab by entering a few characters. You can also search the contents of a saved diagram. The search functionality is not case-sensitive, retrieves all matching entries, and does not require the use of wildcard characters.

Figure 3-11 Navigator Tab



The context menu for a table or view consists of:

Add Object to Diagram: Adds the selected object to the selected diagram. Alternatively, you can drag and drop an object into a selected diagram.

Add Object with dependencies to Diagram: Adds the selected object along with parent and child tables related to the object to the selected diagram.

Add Object as Star Schema to Diagram: Adds the selected object to the diagram and searches the data dictionary for foreign keys and implied foreign keys related to the object. The related tables or views are added to the diagram and the star schema layout is applied. See [Implied Foreign Keys](#)

Edit, Use as Template: Opens the Table Properties Dialog. Use to edit an existing object (such as a table or view) for a specific schema, or create a new object by using an existing one for the initial content. See [The Table Properties Dialog](#) and [The View Properties Dialog](#).

Add/Edit Sequence: Opens the Sequence Properties Dialog. Use to create or edit a sequence for a selected schema. See [The Sequence Properties Dialog](#).

Add Index: Opens the Index Properties Dialog. Use to create an index for a table. See [The Index Properties Dialog](#).

Add Synonym: Opens the Synonym Properties Dialog. Use to create a synonym for a table or view in a selected schema. See [The Synonym Properties Dialog](#)

About the Data Modeler Editor

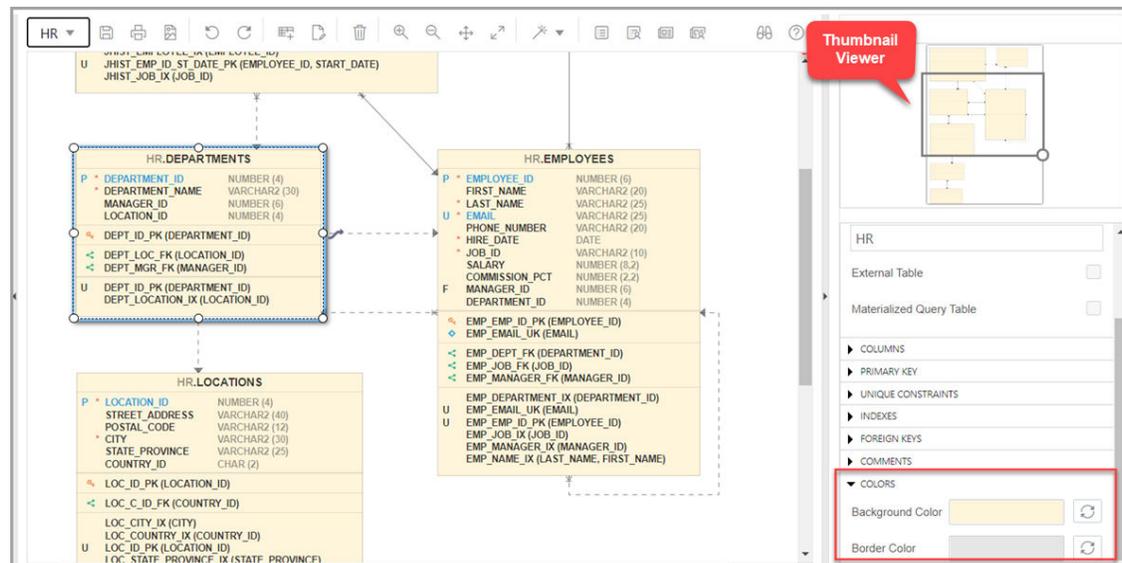
You can create and work with relational diagrams in the editor pane.

When an object is selected in the diagram, you can inspect the properties of the object in the right pane. For a table, the properties displayed are Columns, Primary Key, Unique Constraints, Indexes, Foreign Keys, Comments, and Colors. The only properties that you can edit are background and border color. Click Reset  to return to the default colors.

[Figure 3-12](#) shows the properties of an object selected in the editor pane.

You can drag the thumbnail viewer in the upper right corner to display the appropriate area in the editor pane. This is useful when you are working with long diagrams in the editor pane.

Figure 3-12 Inspect Properties of Object Selected in Editor Pane



When creating diagrams, you can do the following:

- Select an object and then drag to move it around.
- Adjust or move objects with the relationships intact.
- Add elbows to relationship lines to avoid intersecting with lines from other objects. Right-click the relationship line and drag to create the elbow. Click Remove vertex  to restore the original shape.
- Resize objects by dragging the handles that are positioned around the box.
- Select and then right-click an object for the following options:
 - **DDL Preview:** Shows the DDL statements for the object.
 - **Update:** Updates any actions performed on the object.
 - **Delete:** Deletes the object.
 - **Edit:** Opens the Table Properties Dialog for editing a table object or the View Properties Dialog for a view object.
 - **Implied Foreign Keys:** Opens the Implied Foreign Keys Dialog. Use this option to define implied foreign keys for the object. See [Implied Foreign Keys](#)
 - **View JSON Data Guides:** Opens the JSON data guide diagram for the table or view. See [About Viewing JSON Data Guides](#)

The icons on the toolbar are:

- **Save Diagram:** Saves the currently selected diagram. Diagrams are stored in a table that is created in the schema of the user.
- **Print Diagram:** Prints the selected diagram.

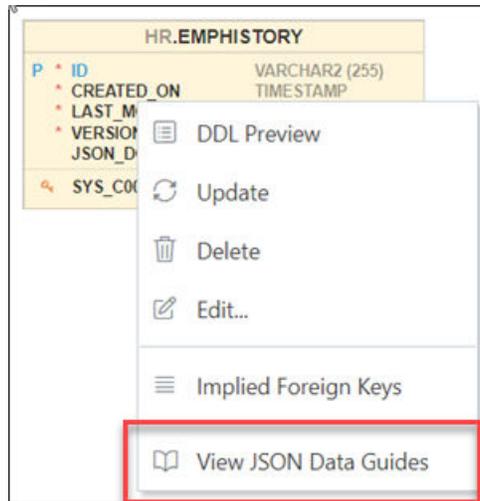
- **Save to SVG:** Saves the currently selected diagram to an image file in SVG format.
- **Add Objects to Diagram:** A dialog is displayed where you can select one or more objects of type tables or views from a specific schema into the selected diagram. Type * to list all the objects of an object type (tables or views) in the schema. You can also search by typing a few characters.
- **Add Note:** Adds notes to the selected diagram. Select the note to add text and to see the associated properties in the right pane, such as Font Size and Colors.
 - To enter text, expand TEXT and type information. Click **Apply**.
 - To select the background, border, or text color for the note, click the box and select the required color. To revert to the default color, click Reset .
- **Delete:** Deletes the selected object or objects from the diagram. To select multiple objects, press the CTRL key and select the objects.
- **Zoom In:** Displays more detail, and potentially fewer objects, in the currently selected diagram.
- **Zoom Out:** Displays less detail, and potentially more objects, in the currently selected diagram.
- **Fit Screen:** Makes all relevant objects fit in the window for the currently selected diagram, adjusting the sizes of shapes and text labels as needed.
- **Actual Size:** Adjusts the shapes and text labels in the currently selected diagram to the default sizes.
- **Layout: Auto Layout:** Rearranges the objects in the diagram to a layout that may be more meaningful and attractive. If you do not like the rearrangement, you can restore the previous layout by clicking Undo.
- **Layout: Star Layout:** Rearranges the objects in the diagram to a star schema layout, where the fact table is in the center and the associated dimension tables surround the fact table.
- **DDL Preview:** Shows the DDL statements for the object. You have the option to save or send the DDL statements to the worksheet. To choose what you want to display, click **Options**.
- **DDL Preview for Current Schema:** Shows the DDL statements for the current schema. You can send the DDL statements to the worksheet.
- **Diagram Report:** Generates a data dictionary report of everything in the diagram.
- **Schema Report:** Generates a data dictionary report of everything in the schema.
- **Help:** Displays the help for the Data Modeling editor.
- [About Viewing JSON Data Guides](#)

A JSON data guide represents the JSON schema for documents in one column with JSON content. A table can have more than one column with JSON content.

About Viewing JSON Data Guides

A JSON data guide represents the JSON schema for documents in one column with JSON content. A table can have more than one column with JSON content.

The View JSON Data Guides option is available in the context menu for a table or view in a Data Modeler diagram.



This is applicable for tables, views and external tables that have columns with JSON content. Columns with JSON content are identified and data guides are retrieved for each such column. This process is quicker with the existence of a JSON search index.

The JSON schema (JSON Data Guide) is presented visually like an entity-relationship diagram. Arrays are presented as one-to-many relationships, contained objects as one-to-one relationships, and "oneOf" constructs as a box that surrounds possible choices. There is a column selector at the top right part of the page enabling you to select a column with JSON content for diagram presentation.

The following example is a JSON schema and its representation in a data guide diagram.

```
{
  "type" : "object",
  "properties" :
  {
    "User" :
    {
      "type" : "string",
      "o:length" : 8,
      "o:preferred_column_name" : "DATA$User"
    },
    "PONumber" :
    {
      "type" : "number",
      "o:length" : 4,
      "o:preferred_column_name" : "DATA$PONumber"
    },
    "LineItems" :
    {
      "type" : "array",
      "o:length" : 1024,
      "o:preferred_column_name" : "DATA$LineItems",
      "items" :
      {
        "properties" :
        {
          "Part" :

```

```

    {
      "type" : "object",
      "o:length" : 256,
      "o:preferred_column_name" : "DATA$Part",
      "properties" :
      {
        "UPCCode" :
        {
          "type" : "number",
          "o:length" : 16,
          "o:preferred_column_name" : "DATA$UPCCode"
        },
        "UnitPrice" :
        {
          "type" : "number",
          "o:length" : 8,
          "o:preferred_column_name" : "DATA$UnitPrice"
        },
        "Description" :
        {
          "type" : "string",
          "o:length" : 128,
          "o:preferred_column_name" : "DATA$Description"
        }
      }
    },
    "Quantity" :
    {
      "type" : "number",
      "o:length" : 4,
      "o:preferred_column_name" : "DATA$Quantity"
    },
    "ItemNumber" :
    {
      "type" : "number",
      "o:length" : 1,
      "o:preferred_column_name" : "DATA$ItemNumber"
    }
  },
  "Reference" :
  {
    "type" : "string",
    "o:length" : 32,
    "o:preferred_column_name" : "DATA$Reference"
  },
  "Requestor" :
  {
    "type" : "string",
    "o:length" : 16,
    "o:preferred_column_name" : "DATA$Requestor"
  },
  "CostCenter" :
  {
    "type" : "string",

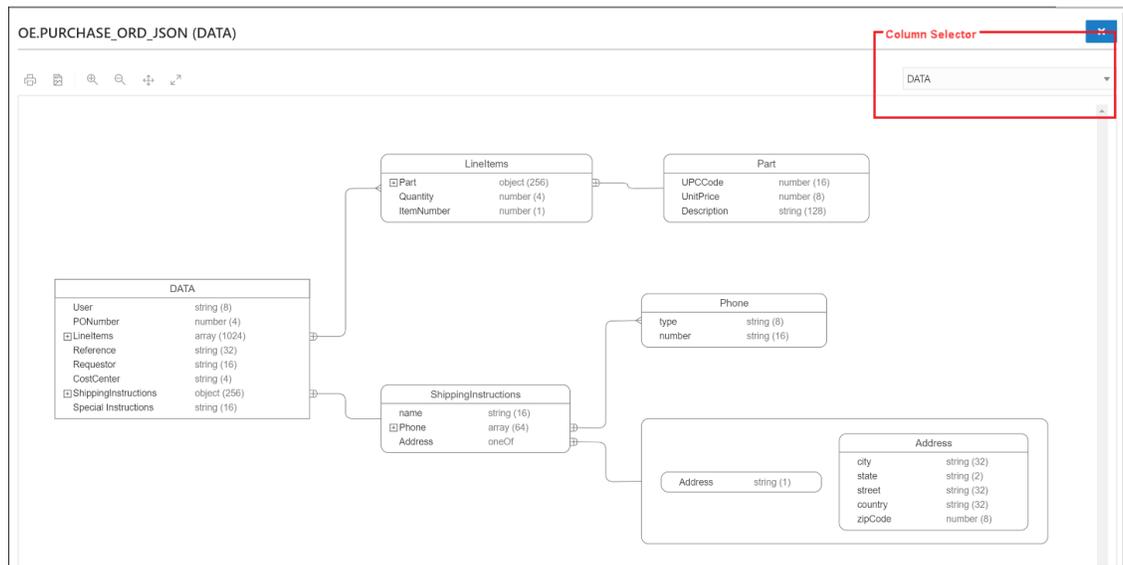
```

```

        "o:length" : 4,
        "o:preferred_column_name" : "DATA$CostCenter"
    },
    "ShippingInstructions" :
    {
        "type" : "object",
        "o:length" : 256,
        "o:preferred_column_name" : "DATA$ShippingInstructions",
        "properties" :
        {
            "name" :
            {
                "type" : "string",
                "o:length" : 16,
                "o:preferred_column_name" : "DATA$name"
            },
            "Phone" :
            {
                "type" : "array",
                "o:length" : 64,
                "o:preferred_column_name" : "DATA$Phone",
                "items" :
                {
                    "properties" :
                    {
                        "type" :
                        {
                            "type" : "string",
                            "o:length" : 8,
                            "o:preferred_column_name" : "DATA$type"
                        },
                        "number" :
                        {
                            "type" : "string",
                            "o:length" : 16,
                            "o:preferred_column_name" : "DATA$number"
                        }
                    }
                }
            }
        }
    },
    "Address" :
    {
        "oneOf" :
        [
            {
                "type" : "string",
                "o:length" : 1,
                "o:preferred_column_name" : "DATA$Address"
            },
            {
                "type" : "object",
                "o:length" : 128,
                "o:preferred_column_name" : "DATA$Address_1",
                "properties" :
                {
                    "city" :

```

```
{
  "type" : "string",
  "o:length" : 32,
  "o:preferred_column_name" : "DATA$city"
},
"state" :
{
  "type" : "string",
  "o:length" : 2,
  "o:preferred_column_name" : "DATA$state"
},
"street" :
{
  "type" : "string",
  "o:length" : 32,
  "o:preferred_column_name" : "DATA$street"
},
"country" :
{
  "type" : "string",
  "o:length" : 32,
  "o:preferred_column_name" : "DATA$country"
},
"zipCode" :
{
  "type" : "number",
  "o:length" : 8,
  "o:preferred_column_name" : "DATA$zipCode"
}
}
]
}
},
"Special Instructions" :
{
  "type" : "string",
  "o:length" : 16,
  "o:preferred_column_name" : "DATA$SpecialInstructions"
}
}
}
```

Figure 3-13 JSON Data Guide Diagram**See Also**

[JSON Data Guide in Oracle AI Database JSON Developer's Guide](#)

Creating and Editing Database Objects

You can create and edit objects using Create and Edit Object wizards available from the Navigator tab in the SQL and Data Modeler pages.

The wizards for creating and editing various object types are described in the following sections:

- [The Table Properties Dialog](#)
- [The Index Properties Dialog](#)
- [The Sequence Properties Dialog](#)
- [The View Properties Dialog](#)
- [The Synonym Properties Dialog](#)
- [Implied Foreign Keys](#)
- [The Materialized View Log Properties Dialog](#)

- [The Table Properties Dialog](#)

The Table Properties Dialog is displayed when you create a table, edit an existing table, or create a table using an existing one as a template.

- [The Index Properties Dialog](#)

The Index Properties dialog box is displayed when you create or edit an index.

- [The Sequence Properties Dialog](#)

The Sequence Properties Dialog is displayed when you create or edit a sequence.

- [The View Properties Dialog](#)
The View Properties Dialog is displayed when you create or edit a view.
- [The Synonym Properties Dialog](#)
The Synonym Properties Dialog is displayed when you create a synonym.
- [The Materialized View Log Properties Dialog](#)
The Materialized View Log Properties dialog is displayed when you create or edit a materialized view log, which is a table associated with the master table of a materialized view.
- [Implied Foreign Keys](#)
Implied foreign keys are dependencies that exist between tables but are not defined in the database. In a data warehouse environment, it is a common practice not to create foreign keys. However, it becomes necessary to show these dependencies for presentation or reporting purposes.

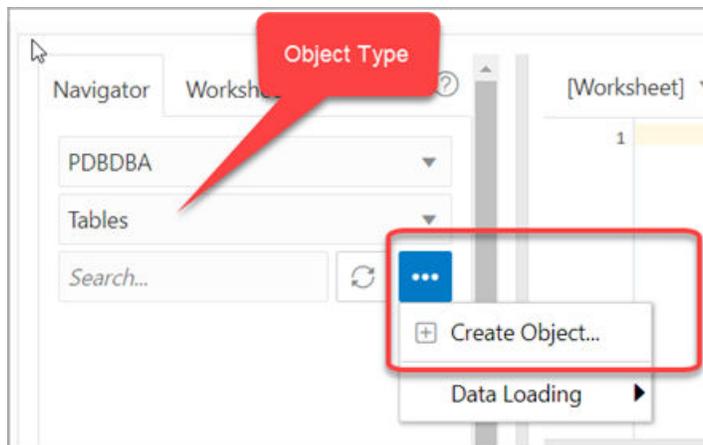
The Table Properties Dialog

The Table Properties Dialog is displayed when you create a table, edit an existing table, or create a table using an existing one as a template.

You can open the Table Properties dialog from the Navigator tab in SQL or Data Modeler.

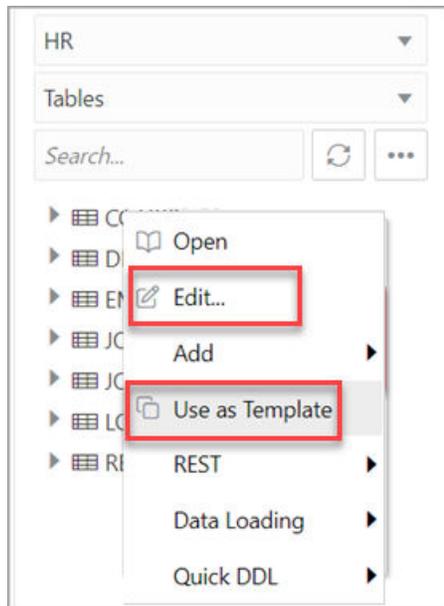
To create a table for a specific schema, in the Navigator tab, select **Tables** from the object type drop-down list, click **Object submenu** **...**, and select **Create Object**.

Figure 3-14 Create Object



To create a table from an existing one for a specific schema, right-click the table object in the Navigator tab, and select **Use as Template** as shown in [Figure 3-15](#).

To edit a table for a specific schema, right-click a table object in the Navigator tab, and select **Edit**.

Figure 3-15 Edit Option for Existing Table

The table properties are grouped in several panes.

If you are editing an existing table, you can visit the panes in any order. If you click Create before you finish creating the table, right-click the table name, select **Edit**, and continue creating the table.

Note

Editing a partitioned table is not recommended. To identify whether a table is partitioned or not, right-click the table name and select **Edit**. If the table is partitioned, a warning message will be displayed.

Schema: Database schema in which to create the table. By default, a new table is created in the existing schema or the schema that you are logged into.

Name: Name for the table.

The different panes in the dialog are described in the following sections:

- [Columns Pane](#)
- [Primary Key Pane](#)
- [Unique Keys Pane](#)
- [Indexes Pane](#)
- [Foreign Keys Pane](#)
- [Table Constraints Pane](#)
- [Comments Pane](#)
- [Storage Pane](#)
- [External Table Properties Pane](#)

- [Materialized View Pane](#)
- [DDL Pane](#)
- [Output Pane](#)
- [Columns Pane](#)
Specifies properties for each column in the table.
- [Primary Key Pane](#)
Specifies the primary key for the table.
- [Unique Keys Pane](#)
Specifies one or more unique constraints for the table.
- [Indexes Pane](#)
Lists the indexes defined for the table.
- [Foreign Keys Pane](#)
Specifies one or more foreign keys for the table.
- [Table Constraints Pane](#)
Specifies one or more check constraints for the table.
- [Comments Pane](#)
- [Storage Pane](#)
Enables you to specify storage options for the table.
- [External Table Properties Pane](#)
Specifies options for an external table.
- [Materialized View Pane](#)
Specifies options for a materialized view.
- [DDL Pane](#)
- [Output Pane](#)

Related Topics

- *Oracle AI Database SQL Language Reference*

Columns Pane

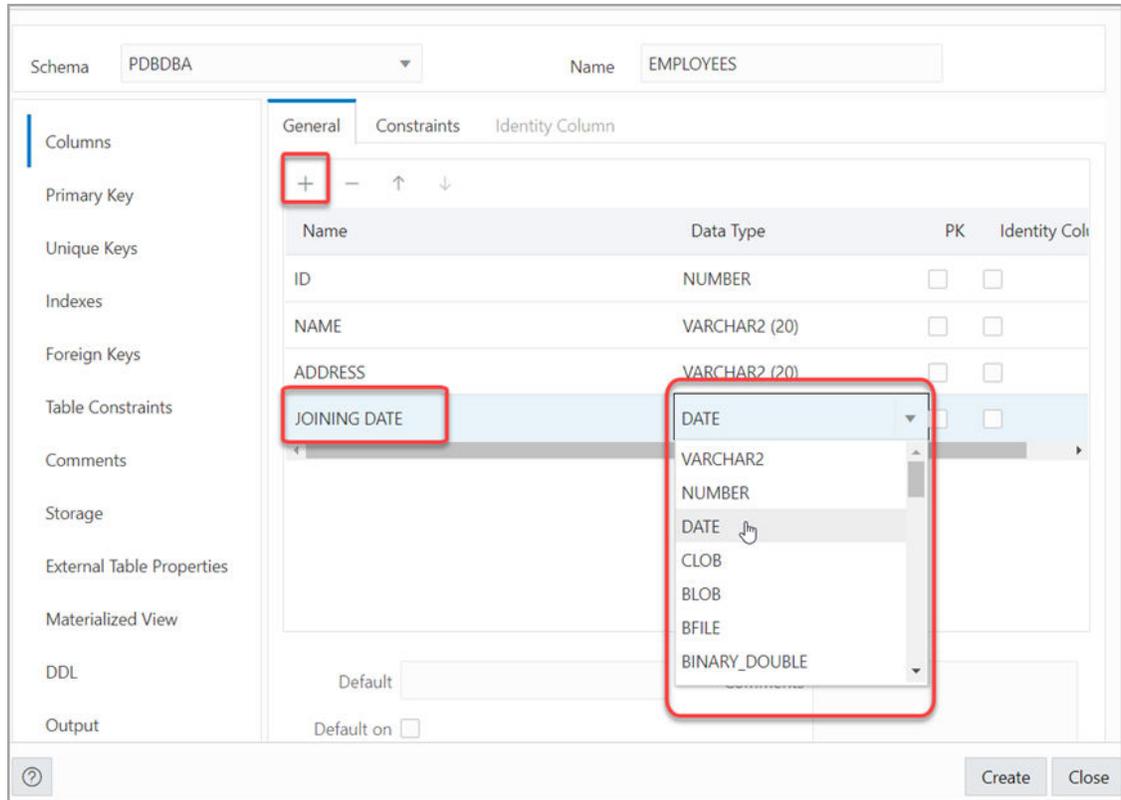
Specifies properties for each column in the table.

General tab

Lists the columns available in the table.

To add a column, click **Add Column** (+). A new row is added to the table below. Select the row and enter the details for the column.

To delete a column, select the row and click **Remove Column** (-). To move a column up or down in the table, select it and use the up-arrow and down-arrow icons.

Figure 3-16 Add a Column

- **Name:** Name for the column.
- **Datatype:** Data type for the column.
- **Default:** If no value is specified, the default value inserted into the column when a row is inserted.
- **Default on NULL:** Applicable for Oracle Database 12c and later releases. If this option is selected, when a row is inserted into the table and the value specified for the column is NULL, the default value is inserted into the column.
- **Expression:** Expression for computing the value in the column.
- **Comments:** Optional descriptive comments about the column. Use this field to provide descriptions for the attributes.

In the table:

- **PK:** If this option is selected, the column becomes the primary key.
- **Identity Column:** If this option is selected, the column becomes an identity column. This is applicable only for Oracle Database 12c and later releases. For more details, see the Identity Column tab.

Constraints tab

Displays the Not Null and Check Constraints for a column. A check constraint requires values in a column to comply with a specified condition.

- **Not Null Constraint: Name:** Name for the Not Null constraint.

- **Not Null Constraint: Not Null:** If this option is selected, the column must contain data. You cannot specify no value or an explicit null value for this column when you insert a row. If this option is not checked, the column can contain either data or no data. A primary key column cannot be null.
- **Check Constraint: Name:** Name for the check constraint definition.
- **Check Constraint: Constraint:** Condition that must be met for a column to fulfill the check constraint. You can use any valid CHECK clause (without the CHECK keyword). For example, to indicate that the value in a numeric column named RATING must be from 1 to 10, you can specify: rating >=1 and rating <= 10.
- **Enabled:** If this option is selected, the constraint is checked when data is entered or updated in the column.
- **Deferrable:** If this option is selected, you can defer checking the validity of the constraint until the end of a transaction.
- **Initially Immediate:** If this option is selected, the constraint is checked whenever you add, update, or delete data from the column.
- **Validate:** If this option is selected, the existing data is checked to see if it conforms to the constraint.

Identity Column tab

Applicable for Oracle Database 12c and later releases. The Identity Column tab lists the properties of the identity column. This tab becomes available only after the Identity Column checkbox is selected for the column in the General tab. An identity column is an autoincrement column that can be used to identify a table row. Only one identity column can be specified for a table.

- **Generate:** Always means that values cannot be explicitly included for the identity column in INSERT OR UPDATE statements, By Default means values for the identity column are generated automatically if no values are specified explicitly, By Default on Null means values are generated for the column only when a NULL value is supplied.
- **Start with:** Starting value of the sequence.
- **Increment:** Interval between successive numbers in a sequence.
- **Min value:** Lowest possible value for the sequence. The default is 1 for an ascending sequence and $-(10^{26})$ for a descending sequence.
- **Max value:** Highest possible value for the sequence. The default is 10^{27} for an ascending sequence and -1 for a descending sequence.
- **Cache and Cache size:** Cache causes sequence values to be preallocated in cache, which can improve application performance; Cache size indicates the number of sequence values preallocated in cache. No Cache causes sequence values not to be preallocated in cache.
- **Cycle:** Indicates whether the sequence "wraps around" to reuse numbers after reaching its maximum value (for an ascending sequence) or its minimum value (for a descending sequence). If cycling of values is not enabled, the sequence cannot generate more values after reaching its maximum or minimum value.
- **Order:** Indicates whether sequence numbers are generated in the order in which they are requested. If No Order is specified, sequence numbers are not guaranteed to be in the order in which they were requested.

Primary Key Pane

Specifies the primary key for the table.

The primary key is the column, or set of columns, that uniquely identifies each row in the table. If the Primary Key checkbox is selected for a column in the General tab, the corresponding fields are automatically populated in the Primary Key pane. You can make changes to the properties as required.

An index is automatically created on the primary key.

- **Name:** Name of the constraint to be associated with the primary key definition.
- **Enabled:** If this option is checked, the primary key constraint is enforced: that is, the data in the primary key column (or set of columns) must be unique and not null.
- **Index:** Name of the index to which the primary key refers.
- **Tablespace:** Name of the tablespace associated with the index.
- **Available Columns:** Lists the columns that are available to be added to the primary key definition. You can select multiple attributes, if required, for the primary key.
- **Selected Columns:** Lists the columns that are included in the primary key definition.

To add a column to the primary key definition, select it in Available Columns and click the Add (>) icon; to remove a column from the primary key definition, select it in Selected Columns and click the Remove (<) icon. To move all columns from available to selected (or the reverse), use the Add All (>>) or Remove All (<<) icon. To move a column up or down in the primary key definition, select it in Selected Columns and use the arrow buttons.

Unique Keys Pane

Specifies one or more unique constraints for the table.

A unique constraint specifies a column, or set of columns, whose data values must be unique: each data value must not be null, and it must not be the same as any other value in the column.

To add a unique constraint, click the Add button; to delete a unique constraint, select it and click the Remove button.

- **Name:** Name of the unique constraint.
- **Enabled:** If this option is selected, the unique constraint is enforced.
- **Rely:** If this option is selected, the constraint in NOVALIDATE mode is taken into account during query rewrite.
- **Deferrable:** If this option is selected, in subsequent transactions, constraint checking can be deferred until the end of the transaction using the SET CONSTRAINT(S) statement.
- **Initially Immediate:** If this option is selected, the constraint is checked at the end of each subsequent SQL statement.
- **Validate:** If the option is selected, the existing data is checked to see if it conforms to the constraint.
- **Index:** Name of the index to which the unique key refers.
- **Tablespace:** Name of the tablespace associated with the index.
- **Available Columns:** Lists the columns that are available to be added to the unique constraint definition.

- **Selected Columns:** Lists the columns that are included in the unique constraint definition.

To add a column to the unique constraint definition, select it in Available Columns and click the Add (➤) icon; to remove a column from the unique constraint definition, select it in Selected Columns and click the Remove (◀) icon. To move all columns from available to selected (or the reverse), use the Add All (➤➤) or Remove All (◀◀) icon. To move a column up or down in the unique constraint definition, select it in Selected Columns and use the arrow buttons.

Indexes Pane

Lists the indexes defined for the table.

To add an index, click Add Index (+); to delete an index, select it and click Remove Index (-).

- **Name:** Name of the index.
- **Type:** The type of Oracle index. *Non-unique* means that the index can contain multiple identical values; *Unique* means that no duplicate values are permitted; *Bitmap* stores rowids associated with a key value as a bitmap.
- **Tablespace:** Name of the tablespace for the index.
- **Expression:** A column expression is an expression built from columns, constants, SQL functions, and user-defined functions. When you specify a column expression, you create a function-based index.
- **Available Columns and Selected Columns:** Columns selected for the index. To select a column, click the column in the Available Columns box, and then click the Add Selected Columns icon to move it to the Selected Columns box.

Foreign Keys Pane

Specifies one or more foreign keys for the table.

A foreign key specifies a column ("local column"), whose data values match values in the primary key or unique constraint of another table.

- **Name:** Name of the foreign key definition.
- **Enabled:** If this option is checked, the foreign key is enforced.
- **Rely, Deferrable, Initially Immediate, Validate:** See the description of these fields in the Unique Keys pane.
- **Referenced Constraint: Schema:** Name of the schema containing the table with the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: Table:** Name of the table with the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: Constraint:** Name of the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: On Delete:** Action to take automatically when a row in the referenced table is deleted and rows with that value exist in the table containing this foreign key: *NO ACTION* (shown by a crossing line in diagrams) performs no action on these rows; *CASCADE* (shown by an "X") deletes these rows; *SET NULL* (shown by a small circle) sets null all columns in those rows that can be set to a null value.
- **Associations: Local Column:** Lists the column in the currently selected (local) table that is included in the foreign key definition. For each referenced column in the foreign key definition, select the name of a column in the edited table.

- **Associations: Referenced Column:** For each local column, identifies the column in the other (foreign) table that must have a value matching the value in the local column.

Table Constraints Pane

Specifies one or more check constraints for the table.

A check constraint specifies a condition that must be met when a row is inserted into the table or when an existing row is modified.

- **Name:** Name of the check constraint definition.
- **Check Condition:** Condition that must be met for a row to fulfil the check constraint. You can use any valid CHECK clause (without the CHECK keyword). For example, to indicate that the value in a numeric column named RATING must be from 1 to 10, you can specify rating >=1 and rating <= 10.
- **Enabled:** If this option is checked, the check constraint is enforced.

Comments Pane

Enter descriptive comments in this pane. This is optional.

Storage Pane

Enables you to specify storage options for the table.

When you create or edit a table or an index, you can override the default storage options.

- **Organization:** Specifies that the table is stored and organized with (Index) or without an index (Heap) or as an external table (External).
- **Tablespace:** Name of the tablespace for the table or index.
- **Logging:** ON means that the table creation and any subsequent INSERT operations against the table are logged in the redo log file. OFF means that these operations are not logged in the redo log file.
- **Row Archival:** YES enables in-database archiving, which allows you to archive rows within the table by marking them as invisible.

External Table Properties Pane

Specifies options for an external table.

An external table is a read-only table whose metadata is stored in the database but whose data is stored outside the database.

External Table

- **Access Driver Type:** Specifies the type of external table.
 - ORACLE_LOADER: Extracts data from text data files. This is the default access driver, which loads data from external tables to internal tables.
 - ORACLE_DATAPUMP: Extracts data from binary dump files. This access driver can perform both loads and unloads.
 - ORACLE_BIGDATA: Extracts data from Oracle Big Data Appliance.
 - ORACLE_HDFS: Extracts data stored in a Hadoop Distributed File System (HDFS).

- ORACLE_HIVE: Extracts data stored in Apache HIVE.
- **Default Directory:** Specifies the default directory to use for all input and output files that do not explicitly name a directory object. The location is specified with a directory object, not a directory path.
- **Access Params:** Assigns values to the parameters of the specific access driver for the external table. Access parameters are optional.
 - OPAQUE_FORMAT_SPEC: The `opaque_format_spec` specifies all access parameters for the ORACLE_LOADER, ORACLE_DATAPUMP, ORACLE_HDFS, and ORACLE_HIVE access drivers. For descriptions of the access parameters, see *Oracle AI Database Utilities*. Field names specified in the `opaque_format_spec` must match columns in the table definition, else Oracle Database ignores them.
 - USING CLOB: Enables you to derive the parameters and their values through a subquery. The subquery cannot contain any set operators or an ORDER BY clause. It must return one row containing a single item of data type CLOB.
- **Reject Limit:** The number of conversion errors that can occur during a query of the external data before an Oracle Database error is returned and the query is aborted.
- **Project Column:** Determines how the access driver validates the rows of an external table in subsequent queries.
 - ALL: Processes all column values, regardless of which columns are selected, and validates only those rows with fully valid column entries. If any column value raises an error, such as a data type conversion error, the row is rejected even if that column was not referenced in the select list of the query.
 - REFERENCED: Processes only those columns in the select list of the query.

The ALL setting guarantees consistent result sets. The REFERENCED setting can result in different numbers of rows returned, depending on the columns referenced in subsequent queries, but is faster than the ALL setting. If a subsequent query selects all columns of the external table, then the settings behave identically.

- **Location:** Specifies the data files for the external table. Use the Add (+) icon to add each location specification.
 - For ORACLE_LOADER and ORACLE_DATAPUMP, the files are named in the form `directory:file`. The directory portion is optional. If it is missing, then the default directory is used as the directory for the file. If you are using the ORACLE_LOADER access driver, then you can use wildcards in the file name. An asterisk (*) signifies multiple characters and a question mark (?) signifies a single character.
 - For ORACLE_HDFS, LOCATION is a list of Uniform Resource Identifiers (URIs) for a directory or for a file. There is no directory object associated with a URI.
 - For ORACLE_HIVE, LOCATION is not used. Instead, the Hadoop HCatalog table is read to obtain information about the location of the data source (which could be a file or another database).

Opaque Format Spec

Specifies all access parameters for the ORACLE_LOADER, ORACLE_DATAPUMP, ORACLE_HDFS, and ORACLE_HIVE access drivers.

For example:

```
RECORDS DELIMITED BY NEWLINE CHARACTERSET US7ASCII
TERRITORY AMERICA
BADFILE log_file_dir:'ext_lv3.bad'
```

```

LOGFILE log_file_dir:'ext_lv3.log'
FIELDS TERMINATED BY "|" OPTIONALLY ENCLOSED BY '^' LDRTRIM
( PROD_ID,
  CUST_ID ,
  TIME_ID DATE(10) "YYYY-MM-DD",
  CHANNEL_ID ,
  PROMO_ID ,
  QUANTITY_SOLD ,
  AMOUNT_SOLD ,
  UNIT_COST ,
  UNIT_PRICE
)

```

and the full statement:

```

CREATE TABLE SH.SALES_TRANSACTIONS_EXT
(
  PROD_ID NUMBER ,
  CUST_ID NUMBER ,
  TIME_ID DATE ,
  CHANNEL_ID NUMBER ,
  PROMO_ID NUMBER ,
  QUANTITY_SOLD NUMBER ,
  AMOUNT_SOLD NUMBER (10,2) ,
  UNIT_COST NUMBER (10,2) ,
  UNIT_PRICE NUMBER (10,2)
)
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY DATA_FILE_DIR
  ACCESS PARAMETERS
  (
    RECORDS DELIMITED BY NEWLINE CHARACTERSET US7ASCII
    TERRITORY AMERICA
    BADFILE log_file_dir:'ext_lv3.bad'
    LOGFILE log_file_dir:'ext_lv3.log'
    FIELDS TERMINATED BY "|" OPTIONALLY ENCLOSED BY '^' LDRTRIM
    ( PROD_ID ,
      CUST_ID ,
      TIME_ID DATE(10) "YYYY-MM-DD",
      CHANNEL_ID ,
      PROMO_ID ,
      QUANTITY_SOLD ,
      AMOUNT_SOLD ,
      UNIT_COST ,
      UNIT_PRICE
    )
  )
)
LOCATION ( "DATA_FILE_DIR":'salelv3.dat' )
)
REJECT LIMIT 100
;

```

CLOB Subquery

Type or copy and paste the query.

Note

For more information about the external table fields, see *Oracle AI Database Utilities* and *Oracle AI Database SQL Language Reference*

Materialized View Pane

Specifies options for a materialized view.

Query: Contains the SQL code for the query part of the view definition. Type or copy and paste the query.

General

- **On Pre-built Table:** If **Yes**, an existing table is registered as a preinitialized materialized view. This option is particularly useful for registering large materialized views in a data warehousing environment. The table must have the same name and be in the same schema as the resulting materialized view, and the table should reflect the materialization of a subquery.
- **Reduced Precision:** **Yes** authorizes the loss of precision that will result if the precision of the table or materialized view columns do not exactly match the precision returned by the subquery. If **No**, the precision of the table or materialized view columns must exactly match the precision returned by the subquery, or the create operation will fail.
- **For Update:** Select **Yes** to allow a subquery, primary key, object, or rowid materialized view to be updated. When used in conjunction with Advanced Replication, these updates will be propagated to the master.
- **Real Time MV:** Select **Yes** to create a real-time materialized view or a regular view. A real-time materialized view provides fresh data to user queries even when the materialized view is not in sync with its base tables due to data changes. Instead of modifying the materialized view, the optimizer writes a query that combines the existing rows in the materialized view with changes recorded in log files (either materialized view logs or the direct loader logs). This is called on-query computation.
- **Query Rewrite:** If **Enable**, the materialized view is enabled for query rewrite, which transforms a user request written in terms of master tables into a semantically equivalent request that includes one or more materialized views.
- **Build:** Specifies when to populate the materialized view. **Immediate** indicates that the materialized view is to be populated immediately. **Deferred** indicates that the materialized view is to be populated by the next refresh operation. If you specify **Deferred**, the first (deferred) refresh must always be a complete refresh; until then, the materialized view has a staleness value of unusable, so it cannot be used for query rewrite.
- **Use Index:** If **Yes**, a default index is created and used to speed up incremental (fast) refresh of the materialized view. If **No**, this default index is not created. (For example, you might choose to suppress the index creation now and to create such an index explicitly later.)
- **Index Tablespace:** Specifies the tablespace in which the materialized view is to be created. If a tablespace is not selected, the materialized view is created in the default tablespace of the schema containing the materialized view.

- **Cache:** If **Yes**, the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This setting is useful for small lookup tables. If **No**, the blocks are placed at the least recently used end of the LRU list.

Refresh Clause

- **Refresh:** Select Yes to enable refresh operations.
- **Refresh Type:** The method of refresh operation to be performed:
 - Complete Refresh: Executes the defining query of the materialized view, even if a fast refresh is possible.
 - Fast Refresh: Uses the incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables. The changes for conventional DML changes are stored in the materialized view log associated with the master table. The changes for direct-path INSERT operations are stored in the direct loader log.
 - Force Refresh: Performs a fast refresh if one is possible; otherwise, performs a complete refresh.
- **Action:** The type of refresh operation to be performed:
 - On Demand: Performs a refresh when one of the DBMS_MVIEW refresh procedures are called.
 - On Commit: Performs a fast refresh whenever the database commits a transaction that operates on a master table of the materialized view. This may increase the time taken to complete the commit, because the database performs the refresh operation as part of the commit process.
 - Specify: Performs refresh operations according to what you specify in the Start on and Next fields.
- **Start Date:** Starting date and time for the first automatic refresh operation. Must be in the future.
- **Next Date:** Time for the next automatic refresh operation. The interval between the Start on and Next times establishes the interval for subsequent automatic refresh operations. If you do not specify a value, the refresh operation is performed only once at the time specified for Start on.
- **With:** Refresh type, which determines the type of materialized view:
 - Primary Key: Creates a primary key materialized view, which allows materialized view master tables to be reorganized without affecting the eligibility of the materialized view for fast refresh.
 - Row ID: Creates a rowid materialized view, which is useful if the materialized view does not include all primary key columns of the master tables.
- **Default Storage:** If Yes, DEFAULT specifies that Oracle Database will choose automatically which rollback segment to use. If you specify DEFAULT, you cannot specify the rollback_segment. DEFAULT is most useful when modifying, rather than creating, a materialized view.
- **Storage Type:** MASTER specifies the remote rollback segment to be used at the remote master site for the individual materialized view. LOCAL specifies the remote rollback segment to be used for the local refresh group that contains the materialized view. This is the default.
- **Rollback Segment:** Enter the name of the rollback segment.

- **Using Constraint:** If this option is checked, more rewrite alternatives can be used during the refresh operation, resulting in more efficient refresh execution. The behavior of this option is affected by whether you select Enforced or Trusted.
 - **Enforced:** Causes only enforced constraints to be used during the refresh operation.
 - **Trusted:** Enables the use of dimension and constraint information that has been declared trustworthy by the database administrator but that has not been validated by the database. If the dimension and constraint information is valid, performance may improve. However, if this information is invalid, then the refresh procedure may corrupt the materialized view even though it returns a success status.

DDL Pane

You can review and save the SQL statements that are generated when creating or editing the object. If you want to make any changes, go back to the relevant panes and make the changes there.

- For a new table, click **CREATE** to view the generated DDL statements.
- When you edit table properties, click **UPDATE** to view the generated ALTER statements. For a new table, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output Pane

Displays the results of the DDL commands. If there are any errors, go to the appropriate pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

The Index Properties Dialog

The Index Properties dialog box is displayed when you create or edit an index.

To create an index for a selected schema, in SQL, in the Navigator tab, select **Indexes** from the object type drop-down list, click Object submenu *******, and select **Create Object**.

To edit an index for a selected schema, right-click a table object in the Navigator tab, and select **Edit**.

Definition pane

- **Schema:** Database schema that owns the table associated with the index.
- **Table:** Name of the table associated with the index.
- **Schema:** Database in which to create the index.
- **Tablespace:** Tablespace for the index.
- **Name:** Name of the index.
- **Type:** The type of Oracle index.
 - **Non-unique** means that the index can contain multiple identical values.
 - **Unique** means that no duplicate values are permitted.
 - **Bitmap** stores rowids associated with a key value as a bitmap.

- **Expression:** A column name or column expression. A column expression is an expression built from columns, constants, SQL functions, and user-defined functions. When you specify a column expression, you create a function-based index.
- **Available Columns:** Columns available in the table.
- **Selected Columns:** Columns selected for the index. Click Add Selected Columns > to move columns from the Available Columns list.
- **Order:** **ASC** for an ascending index (index values sorted in ascending order); **DESC** for a descending index (index values sorted in descending order).

DDL pane

You can review and save the SQL statements that are generated when creating or editing the index. If you want to make any changes, go back to the Definition pane and make the changes there.

- For a new index, click **CREATE** to view the generated DDL statements.
- When you edit index properties, click **UPDATE** to view the generated ALTER statements. For a new index, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output pane

Displays the results of the DDL commands. If there are any errors, go to the Definition pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

The Sequence Properties Dialog

The Sequence Properties Dialog is displayed when you create or edit a sequence.

To create a sequence for a selected schema, in SQL, in the Navigator tab, select **Sequences** from the object type drop-down list, click Object submenu *******, and select **Create Object**.

To edit a sequence for a selected schema, right-click a sequence object in the Navigator tab and select **Edit**.

A sequence is an object from which multiple users may generate unique integers. You can use sequences to automatically generate primary key values.

Properties pane

- **Schema:** Database schema in which to create the sequence.
- **Name:** Name of the sequence.
- **Start with:** Starting value of the sequence.
- **Increment:** Interval between successive numbers in a sequence.
- **Min value:** Lowest possible value for the sequence. The default is 1 for an ascending sequence and $-(10^{26})$ for a descending sequence.
- **Max value:** Highest possible value for the sequence. The default is 10^{27} for an ascending sequence and -1 for a descending sequence.
- **Cache and Cache size:** **Cache** causes sequence values to be preallocated in cache, which can improve application performance; **Cache size** indicates the number of sequence values preallocated in cache. **No Cache** causes sequence values not to be preallocated in cache.

- **Cycle:** Indicates whether the sequence "wraps around" to reuse numbers after reaching its maximum value (for an ascending sequence) or its minimum value (for a descending sequence). If cycling of values is not enabled, the sequence cannot generate more values after reaching its maximum or minimum value.
- **Order:** Indicates whether sequence numbers are generated in the order in which they are requested. If **No Order** is specified, sequence numbers are not guaranteed to be in the order in which they were requested.

DDL pane

You can review and save the SQL statements that are generated when creating or editing the sequence. If you want to make any changes, go back to the Properties pane and make the changes there.

- For a new sequence, click **CREATE** to view the generated DDL statements.
- When you edit a sequence, click **UPDATE** to view the generated ALTER statements. For a new sequence, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output pane

Displays the results of the DDL commands. If there are any errors, go to the Properties pane, fix the errors, and run the commands again. You have save to a text file or clear the output.

The View Properties Dialog

The View Properties Dialog is displayed when you create or edit a view.

You can open the View Properties Dialog from the Navigator tab in SQL or Data Modeler.

To create a view for a selected schema, in SQL, in the Navigator tab, select **Views** from the object type drop-down list, click Object submenu ******* , and select **Create Object**.

To create a view from an existing template for a selected schema, in the Navigator tab, select the view to create from, right-click and select **Use as Template**.

To edit a view for a selected schema, right-click a view object in the Navigator pane, and select **Edit**.

Schema: Database schema in which to create the view.

Name: Name of the view.

The different panes in the dialog are described in the following sections:

SQL Query pane

Enter or copy and paste the SQL query for the view, using the SELECT and FROM keywords along with the syntax needed to retrieve the desired information. A semicolon is not required after the query.

Columns pane

Click Refresh Columns  to automatically populate the columns in this pane. You can edit the columns by selecting the required row and making changes in the Header Alias and Comments fields.

Storage pane

- **Force on Create:** Select **Yes** to create the view regardless of whether the base tables of the view or the referenced object types exist or the owner of the schema containing the view has privileges on them. These conditions must be true before any SELECT, INSERT, UPDATE, or DELETE statements can be issued against the view. If the view definition contains any constraints, CREATE VIEW ... FORCE fails if the base table does not exist or the referenced object type does not exist. CREATE VIEW ... FORCE also fails if the view definition names a constraint that does not exist.
- **Query Restriction: Read Only** prevents the view from being used to add, delete, or change data in the underlying table. **Check Option** prohibits any changes to the underlying table that would produce rows that are not included in this view.

Use the Primary Key, Unique Keys, Foreign Keys, and Comments panes to add or edit properties as required.

DDL pane

Based on the inputs provided, the DDL statements are generated. You can review and save the SQL statements. If you want to make any changes, go back to the relevant pane and make the changes there.

- For a new view, click **CREATE** to view the generated DDL statements.
- When you edit a view, click **UPDATE** to view the generated ALTER statements. For a new view, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output pane

Displays the results of the DDL commands. If there are any errors, go to the respective pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

The Synonym Properties Dialog

The Synonym Properties Dialog is displayed when you create a synonym.

There are two ways of creating a synonym for a selected schema:

- In SQL, in the Navigator tab, right-click the object for which you want to create the synonym, and select **Add Synonym**. In this case, the only fields that you can edit in the Properties pane are Public and Synonym Name. The values of the remaining fields are predetermined by the object selected.
- In SQL, in the Navigator tab, select the object type as **Synonyms** or **Public Synonyms** from the drop-down list. Click Object submenu *******, and select **Create Object**. All the fields in the Properties dialog are available for edit.

The different panes in the dialog are described in the following sections:

Properties pane

- **Public:** If this option is checked, the synonym is accessible to all users. However, each user must have appropriate privileges on the underlying object to use the synonym. If this option is not checked, the synonym is a private synonym, and is accessible only within its schema.
- **Synonym Schema:** Database schema in which to create the synonym.
- **Synonym Name:** Name of the synonym. A private synonym must be unique within its schema; a public synonym must be unique within the database.
- **Object Type:** Specify the type of object to which this synonym refers.

- **Object Schema:** Schema containing the object or name to which this synonym refers.
- **DB Filter:** After selecting the Object Type and Object Schema, the list of objects of the selected type may be very long. To filter the object names, enter the search entry and click Refresh . The Object Name field is auto-filled with appropriate object names in the drop-down list.
- **Object Name:** Select the name of the object to which this synonym refers.
- **DB Link:** Enter a complete or partial database link to create a synonym for a schema object on a remote database where the object is located. If you specify DB Link and omit schema, then the synonym refers to an object in the schema specified by the database link. Oracle recommends that you specify the schema containing the object in the remote database. If you omit DB Link, then Oracle Database assumes the object is located on the local database.

DDL pane

Based on the inputs provided, the DDL statements are generated. You can review and save the SQL statements. If you want to make any changes, go back to the relevant pane and make the changes there.

- For a new view, click **CREATE** to view the generated DDL statements.
- When you edit a view, click **UPDATE** to view the generated ALTER statements. For a new view, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output pane

Displays the results of the DDL commands. If there are any errors, go to the respective pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

The Materialized View Log Properties Dialog

The Materialized View Log Properties dialog is displayed when you create or edit a materialized view log, which is a table associated with the master table of a materialized view.

To create a materialized view log for a selected schema, in SQL, in the Navigator tab, select **Materialized View Logs** from the object type drop-down list, click Object submenu , and select **Create Object**.

To edit, right-click a materialized view log object in the Navigator pane and select **Edit**.

Schema: Database schema in which to create the materialized view log.

Table: Name of the master table of the materialized view to be associated with this materialized view log.

Properties tab

- **Row ID Logged:** **Yes** indicates that the rowid of all rows changed should be recorded in the materialized view log. **No** indicates that the rowid of all rows changed should not be recorded in the materialized view log.
- **PK Logged:** **Yes** indicates that the primary key of all rows changed should be recorded in the materialized view log; **No** indicates that the primary key of all rows changed should not be recorded in the materialized view log.
- **New values:** **Yes** saves both old and new values for update DML operations in the materialized view log; **No** disables the recording of new values in the materialized view log.

If this log is for a table on which you have a single-table materialized aggregate view, and if you want the materialized view to be eligible for fast refresh, you must specify **Yes**.

- **Object ID Logged:** For a log on an object table only: **Yes** indicates that the system-generated or user-defined object identifier of every modified row should be recorded in the materialized view log. **No** indicates that the system-generated or user-defined object identifier of every modified row should not be recorded in the materialized view log.
- **Cache:** For data that will be accessed frequently, **CACHE** specifies that the blocks retrieved for this log are placed at the most recently used end of the least recently used list in the buffer cache when a full table scan is performed. This attribute is useful for small lookup tables. **NOCACHE** specifies that the blocks are placed at the least recently used end of the LRU list.
- **Parallel:** If **YES**, parallel operations will be supported for the materialized view log.
- **Sequence Logged:** **Yes** indicates that a sequence value providing additional ordering information should be recorded in the materialized view log. **No** indicates that a sequence value providing additional ordering information should not be recorded in the materialized view log. Sequence numbers (that is, **Yes** for this option) are necessary to support fast refresh after some update scenarios.
- **Commit SCN:** If this option is enabled, the database is instructed to use commit SCN data rather than timestamps.
- **Available Columns and Selected Columns:** Additional columns, which are non-primary-key columns referenced by subquery materialized views, to be recorded in the materialized view log. To select one or more filter columns, use the arrow buttons to move columns from Available to Selected.

Storage tab

- **Tablespace:** Tablespace in which the materialized view log is to be created
- **Logging:** **YES** or **NO**, to establish the logging characteristics for the materialized view log.
- **Buffer Mode:** Select **KEEP** to put blocks from the segment into the KEEP buffer pool. Select **RECYCLE** to put blocks from the segment into the RECYCLE pool. Select **DEFAULT** to indicate the default buffer pool.
- **Percent Free:** Specify a whole number representing the percentage of space in each data block of the database object reserved for future updates to rows of the object. The value of **PCTFREE** must be a value from 0 to 99.
- **Percent Used:** Specify a whole number representing the minimum percentage of used space that Oracle maintains for each data block of the database object. **PCTUSED** is specified as a positive integer from 0 to 99 and defaults to 40.
- **Initrans:** Specify the initial number of concurrent transaction entries allocated within each data block allocated to the database object. This value can range from 1 to 255 and defaults to 1.
- **Freelists:** In tablespaces with manual segment-space management, for objects other than tablespaces and rollback segments, specify the number of free lists for each of the free list groups for the table, partition, cluster, or index. The default and minimum value for this parameter is 1, meaning that each free list group contains one free list.
- **Freelist Groups:** In tablespaces with manual segment-space management, specify the number of groups of free lists for the database object you are creating.
- **Initial Extent:** Specify the size of the first extent of the object.
- **Next Extent:** Specify in bytes the size of the next extent to be allocated to the object.

- **Percent Increase:** In locally managed tablespaces, Oracle Database uses the value of PCTINCREASE during segment creation to determine the initial segment size and ignores this parameter during subsequent space allocation.
- **Min Extent:** In locally managed tablespaces, Oracle Database uses the value of MINEXTENTS in conjunction with PCTINCREASE, INITIAL and NEXT to determine the initial segment size.
- **Max Extent:** This storage parameter is valid only for objects in dictionary-managed tablespaces. Specify the total number of extents, including the first, that Oracle can allocate for the object.
- **Unlimited:** Select this option if you want extents to be allocated automatically as needed. Oracle recommends this setting as a way to minimize fragmentation.

Purge tab

- **Type:** In **IMMEDIATE SYNCHRONOUS**, the materialized view log is purged immediately after refresh. This is the default. In **IMMEDIATE ASYNCHRONOUS**, the materialized view log is purged in a separate Oracle Scheduler job after the refresh operation.
- **Deferred, Start With, Next, Repeat Interval:** Sets up a scheduled purge that is independent of the materialized view refresh and is initiated during CREATE or ALTER MATERIALIZED VIEW LOG statement.

Refresh tab

- **Type: Synchronous Refresh** creates a staging log that can be used for synchronous refresh. Specify the name of the staging log to be created. The staging log will be created in the schema in which the master table resides. **Fast Refresh** creates a materialized view log that can be used for fast refresh. The materialized view log will be created in the same schema in which the master table resides. This is the default.

DDL pane

Based on the inputs provided, the DDL statements are generated. You can review and save the SQL statements. If you want to make any changes, go back to the relevant pane and make the changes there.

- For a new materialized view log, click **CREATE** to view the generated DDL statements.
- When you edit a materialized view log, click **UPDATE** to view the generated ALTER statements. For a new materialized view log, the UPDATE tab will not be available.

When you are finished, click **Apply**.

Output pane

Displays the results of the DDL commands. If there are any errors, go to the respective pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

Implied Foreign Keys

Implied foreign keys are dependencies that exist between tables but are not defined in the database. In a data warehouse environment, it is a common practice not to create foreign keys. However, it becomes necessary to show these dependencies for presentation or reporting purposes.

You can display implied foreign keys for objects in a star schema by defining them or by discovering them in the data dictionary. It is possible to have more than one source for implied foreign keys.

Define Implied Foreign Keys

You can define implied foreign keys in two ways:

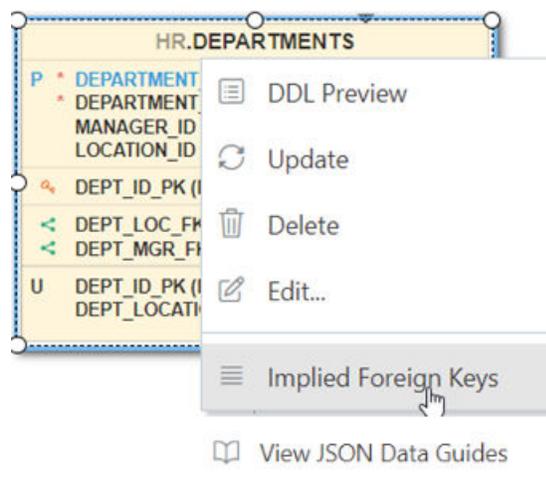
- Using the Implied Foreign Keys dialog
- By dragging the arrow to the referenced object in the diagram

Using the Implied Foreign Keys Dialog

You can define implied foreign keys using the Implied Foreign Keys dialog in Data Modeler.

1. In a Data Modeler diagram, right-click an object (table or view) and select **Implied Foreign Keys**.

Figure 3-17 Select Implied Foreign Keys for an Object



The Implied Foreign Keys dialog is displayed.

2. In the Implied Foreign Keys dialog, click + to add an entry in the grid.
3. Select the entry in the grid to enable and enter values in the following fields:
 - **Referenced Object:** Object in the diagram that has a dependency to the source object.
 - **Local Column:** Name of the column in the source object.
 - **Referenced Column:** Name of the column in the targeted object.
 - **Discovery Sources:** Automatically prefilled, displays whether the implied foreign keys have been defined or were discovered in the data dictionary.
4. Click **OK**. The implied foreign key dependency is displayed with a dotted line on the diagram.

Figure 3-18 Dotted Line Between Two Objects

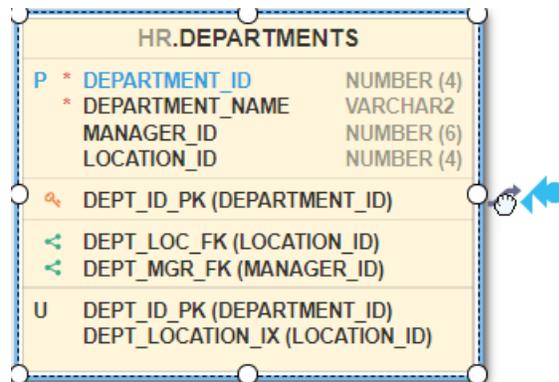


By Dragging the Arrow to the Referenced Object in the Diagram

You can also define an implied foreign key in the following way:

1. Select the source object on the diagram.
2. Click and drag the small curved arrow with a blue indicator to the referenced object. The dependency will be displayed with a dotted line on the diagram.

Figure 3-19 Drag Blue Indicator



3. Right-click the source object and enter the column names in the Implied Foreign Keys dialog.

Discover Implied Foreign Keys in the Data Dictionary

A star schema can be discovered by searching for several types of definitions in the data dictionary.

In the Navigator tab, right-click an object (table or view) and select **Add Object as Star Schema to Diagram**. The object must be a fact table. The data dictionary is then searched for joins and dependencies related to the object, such as:

- Foreign keys defined for the selected table to other tables. If implied foreign keys are later discovered for the same columns, they will not be displayed on the diagram.
- Joins used in the definitions of bitmap join indexes, materialized views with aggregates, and attribute clustering.
- Dependencies based on dimension definitions and column name matching in fact table.
- Fact and dimension definitions for Analytic views and OLAP cube and dimensions.

4

The REST Pages

The REST pages enable you to develop and protect ORDS (Oracle REST Data Services) based RESTful web services for your database.

To navigate to the REST pages, do either of the following:

- In the Launchpad page, select the **Development** tab and click **REST**.
- Click **Selector**  to display the navigation menu. Under Development, select **REST**.

Topics

- [About RESTful Web Services](#)
- [RESTful Services Terminology](#)
- [About the Overview Page](#)
- [About the AutoREST Page](#)
- [About the REST Search Toolbar](#)
- [Creating RESTful Web Services](#)
- [Securing RESTful Web Services](#)
- [About RESTful Web Services](#)
Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. A service is described as RESTful when it conforms to the tenets of REST.
- [RESTful Services Terminology](#)
- [About the Overview Page](#)
The Overview page provides an overview of the activity in the REST pages.
- [About the AutoREST Page](#)
Use the Automatic Enabling of Schema Objects for REST Access (AutoREST) feature to enable REST access for database objects in the current schema.
- [About the REST Search Toolbar](#)
This section describes the various options available on the search toolbar in the REST pages. The options selected are saved for your session.
- [Creating RESTful Web Services](#)
Create RESTful web services using the Modules, Templates and Handlers pages.
- [Securing RESTful Web Services](#)
Define roles, privileges and OAuth Clients to ensure authentication and authorization are required for accessing RESTful web services.

About RESTful Web Services

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. A service is described as RESTful when it conforms to the tenets of REST.

A RESTful service has the following characteristics:

- Data is modeled as a set of resources. Resources are identified by URIs.
- A small, uniform set of operations are used to manipulate resources (for example, GET, POST, PUT, DELETE).
- A resource can have multiple representations (for example, a blog might have an HTML representation and a RSS representation).
- Services are stateless and because it is likely that the client will want to access related resources, these should be identified in the representation returned, typically by providing hypertext links.

RESTful Services Terminology

The following are some major terms related to RESTful services.

- **RESTful service:** An HTTP web service that conforms to the tenets of the RESTful architectural style.
- **Resource module:** An organizational unit that is used to group related resource templates.
- **Resource template:** An individual RESTful service that is able to service requests for some set of URIs (Universal Resource Identifiers). The set of URIs is defined by the URI pattern of the Resource Template
- **URI pattern:** A pattern for the resource template. Can be either a route pattern or a URI template, although you are encouraged to use route patterns.
 - **Route pattern:** A pattern that focuses on decomposing the path portion of a URI into its component parts. For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the `emp` resource with `id` of 101) and will also match `/objects/emp/` (matches a request for the `emp` resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the `id` parameter is optional).
 - **URI template:** A simple grammar that defines the specific patterns of URIs that a given resource template can handle. For example, the pattern `employees/{id}` will match any URI whose path begins with `employees/`, such as `employees/2560`.
- **Resource handler:** Provides the logic required to service a specific HTTP method for a specific resource template. For example, the logic of the GET HTTP method for the preceding resource template might be:

```
select empno, ename, dept from emp where empno = :id
```
- **HTTP operation:** HTTP (HyperText Transport Protocol) defines the standard methods that are supported in Oracle REST Data Services for building RESTful services: GET (retrieve the resource contents), POST (store a new resource), PUT (update an existing resource), and DELETE (remove a resource).

📘 See Also

- [Structure of RESTful Web Services Video](#)
- [Getting Started with RESTful Services](#)

About the Overview Page

The Overview page provides an overview of the activity in the REST pages.

To navigate to the Overview page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **REST**.
- Click **Selector**  to display the navigation menu. Under Development, select **REST**.

You can navigate to pages within REST by using the menu in the header: Overview, Modules, AutoREST and Security.

Use **Export Schema** to copy or download all the roles, privileges, modules, REST-enabled objects, and OAuth clients defined in the schema.

The Overview page consists of three sections.

Objects

Displays the total number of resource modules, schema objects with REST access, roles, privileges, OAuth clients, and pre-authenticated requests created. Click the respective card to navigate to the corresponding pages for modules, roles, privileges, OAuth clients, and pre-authenticated requests.

In the REST pages, the term "object" is used to refer to a module, template, handler, role, privilege, OAuth Client, or pre-authenticated request.

Security

Metadata Catalog: Displays whether access to the ORDS metadata catalog of the schema is protected. The metadata catalog lists all the services in the schema. The user authorization for the metadata catalog is provided at the time of enabling user access.

Modules: Displays the number of modules that are published out of the total modules created.

Module Security: Displays the number of modules that are protected by a privilege out of the total modules created. A partially secured privilege is where the module is protected both by a privilege and a pattern.

Recent Objects

Displays recently updated or created objects. Each entry has a context menu available at the end of the row. Click **Actions**  to access options such as Edit and Delete.

See Also

[Accessing Database Actions](#)

About the AutoREST Page

Use the Automatic Enabling of Schema Objects for REST Access (AutoREST) feature to enable REST access for database objects in the current schema.

Enabling REST access for a schema object such as table, view, function, procedure and package allows it to be accessed through RESTful services. See AutoREST in *Oracle REST Data Services Developer's Guide*

To navigate to the AutoREST page, in the REST Overview page, click **AutoREST** or select **AutoREST** from the menu in the header.

The AutoREST page consists of three parts:

- **Objects:** Displays the number of REST-enabled schema objects that are protected and require authorization out of the total number of schema objects enabled in the schema.
- **AutoREST:** Displays the number of tables and views, packages, procedures and functions that are REST enabled in the schema. Click a card to see the corresponding objects listed below.
- The bottom part of the page lists schema objects based on the selection made in AutoREST. You can view these objects in card view or grid view. The actions available in the context menu for an object are:
 - **Edit:** Edit the Object Alias or Require Authentication fields for the REST enabled object. For a description of the fields, see [Enabling REST Access for a Database Object](#)
 - **Disable:** Disable REST access for the object. See [Disabling REST Access for a Database Object](#)
 - **Get Curl:** Generate cURL calls for a REST enabled object. See [Generating cURL Requests for a REST-Enabled Database Object](#)
 - **Export OpenAPI:** Export the object as JSON Open API code.
 - **OpenAPI View:** Display the object as a Swagger UI implementation. You can view and execute the handlers, pass parameters to the handlers, and copy or download the responses.
- [Enabling REST Access for a Database Object](#)
You can enable REST access for a table, view, materialized view, function, procedure or package.
- [Disabling REST Access for a Database Object](#)
This section describes how to disable REST access for a database object.
- [Generating cURL Requests for a REST-Enabled Database Object](#)
For a REST-enabled database object, you can generate sample cURL calls for GET, POST, PUT and DELETE requests for the selected object. Use the Copy to Clipboard icon to copy the cURL code.

📘 See Also

- [About the REST Search Toolbar](#)
- [About the Overview Page](#)

Enabling REST Access for a Database Object

You can enable REST access for a table, view, materialized view, function, procedure or package.

To REST enable a schema object:

1. In the Navigator tab in the SQL page, right-click an object and select **REST**, then select **Enable**.
2. Enter the following fields:
 - **Object Alias:** Enter another name for the object. For security purposes, it is recommended to use a different name than the actual object name.
 - **Require Authentication:** Select this option to secure the REST endpoint.
 - **Authorization Role** and **Authorization Privilege:** These fields are automatically pre-filled.

Show code: Select this option to view the PL/SQL code equivalent of the Enable REST Access slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Enable** in the Enable REST Access slider.

3. Click **Enable**.

A REST-enabled object is indicated by a **REST Enabled** icon  in the Navigator tab.

Disabling REST Access for a Database Object

This section describes how to disable REST access for a database object.

1. In the SQL page, in the Navigator tab, right-click the database object that is REST enabled and select **Disable**.
2. You see a prompt to confirm. Click **OK**.

Generating cURL Requests for a REST-Enabled Database Object

For a REST-enabled database object, you can generate sample cURL calls for GET, POST, PUT and DELETE requests for the selected object. Use the Copy to Clipboard icon to copy the cURL code.

To view the cURL call for a specific method for the selected object:

1. In the SQL page, in the Navigator tab, right-click the REST-enabled object, select **REST** and then select **cURL command**.
2. Select the HTTP method from the left pane, enter values if required and click **Next** to see the cURL command.

For more information about each HTTP method, see [Examples in Oracle REST Data Services](#).

About the REST Search Toolbar

This section describes the various options available on the search toolbar in the REST pages. The options selected are saved for your session.



Search: Performs a search based on the text entered in the field. This field is not case sensitive. When you start typing in the Search field, the  icon appears. If you click the icon, the text that you typed is deleted and you can re-enter text again.

Filter by: Performs a search based on the filters selected. In the Filter by drop down, first select the column to apply the filter on, and then select the values to search for in that column.

Sort By: Sorts the data in ascending or descending based on the columns selected. The columns can be turned on or off. Reset restores the columns to the original state. The sort criteria selected is shown in the upper right corner as Sort By.

Page Size. First Page, Previous page, Next Page: Only displayed in the Card View. Enables you to select page size, and move between pages.

 **Reload:** Refreshes the information on the page.

There are two views available:

 **Open Card View:** This is the default display view. Information is displayed for each object in a card format. Each card displays associated attributes for the object along with the context menu, which list the actions available for the object.

 **Open Grid View:** Displays the information for each object in a single row of a table. The last column in each row contains the context menu icon, which shows the actions available for the object.

Creating RESTful Web Services

Create RESTful web services using the Modules, Templates and Handlers pages.

To create a RESTful web service, you need to:

- Define a resource module
- Define a resource template
- Define one or more resource handlers such as GET, PUT, POST or DELETE. Optionally, define parameters that you need to pass to the resource handler.

The following sections provide information on how to create resource modules, resource templates and resource handlers.

- [Managing Resource Modules](#)
- [Managing Resource Templates](#)
- [Managing Resource Handlers](#)
- [Example: Inserting a Record using a POST Handler](#)
- [Viewing Resource Handler Details and Managing Parameters](#)
- [Using Pre-Authenticated Requests to Access ORDS RESTful Services](#)
- [Managing Resource Modules](#)
You can define, edit and delete resource modules in the Modules page.
- [Managing Resource Templates](#)
You can define, edit and delete resource templates for a module in the Templates page.
- [Managing Resource Handlers](#)
You can create, edit and delete resource handlers for a template in the Handlers page.
- [Example: Inserting a Record using a POST Handler](#)
The following example illustrates how to insert a record in the DEPT table.

See Also

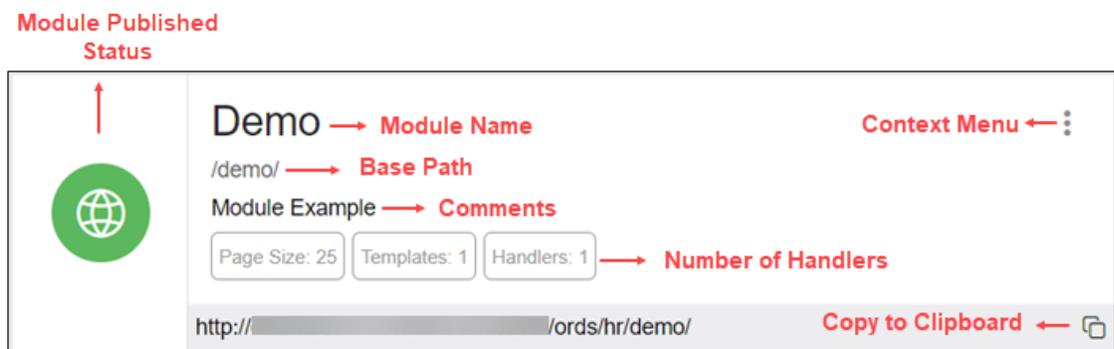
- [REST Workshop Overview Video](#)
- [Creating a RESTful Service in Oracle REST Data Services Quick Start Guide](#)

Managing Resource Modules

You can define, edit and delete resource modules in the Modules page.

To navigate to the Modules page, in the REST Overview page, click **Modules** in Objects, or from the menu in the header, select **Modules**.

The module attributes displayed by default in card view are shown in the following figure.



The module status can be published or unpublished. If the status is unpublished, the icon is displayed in a red color.

Click the module name in card view to go to the Templates page for that module. You can also navigate to the Templates page through the context menu.

The actions available in the context menu are:

- [Creating a Resource Module](#)
- [Importing a Resource Module](#)
- [Editing a Resource Module](#)
- [Deleting a Resource Module](#)
- [Publishing/Unpublishing a Resource Module](#)
- [Exporting a Resource Module](#)
- [Viewing the Module in OpenAPI View](#)
- Navigating to the Templates page for a module
- [Creating a Resource Module](#)
This section describes how to create a resource module.
- [Importing a Resource Module](#)
- [Editing a Resource Module](#)
This section describes how to edit a resource module.

- [Deleting a Resource Module](#)
This section describes how to delete a resource module.
- [Publishing/Unpublishing a Resource Module](#)
This section describes how to make a resource module available or not.
- [Exporting a Resource Module](#)
You can export the PL/SQL source code for the API or Open API (Swagger) JSON code for a resource module or RESTful service. If the module has templates, handlers, parameters, roles, and so on, these objects are also displayed and exported.
- [Viewing the Module in OpenAPI View](#)
This section describes how to open the module in Open API view.

① See Also

[About the REST Search Toolbar](#)

Creating a Resource Module

This section describes how to create a resource module.

1. In the Modules page, click **Create Module**.
2. Enter the following fields. The fields with an asterisk (*) are mandatory:

Module Definition tab

- **Module Name:** Enter the name of the module. This field is case sensitive.
- **Base Path:** Enter the base of the URI for accessing the RESTful service. For example: hr/ means that all URIs starting with hr/ will be serviced by this resource module. Note the change in the **Preview URL** as you enter the base path.
- **Is Published:** Enable to make the RESTful service available for use.
- **Pagination Size:** Enter or select the number of results to return on each page based on a database query. The default value is 25.
- **Protected By Privilege:** Select a privilege to protect the module from the drop-down list.
- **Comments:** Enter descriptive comments.
- **Go to Module after creation:** Select this option to go to the Templates page after the module is created.

Origins Allowed tab

- Enter origins that are allowed to access the resource templates. Click **Add (+)** to add each origin. For example:

http://example1.org

https://*.example2.com

Show code: Select this option to view the PL/SQL code equivalent of the Create Module slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Module slider.

3. Click **Create**.

The new module is displayed on the Modules page. If you cannot see it, locate the module using the Search field.

Importing a Resource Module

This section describes how to import a resource module.

Note

You can import an OpenAPI version 2.0 or 3.0 document in JSON format only.

1. In the Modules page, at the top right, click **Import Module**.
2. Browse and select the JSON file to import. Click **Open**.

The Import Module slider is displayed with **Base Path** and **Preview URL** fields prefilled. Edit the Base Path as needed. Editing the base path field will also update the base path in the JSON format that is displayed in the slider.

3. Click **Import**.

When the module is imported, a privilege configuration is created and set to protect the module. The name of the privilege is generated by using the base path and adding "_priv" at the end of the base path string and also replacing every slash "/" in the base path to underscore "_".

4. Your module is ready for editing.

After your module is imported, the templates and handlers are created, but the module is not published. You need to supply SQL and PL/SQL blocks for each handler, and publish your module when it is ready for testing or production.

Editing a Resource Module

This section describes how to edit a resource module.

1. In the Modules page, for the specific module, click  and select **Edit**.
2. Edit the required fields and click **Save**. For a description of the fields, see [Creating a Resource Module](#).

Deleting a Resource Module

This section describes how to delete a resource module.

1. In the Modules page, for the specific module, click  and select **Delete**.
A prompt appears asking you to confirm.
2. Click **OK** to delete.

Publishing/Unpublishing a Resource Module

This section describes how to make a resource module available or not.

1. In the Modules page, for the specific module, click  and then select **Publish** or **Unpublish**.
You see a prompt to confirm.
2. Click **OK**.

Exporting a Resource Module

You can export the PL/SQL source code for the API or Open API (Swagger) JSON code for a resource module or RESTful service. If the module has templates, handlers, parameters, roles, and so on, these objects are also displayed and exported.

1. In the Modules page, for a specific module, click  and select **Export Module**.
2. Select **PL/SQL** to copy or export the PL/SQL source for the API. Select **OpenAPI** to copy or export the Open API (Swagger) JSON code for the module.

Note

A module exported using Open API is always imported with an unpublished status.

3. Click **Copy to Clipboard** to copy the code or click **Download** to download to your local computer.

Viewing the Module in OpenAPI View

This section describes how to open the module in Open API view.

The Open API view option is available for modules that are published and not protected.

1. In the Modules page, for the specific module, click  and select **OpenAPI View**.
2. The module is displayed as a Swagger UI implementation.

You can view and execute the handlers, pass parameters to the handlers, and copy or download the responses.

To learn more about Swagger UI implementation, refer to the official [Swagger UI](#) website.

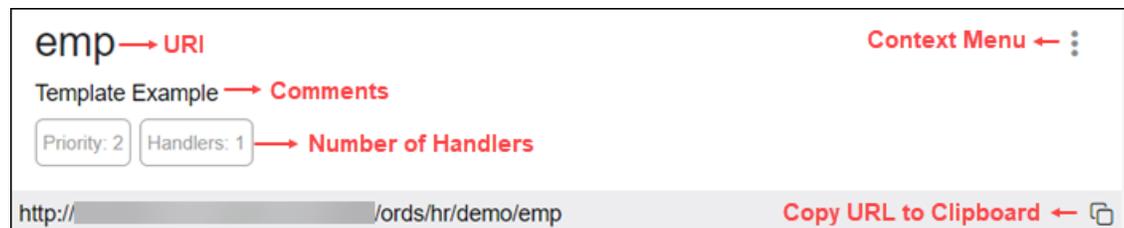
Managing Resource Templates

You can define, edit and delete resource templates for a module in the Templates page.

To navigate to this page, in the Modules page, click  for a module and select **Templates**, or click the name of the module in card view.

At the top of the page, the module card is available with context menu options such as Edit, Delete, Publish and Export.

The templates attributes displayed by default in card view are shown in the following figure.



Click the template name in card view to go to the Handlers page. You can also navigate to the Handlers page through the context menu.

The actions available in the context menu are:

- [Creating a Resource Template](#)
- [Editing a Resource Template](#)
- [Deleting a Resource Template](#)
- Navigating to the Handlers page for the template
- [Creating a Resource Template](#)
This section describes how to create a resource template.
- [Editing a Resource Template](#)
This section describes how to edit a resource template.
- [Deleting a Resource Template](#)
This section describes how to delete a resource template.

Creating a Resource Template

This section describes how to create a resource template.

1. In the Templates page for a module, click **Create Template**.
2. Enter the following fields. The fields with an asterisk (*) indicate that they are mandatory:

- **URI Template:** Enter the URI pattern for the resource template.

For example, a pattern of `/objects/:object/:id?` will match `/objects/emp/101` (matches a request for the item in the emp resource with id of 101) and will also match `/objects/emp/` (matches a request for the emp resource, because the `:id` parameter is annotated with the `?` modifier, which indicates that the id parameter is optional).

Note that the **Preview URL** changes as you type the URI template.

- **Priority:** Enter or choose the priority for how the resource template should be evaluated (1 is low priority and 9 is high priority).
- **HTTP Entity Tag Type:** Select the type of entity tag to be used by the resource template. An entity tag is an HTTP Header that acts as a version identifier for a resource.

Options include:

- **Secure HASH (default):** The contents of the returned resource representation are hashed using a secure digest function to provide a unique fingerprint for a given resource version.
- **Query:** Define a query manually that uniquely identifies a resource version. A manually defined query can often generate an entity tag more efficiently than hashing the entire resource representation.
- **None:** Do not generate an entity tag.
- **Comments:** Enter descriptive comments if any.
- **Go to Handlers after creation:** Select this option to go to the Handlers page after the template is created.

Show code: Select this option to view the PL/SQL code equivalent of the Create Template slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Template slider.

3. Click **Create**.

The new template for the module is added to the Templates page.

If "Go to Handlers after creation" is selected, you can use the module name in the breadcrumbs at the top of the page to see the list of templates for the module.

Editing a Resource Template

This section describes how to edit a resource template.

1. In the Templates page, for a specific template, click  and then select **Edit**.
2. Edit the required fields and click **Save**.

For a description of the fields, see [Creating a Resource Template](#).

Deleting a Resource Template

This section describes how to delete a resource template.

1. In the Templates page, for a specific template, click  and then select **Delete**.
You are prompted to confirm.
2. Click **OK** to delete.

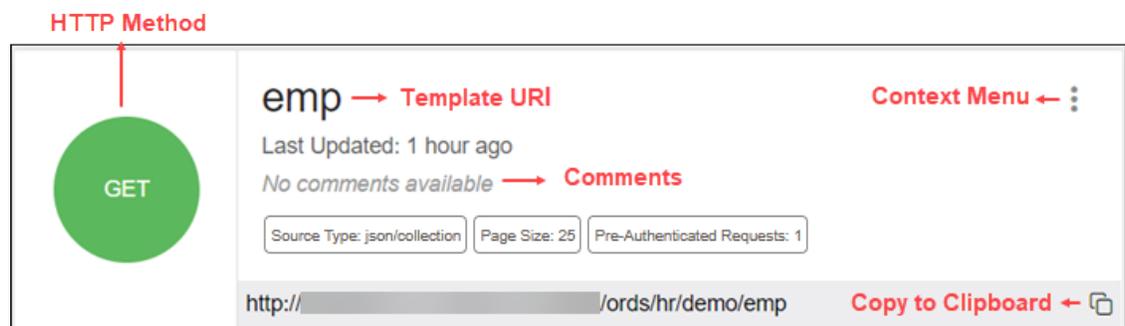
Managing Resource Handlers

You can create, edit and delete resource handlers for a template in the Handlers page.

To navigate to this page, in the Templates page, click  for a template and select **Handlers**, or click the name of the template in card view.

At the top of the page, the template card is available with context menu options such as Edit and Delete.

The handler attributes displayed by default in card view are shown in the following figure.



The HTTP Method can be one of the following: GET, PUT, POST and DELETE.

You can navigate to the HTTP method page using the context menu. There are three tabs displayed in the HTTP method page: Source, Parameters and Pre-Authenticated Requests.

To create a resource handler, see [Creating a Resource Handler](#).

Note

This option is not available if the template has all four handlers created (GET, POST, PUT, DELETE).

The actions available in the context menu are:

- **Edit:** See [Editing a Resource Handler](#)
- **Delete:** See [Deleting a Resource Handler](#)
- **Details:** See [Viewing Resource Handler Details and Managing Parameters](#)
- **Pre-Authenticated Requests:** See [Pre-authenticated Requests](#)
- [Creating a Resource Handler](#)
This section describes how to create a resource handler.
- [Editing a Resource Handler](#)
This section describes how to edit a resource handler.
- [Deleting a Resource Handler](#)
This section describes how to delete a resource handler.
- [Viewing Resource Handler Details and Managing Parameters](#)
You can view resource handler details and manage parameters in the *HTTP method* page.
- [Using Pre-Authenticated Requests to Access ORDS RESTful Services](#)
To access protected RESTful services, you can create and revoke pre-authenticated requests.

Creating a Resource Handler

This section describes how to create a resource handler.

1. In the Handlers page, click **Create Handler**.
2. Specify the properties of the resource handler. The specific options depend on the method type.

Handler Definition Tab

- **Method:** Enter the HTTP request method for this handler: GET (retrieves a representation of a resource), POST (creates a new resource or adds a resource to a collection), PUT (updates an existing resource), or DELETE (deletes an existing resource). Only one handler for each HTTP method is permitted.
- **Items Per Page:** Enter or choose the default pagination size, or the number of rows to return for a database query. This option is only available for GET handlers. If this value is not defined, the number of items per page is the one defined in the module.
- **Source Type:** Select the source implementation for the selected HTTP method. The default is collection query.
 - **Collection Query:** Executes a SQL query and transforms the result set into a **JSON** representation. Available when the HTTP method is GET.
 - **Collection Item:** Executes a SQL query returning one row of data into a **JSON** representation. Available when the HTTP method is GET.
 - **Media:** Executes a SQL Query conforming to a specific format and turns the result set into a **binary** representation with an accompanying HTTP Content-Type header identifying the Internet media type of the representation.

- **PL/SQL**: Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a **JSON** representation.
- **Source**: Enter the SQL query or the PL/SQL block for the HTTP method. In the handler details page, you can test the SQL or PL/SQL handler code in the handler editor.
- **Go to Handler after creation**: Select this option to go to the details page after the handler is created.

MIMEs Allowed tab

Identifies the type of information included in the HTTP Request Body. For example, if the POST Handler is expecting JSON in the request, add "Application/JSON" as the MIME type.

Show code: Select this option to view the PL/SQL code equivalent of the Create Handler slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Handler slider.

3. Click **Create**.

The handler is displayed on the Handlers page for the specific template.

You can test the REST service endpoint using a REST client or a command-line tool such as cURL.

① See Also

About cURL and Testing RESTful Services in *Oracle REST Data Services Developer's Guide*.

Editing a Resource Handler

This section describes how to edit a resource handler.

1. In the Handlers page, for a specific handler, click  and then click **Details**.
2. Edit the required fields and click **Save**.

For a description of the fields, see [Creating a Resource Handler](#).

Deleting a Resource Handler

This section describes how to delete a resource handler.

1. In the Handlers page, for the specific handler, click  and then click **Delete**.
You are prompted to confirm.
2. Click **OK** to delete.

Viewing Resource Handler Details and Managing Parameters

You can view resource handler details and manage parameters in the *HTTP method* page.

To navigate to this page, for a specific handler, click  and then select **Details** or click the name of the handler in card view.

At the top of the page, the handler card is available with context menu options such as Edit and Delete.

The Handler *HTTP method* page has two major sections:

- A code editor for executing the SQL or PL/SQL source code that was defined in the resource handler. You can execute the code, save, undo, redo, download and clear output. For more information about the editor, see [Executing SQL Statements in the Code Editor](#).

Implicit parameters used in REST service handlers are displayed in a scrolling list towards the right side of the editor. See [Implicit Parameters](#)

- A section for managing parameters required for running the HTTP method.

By default, the parameters are displayed in grid view. The attributes displayed on a parameter in card view are shown in the following figure.



The actions available in the context menu are:

- [Creating a Parameter](#)
- [Editing a Parameter](#)
- [Deleting a Parameter](#)
- [Creating a Parameter](#)
This section describes how to create a parameter for a resource handler.
- [Editing a Parameter](#)
This section describes how to edit a parameter.
- [Deleting a Parameter](#)
This section describes how to delete a parameter.
- [Implicit Parameters](#)
This section describes the implicit parameters in the Resource Handler Details page. These parameters are automatically added to the resource handlers.

[See Also](#)

[About the REST Search Toolbar](#)

Creating a Parameter

This section describes how to create a parameter for a resource handler.

1. In the handler *HTTP method* page, click **Create Parameter**.
2. Enter the following fields. The fields with an asterisk (*) are mandatory:

- **Parameter Name:** Enter the name of the parameter, as it is named in the URI Template or HTTP Header. The name defines how the parameter is identified in the incoming request or how the parameter is named in the Response.
- **Bind Variable Name:** Enter the name of the parameter, as it will be referred to in SQL. If NULL is specified, then the parameter is unbound.
- **Source Type:** Select the type that is identified if the parameter originates in the URI Template or HTTP Header.
- **Parameter Type:** Select the native type of the parameter.
- **Access Method:** Select the parameter access method. Indicates if the parameter is an input value, output value, or both.

Show code: Select this option to view the PL/SQL code equivalent of the Create Parameter slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Parameter slider.

3. Click **Create**.

The parameter is added in the Parameters section.

Editing a Parameter

This section describes how to edit a parameter.

1. In the Parameters section, for the specific parameter, click  and then select **Edit**.
2. Edit the required fields and click **Save**.

For a description of the fields, see [Creating a Parameter](#).

Deleting a Parameter

This section describes how to delete a parameter.

1. In the Parameters section, for the specific parameter, click  and then select **Delete**.
You are prompted to confirm.
2. Click **Yes** to delete.

Implicit Parameters

This section describes the implicit parameters in the Resource Handler Details page. These parameters are automatically added to the resource handlers.

Table 4-1 Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description
:body	BLOB	IN	N/A	Specifies the body of the request as a temporarily BLOB.

Table 4-1 (Cont.) Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description
:body_text	CLOB	IN	N/A	Specifies the body of the request as a temporary CLOB.
:content_type	VARCHAR	IN	Content-Type	Specifies the MIME type of the request body, as indicated by the Content-Type request header.
:current_user	VARCHAR	IN	N/A	Specifies the authenticated user for the request. If no user is authenticated, then the value is set to null.
:forward_location	VARCHAR	OUT	X-ORDS-FORWARD-LOCATION	Specifies the location where Oracle REST Data Services must forward a GET request to produce the response for this request.

Table 4-1 (Cont.) Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description
:fetch_offset	NUMBER	IN	N/A	Specifies the zero-based offset of the first row to be displayed on a page.
:fetch_size	NUMBER	IN	N/A	Specifies the maximum number of rows to be retrieved on a page.
:page_offset	NUMBER	IN	N/A	Specifies the zero based page offset in a paginated request. Note: The :page_offset parameter is deprecated. Use :row_offset parameter instead.

Table 4-1 (Cont.) Implicit Parameters

Name	Type	Access Mode	HTTP Header	Description
:page_size	NUMBER	IN	N/A	Specifies the maximum number of rows to be retrieved on a page. The :page_size parameter is deprecated. Use :fetch_size parameter instead.
:row_offset	NUMBER	IN	N/A	Specifies the one-based index of the first row to be displayed in a paginated request.
:row_count	NUMBER	IN	N/A	Specifies the one-based index of the last row to be displayed in a paginated request.
:status_code	NUMBER	OUT	X-ORDS-STATUS-CODE	Specifies the HTTP status code for the request.

For more information about the implicit parameters, see *Oracle REST Data Services Developer's Guide*.

Using Pre-Authenticated Requests to Access ORDS RESTful Services

To access protected RESTful services, you can create and revoke pre-authenticated requests.

Pre-authenticated requests enable you to access protected ORDS RESTful services without user credentials. When you create pre-authenticated request, a unique URL is generated. You can provide this URL to an external user to interact with the particular RESTful entity using standard HTTP tools.

For more information about Pre-Authenticated Requests, see Oracle REST Data Services Pre-Authenticated Requests in the *Oracle REST Data Services Developer's Guide*.

To navigate to the Pre-Authenticated Requests page, you can do either of the following:

- From the Overview page, click the **Pre-Authenticated Requests** tile.
- From the menu bar, expand **Security** and select **Pre-Authenticated Requests**.

Topics:

- [Creating a Pre-Authenticated Request](#)
- [Viewing a Pre-Authenticated Request](#)
- [Creating a Pre-Authenticated Request](#)
- [Viewing a Pre-Authenticated Request](#)
- [Revoking a Pre-Authenticated Request](#)

Creating a Pre-Authenticated Request

To create a pre-authenticated request for an existing handler:

1. From the context menu for an existing handler, select **Create New Pre-authenticated Request**.

The Create Pre-authenticated Request panel appears.

2. In the **Token Expiration** field, enter the token expiry duration in seconds.

Select **Show Code** to view the underlying source code for creating the pre-authenticated request.

Click **Create**.

The token, alias, and relative link to access the resource are displayed. You must keep track of the token and alias as you cannot obtain their values later. If you forget the values, you can revoke the token and create a new one again.

Viewing a Pre-Authenticated Request

From the context menu for an existing handler, select **Pre-Authenticated Requests** and then select **Details**.

The pre-authenticated requests for the specific handler appears.

Revoking a Pre-Authenticated Request

To revoke the pre-authenticated request for a handler:

1. From the context menu for an existing handler, select **Pre-Authenticated Requests** and then select **Details**.
The pre-authenticated requests for the specific handler appears.
2. From the context menu for the pre-authenticated request, select **Revoke**.
You are prompted to confirm.
3. Click **Confirm** to revoke the request.

Example: Inserting a Record using a POST Handler

The following example illustrates how to insert a record in the DEPT table.

Prerequisites

- Using the worksheet in the SQL page, create a **DEPT** table.

```
CREATE TABLE DEPT(
  DEPTNO      number(2,0),
  DNAME       varchar2(14),
  LOC         varchar2(13),
  CONSTRAINT PK_DEPT PRIMARY KEY (DEPTNO)
)
```

- Create a module named **example**. See [Creating a Resource Module](#)
- Create a template named **emp/** for the module. See [Creating a Resource Template](#)

To insert a record:

1. In the Handlers page, click **Create Handler**.
2. Enter the following details:
 - In the **Method** field, select **POST**.
 - In the **Source Type** field, select **PL/SQL**.
 - In the **Source** field, enter the following PL/SQL block:

```
DECLARE
  id number;
BEGIN
  INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES
  (:DEPTNO, :DEPTNAME, :DEPTLOC) RETURNING DEPTNO INTO id;
  :status_code := 201;
  :forward_location := id;
END;
```

where

- RETURNING DEPTNO INTO id returns the value assigned to DEPTNO into the variable id.
- :status_code is an implicit parameter that is assigned 201 to indicate that the resource is created.
- :forward_location is an implicit parameter that is assigned the id that contains the value of DEPTNO. This parameter specifies the location to forward a GET request to produce the response for the POST request.

- Ensure **Go to Handler page after creation** is selected.
- In the **MIMEs Allowed** Tab, select **application/json** from the drop-down list and click **Add +**.
- Click **Create**.

The Handler POST page appears.

You can test the source code by clicking **Run Statement**  under the Source tab.

3. Create a GET resource handler to produce a response for the POST request.

In the Example module, create a template named **emp/:id**.

Create a GET resource handler where **Source Type** is **Collection Item** and **Source** contains the following query:

```
SELECT * FROM DEPT WHERE DEPTNO = :id
```

4. Test the RESTful service endpoint using the following command in cURL. You can copy the URL (<http://xyz.us.comp.com:1234/ords/pdbdba/example/emp>) using the Copy to Clipboard icon in the POST handler card at the top.

```
curl --location --request POST --header "Content-Type: application/json"
--data '{"DEPTNO": 54, "DEPTNAME": "HR", "DEPTLOC": "America"}' 'http://
xyz.us.comp.com:1234/ords/pdbdba/example/emp/'
```

The output is:

```
{"deptno":54,"dname":"HR","loc":"America","links":[{"rel":"collection",
"href":"xyz.us.comp.com:1234/ords/pdbdba/example/emp/"}]}
```

5. In the SQL page, check the DEPT table to see if the new record has been inserted by using the following statement:

```
SELECT * FROM DEPT;
```

Securing RESTful Web Services

Define roles, privileges and OAuth Clients to ensure authentication and authorization are required for accessing RESTful web services.

To protect a RESTful web service, you need to:

- Create a role
- Create a privilege selecting the role and modules or resources to protect

To enable access to a protected RESTful service using the OAUTH2 Workflow, create an OAuth client using the role and privilege created for protecting the RESTful service.

The following sections provide information on how to create roles, privileges and OAuth clients:

- [Managing Roles](#)
- [Managing Privileges](#)
- [Managing OAuth Clients](#)
- [Examples](#)

- [Managing Roles](#)
You can create, edit and delete roles for RESTful services in the Roles page.
- [Managing Privileges](#)
You can create, edit and delete privileges for RESTful services in the Privileges page.
- [Managing OAuth Clients](#)
Using OAuth 2.0-based authentication, you can ensure that your RESTful web services are accessed only by specific users or clients.
- [Examples](#)
This section provides a few examples on creating an OAuth client with different grant types.

📘 See Also

- [REST Workshop Overview Video](#)
- Tutorial for Protecting and Accessing Resources

Managing Roles

You can create, edit and delete roles for RESTful services in the Roles page.

To navigate to the Roles page, from the REST Overview page, click **Roles** in Objects, or from the menu in the header, select **Security** and then select **Roles**.

The actions available in the context menu are:

- [Creating a Role](#)
- [Editing a Role](#)
- [Deleting a Role](#)
- [Viewing Assigned Privileges](#)
- [Creating a Role](#)
Create a role with a specific name. After the role is created, you can associate it with a privilege.
- [Editing a Role](#)
This section describes how to edit a role.
- [Deleting a Role](#)
This section describes how to delete a role.
- [Viewing Assigned Privileges](#)
This section describes how to view the privileges associated with a role.

📘 See Also

About Users and Roles in *Oracle REST Data Services Developer's Guide*.

Creating a Role

Create a role with a specific name. After the role is created, you can associate it with a privilege.

1. In the Roles page, click **Create Role**.
2. In the Role Name field, enter the name of the role to create.

Show code: Select this option to view the PL/SQL code equivalent of the Create Role slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Role slider.

3. Click **Create**.

The new assigned role is displayed on the Roles page.

Editing a Role

This section describes how to edit a role.

1. In the Roles page, for the specific role, click **Actions**  and select **Edit**.
2. Enter the changes required and click **Save**.

For a description of the fields, see [Creating a Role](#).

Deleting a Role

This section describes how to delete a role.

1. In the Roles page, for the specific role, click **Actions**  and select **Delete**.
You are prompted to confirm.
2. Click **Yes** to delete.

Viewing Assigned Privileges

This section describes how to view the privileges associated with a role.

In the Roles page, for the specific role, click **Actions**  and select **Details**. The privileges assigned to the role are displayed.

Managing Privileges

You can create, edit and delete privileges for RESTful services in the Privileges page.

A privilege defines the set of roles, at least one of which an authenticated user must possess to access a RESTful service protected by a privilege.

To navigate to the Privileges page, from the REST Overview page, click **Privileges** in Objects, or from the menu in the header, select **Security** and then select **Privileges**.

The privilege attributes displayed by default in card view are shown in the following figure.



The actions available in the context menu are:

- [Creating a Privilege](#)
- [Editing a Privilege](#)
- [Deleting a Privilege](#)
- [Creating a Privilege](#)
This section describes how to create a privilege.
- [Editing a Privilege](#)
This section describes how to edit a privilege.
- [Deleting a Privilege](#)
This section describes how to delete a privilege.

Creating a Privilege

This section describes how to create a privilege.

1. In the Privileges page, click **Create Privilege**.
2. Enter the following fields. The fields with an asterisk (*) are mandatory:
 - **Label:** Enter the name of the privilege in a way that it is easy to understand.
 - **Name:** Enter a unique name for the privilege.
 - **Description:** Enter a brief description of the purpose of the privilege.
 - **Comments:** Enter comments.
 - **Roles:** Enter one or more roles assigned to the privilege.
 - **Protected Modules:** Select the modules to protect.
 - **Protected Resources:** Instead of Protected Modules, use the Protect Resources tab to apply security based on a URI pattern.

Show code: Select this option to view the PL/SQL code equivalent of the Create Privilege slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Privilege slider.
3. Click **Create**.

The new privilege is displayed on the Privileges page.

Editing a Privilege

This section describes how to edit a privilege.

1. In the Privileges page, for the specific privilege, click **Actions**  and select **Edit**.
2. Make the required changes and click **Save**.
For a description of the fields, see [Creating a Privilege](#).

Deleting a Privilege

This section describes how to delete a privilege.

1. In the Privileges page, for the specific privilege, click **Actions**  and select **Delete**.
2. You are prompted to confirm. Click **Yes**.

Managing OAuth Clients

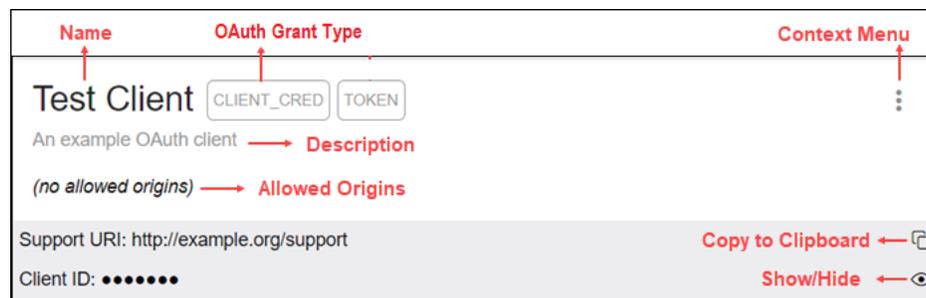
Using OAuth 2.0-based authentication, you can ensure that your RESTful web services are accessed only by specific users or clients.

OAuth 2.0 is a standard Internet protocol that defines flows to provide conditional and limited access to a RESTful API. For more information, see [OAuth-Based Authentication](#).

You can create, edit and delete OAuth Clients in the OAuth Clients page.

To navigate to the OAuth Clients page, from the REST Overview page, click **Clients** in Objects, or from the menu in the header, select **Security** and then select **OAuth Client**.

The OAuth Client attributes displayed by default in card view are shown in the following figure.



To create an OAuth client, see [Creating an OAuth Client](#).

The actions available in the context menu are:

- **Edit:** See [Editing an OAuth Client](#)
- **Export:** See [Exporting an OAuth Client](#)
- **Delete:** See [Deleting an OAuth Client](#)
- **Get Bearer Token:** This option is displayed only if the client is created with a non-encrypted client secret. It provides the access token to call the RESTful service for Client Credentials and Implicit OAuth grant types. See [Creating an OAuth Client Using the Client Credentials Grant Type](#)
- **Manage Secrets:** Applicable for Client Credentials and Auth Code grant types. See [Managing Secrets](#)
 - **Rotate:** Removes the existing client secret and generates a new one for the OAuth client.

- **Revoke:** Removes the client secret associated with the OAuth client.
- **Auth Details:** Displays the Unique Value and Authorization URI for the Auth Code OAuth grant type. See [Creating an OAuth Client Using the Auth Code Grant Type](#)
- [Creating an OAuth Client](#)
Creates the OAuth Client and grants the required roles and privileges.
- [Editing an OAuth Client](#)
This section describes how to edit an OAuth Client.
- [Deleting an OAuth Client](#)
This section describes how to delete an OAuth Client.
- [Exporting an OAuth Client](#)
This section describes how to export an OAuth Client.
- [Managing Secrets](#)
After a client secret is generated for an OAuth client, you can rotate or revoke the secret when required. The Rotate and Revoke options are available in the specific OAuth client's context menu.

See Also

Tutorial for Protecting and Accessing Resources

Creating an OAuth Client

Creates the OAuth Client and grants the required roles and privileges.

1. In the OAuth Clients page, select **Create OAuth Client**.
2. Enter the following fields. The fields with an asterisk (*) are mandatory:

Client Definition tab

- **Grant type:** Select the authorization grant type. The options are Client_Cred, Auth Code, or Implicit.
For more information about these grant types, see OAuth Flows in *Oracle REST Data Services Developer's Guide*.
- **Name:** Name of the client.
- **Description:** Description of the purpose of the client.
- **Redirect URI:** Enter the client-controlled URI to which redirect containing an OAuth access token or error will be sent.
- **Support URI:** Enter the URI where end users can contact the client for support.
Example: `http://www.myclientdomain.com/support/`
- **Support Email:** Enter the email where end users can contact the client for support.
- **Logo:** Upload your logo in JPG, BMP or SVG file format. Ensure that the logo is less than 100 KB in size.
- **Roles:** Select roles to be granted.
- **Allowed Origins:** Add the list of URL prefixes.
- **Privileges:** Select privileges that the client wants to access.

Show code: Select this option to view the PL/SQL code equivalent of the Create OAuth Client panel. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create OAuth Client panel.

3. Click **Create**.

The OAuth Client registered is displayed on the OAuth Clients page.

A preview of the client secret is displayed. Secrets are applicable only for Client Credentials and OAuth Code grant types. Copy the secret value as this is the only instance when it is shown. If you forget the client secret, you can use the Rotate Secret action to change it. See [Managing Secrets](#).

The Client ID value represents the secret credential for the OAuth client. Click **Show/Hide**  to see the values.

Test the secured REST service endpoint using a REST client or the cURL command line tool.

 **See Also**

Tutorial for Protecting and Accessing Resources in *Oracle REST Data Services Developer's Guide*

Editing an OAuth Client

This section describes how to edit an OAuth Client.

1. In the OAuth Clients page, for the specific client, click **Actions**  and select **Edit**.
2. Edit the required fields and click **Save**.

For a description of the fields, see [Creating an OAuth Client](#).

Deleting an OAuth Client

This section describes how to delete an OAuth Client.

1. In the OAuth Clients page, for the specific client, click **Actions**  and select **Delete**.
2. You are prompted to confirm. Click **Yes**.

Exporting an OAuth Client

This section describes how to export an OAuth Client.

1. In the OAuth Clients page, for the specific client, click **Actions**  and select **Export**.
2. In the OAuth Client panel, click the **Copy** icon or **Download** to copy or download the OAuth Client information.

Managing Secrets

After a client secret is generated for an OAuth client, you can rotate or revoke the secret when required. The Rotate and Revoke options are available in the specific OAuth client's context menu.

Note

With the deprecation of OAUTH and OAUTH_ADMIN PL/SQL packages (see [ORDS_SECURITY PL/SQL Package Reference](#)), the OAuth client secret creation process changes for Client Credentials and Auth Code grant types.

For OAuth clients that have been created with a non-encrypted secret, the label `Legacy` is displayed on the card.

Rotate a Client Secret

Removes the existing secret and creates a new one. This is useful when you cannot remember the existing client secret value.

- If a client secret exists for the OAuth client, click **Rotate** to remove the existing one and generate a new client secret.
- If a client secret is revoked and there is no secret value assigned, click **Register** to generate a client secret for the OAuth client.

Revoke a Client Secret

Removes the existing client secret.

Select **Revoke sessions** to remove all existing client sessions.

Examples

This section provides a few examples on creating an OAuth client with different grant types.

Topics:

- [Creating an OAuth Client Using the Client Credentials Grant Type](#)
- [Creating an OAuth Client Using the Auth Code Grant Type](#)
- [Creating an OAuth Client Using the Client Credentials Grant Type](#)
This section describes how to create an OAuth Client using the Client Credential grant type.
- [Creating an OAuth Client Using the Auth Code Grant Type](#)
This section describes how to create an OAuth Client using the Auth Code grant type.

Creating an OAuth Client Using the Client Credentials Grant Type

This section describes how to create an OAuth Client using the Client Credential grant type.

Create an OAuth Client for the created module "example" in [Example: Inserting a Record using a POST Handler](#). The endpoint for the RESTful service is `http://xyz.us.comp.com:1234/ords/pdbdba/example/emp/`.

Prerequisites

Create a role named HR Admin. See [Creating a Role](#)

Create a privilege named Example.HR. See [Creating a Privilege](#)

1. In the OAuth Clients page, click **Create OAuth Client**.
2. Enter the following fields:
 - In the Grant type field, select **CLIENT_CRED**.
 - Enter Name, Description, Redirect URI, Support URI and Support Email for your OAuth Client.
 - In the Roles tab, add the created role (HR Admin).
 - In the Privileges tab, add the created privilege (Example.HR).
 - Click **Create**.

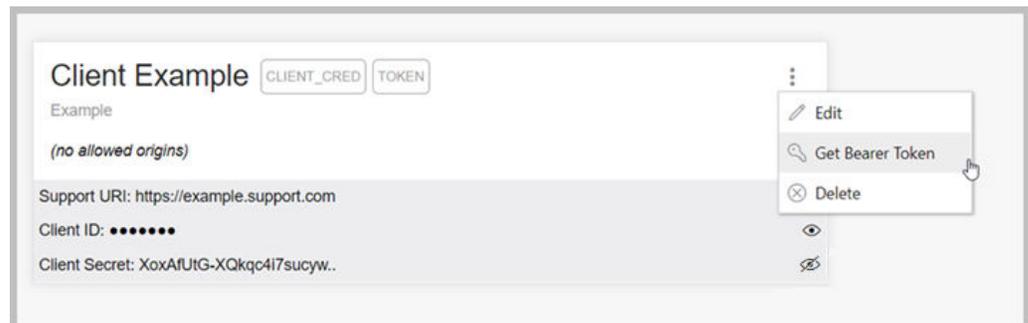
The new OAuth Client card appears on the OAuth Clients page.

3. Using cURL, test the service endpoint without OAuth credentials.

```
curl --location --request POST --header "Content-Type: application/json"
--data '{"DEPTNO": 55, "DEPTNAME": "Sales", "DEPTLOC": "Australia" }'
'http://xyz.us.comp.com:1234/ords/pdbdba/example/emp/'
```

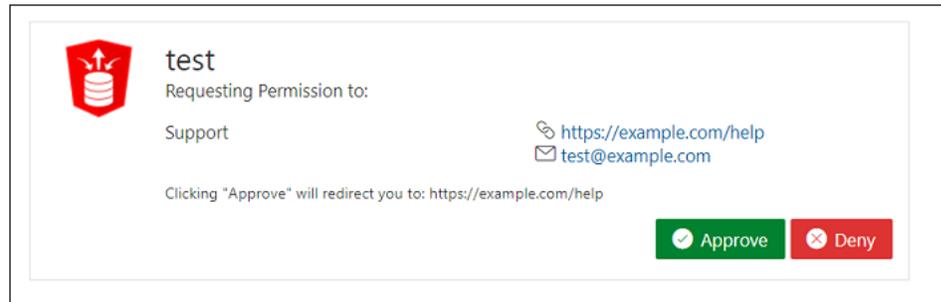
You see an error.

4. Test the endpoint with OAuth credentials:
 - a. To get an access token, select **Get Bearer Token** from the context menu in the OAuth Client card.



The OAuth Token dialog appears. Use **Copy to Clipboard**  to copy the token.

4. In the Authorization URI field, click the **Open in New Tab** icon. A new window is displayed with an `Unauthorized` error message along with a Sign In link.
5. Click **Sign in** and enter your login credentials again.
6. You are prompted to approve the request to access the protected URI. Click **Approve**.



5

The JSON Page

Note

Available for Oracle Database release 19c and later releases and only if you signed in as a database user with the **SODA_APP** role.

Use the JSON page in Database Actions to view and manage JSON collections, search for collection items using Query-by-Example (QBE), create JSON search indexes and display data guide diagrams for collections.

To navigate to the JSON page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **JSON**.
- Click **Selector**  to display the navigation menu. Under **Development**, select **JSON**.

Topics

- [About the JSON User Interface](#)
- [Managing JSON Collections](#)
- [About Querying Documents in a Collection](#)
- [Creating Indexes for JSON Collections](#)
- [Creating Relational Views of JSON Documents](#)
- [Viewing the JSON Data Guide Diagram for a Collection](#)

- [About the JSON User Interface](#)
The JSON user interface consists of the left pane for listing and searching saved collections and the right pane for viewing and managing documents in a collection.
- [Managing JSON Collections](#)
You can add, view or drop collections, or browse, add, edit and delete JSON documents in a collection.
- [About Querying Documents in a Collection](#)
You can search for one or more documents in a collection by using a filter specification or Query-by-Example (QBE).
- [Creating Indexes for JSON Collections](#)
- [Creating Relational Views of JSON Documents](#)
- [Viewing the JSON Data Guide Diagram for a Collection](#)
The diagram view displays the JSON data guide for a collection as a hierarchical structure using a format similar to entity-relationship diagrams.

See Also

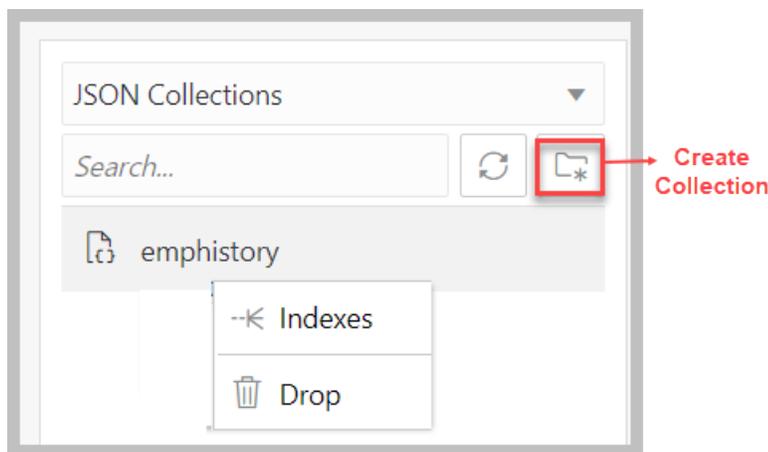
[SODA Overview in Oracle AI Database Introduction to Simple Oracle Document Access \(SODA\)](#)

About the JSON User Interface

The JSON user interface consists of the left pane for listing and searching saved collections and the right pane for viewing and managing documents in a collection.

Listing and Searching JSON Collections

The following figure shows the main items in the left pane of the JSON page.



Select the appropriate option in the drop-down list to display saved JSON collections or recently accessed JSON collections.

The search functionality is not case-sensitive, retrieves all matching entries, and does not require the use of wildcard characters.

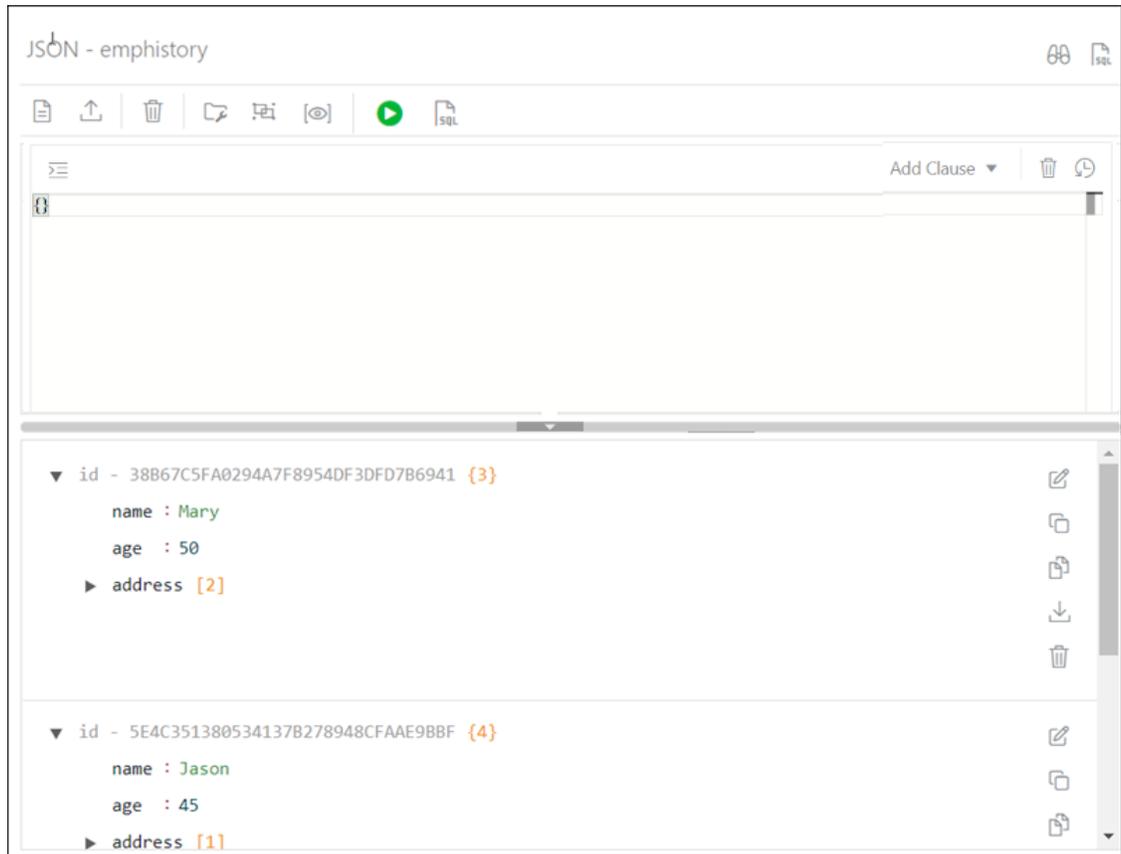
Click the Create Collection icon to create a new collection. See [Creating a Collection](#)

Right-click a collection name to open the context menu. The available options are:

- **Indexes:** Enables you to view existing JSON indexes and create search, functional or spatial indexes.
- **Drop:** Removes the collection from the database completely.

Viewing the Contents of a Collection

When a specific collection is selected in the left pane, the JSON documents that belong to the collection are displayed in the lower part of the right pane. The top part of the right pane contains the JSON editor where you can run queries for filtering or sorting the documents.



The two icons in the right corner are:

- **Tour** : Starts the JSON tour, which provides information about the features available.
- **Go to SQL** : Navigates to the SQL page.

The icons in the toolbar are:

- **New JSON Document** : Adds a new document to the collection. See [About Adding or Editing a JSON Document](#)
- **Import Document(s)** : Imports one or more existing JSON files from your local computer into the collection.
- **Delete All Documents in the list** : Deletes all JSON documents in the collection that match the current QBE search string. If the current string is {}, then all the documents in the collection are deleted.
- **Collection Details** : Enables you to view collection properties, JSON data guide (if created) and related statistics if they are gathered, size of search index, and page for managing JSON indexes.
- **Diagram** : Displays the the JSON data guide as a diagram in a hierarchical format. See [Viewing the JSON Data Guide Diagram for a Collection](#)
- **New Collection View** : Creates relational views of documents in a collection.

- **Run Query** : Filters documents using the QBE condition entered in the JSON editor. See [About Querying Documents in a Collection](#)
- **View Collection SQL** : Displays the JSON collection in SQL format. You can execute the SQL code, download it or copy to clipboard.

Each JSON document has the following icons:

- **Edit Document** : Edits the JSON document. See [About Adding or Editing a JSON Document](#)
- **Clone Document** : Creates a clone of the document.
- **Copy Document** : Copies the document to the clipboard.
- **Export Document** : Downloads the document as a .JSON file.
- **Delete Document** : Deletes the document.

Managing JSON Collections

You can add, view or drop collections, or browse, add, edit and delete JSON documents in a collection.

- **Create a Collection:** See [Creating a Collection](#)
- **Add or Edit Documents in a Collection:** See [About Adding or Editing a JSON Document](#)
- **View Documents in a Collection:** Select the collection name in the left pane. The documents in the collection are displayed in the right pane.
- **View Collection Details:** Select the specific collection in the left pane, and then click **Collection Details**  in the right pane toolbar to view collection properties, JSON data guide (if created) and related statistics if gathered, the size of search index, and the page for managing JSON indexes.
- **Drop a Collection:** Right-click a collection name in the left pane to open the context menu. Select **Drop** to remove the collection from the database completely.
- [Creating a Collection](#)
This section describes how to create a collection.
- [About Adding or Editing a JSON Document](#)
You can add and edit JSON documents using the JSON editor.

Creating a Collection

This section describes how to create a collection.

1. When you navigate to the JSON page:
 - If there are no collections created, you see the JSON home page. Click **Create Collection**.
 - If there are existing collections available, then in the left pane, click **Create Collection** .
2. Enter the collection name, which is case sensitive.
3. If the **MongoDB Compatible** option is selected, then the document IDs will be Embedded IDs.

- Document IDs are part of the JSON document (field name is "_id"). If an _id field is missing, it is generated and added to the JSON document.
- This collection is MongoDB compatible. MongoDB drivers connecting to the MongoDB API for the Oracle Database can read and write to this collection.
- To use these collections from SODA, you need a SODA driver that supports the EMBEDDED_OID key assignment method.
- This method is documented as EMBEDDED_OID.
- The name of the content column (the column where JSON documents are stored) in the underlying table is DATA.

Out of line IDs:

- SODA always generates the ID for a new document.
 - The ID is not embedded in the JSON document but stored separately.
 - MongoDB drivers connecting to the MongoDB API for the Oracle Database can only read from this collection.
 - This method is documented as UUID.
 - The name of the content column in the underlying table is JSON_DOCUMENT.
4. Click **Create** to create the collection.
- Click **Cancel** or press the **Esc** key to cancel the collection.

See Also

[SODA Collection Metadata Components in Oracle AI Database Introduction to Simple Oracle Document Access \(SODA\)](#)

About Adding or Editing a JSON Document

You can add and edit JSON documents using the JSON editor.

To add a document, click **New JSON Document**  in the right pane toolbar for a specific collection. In the JSON editor, you can copy and paste the JSON document or use the Import icon to import the JSON document.

To edit a document, click **Edit Document**  in the document card.

The following figure shows an open document in the JSON editor.

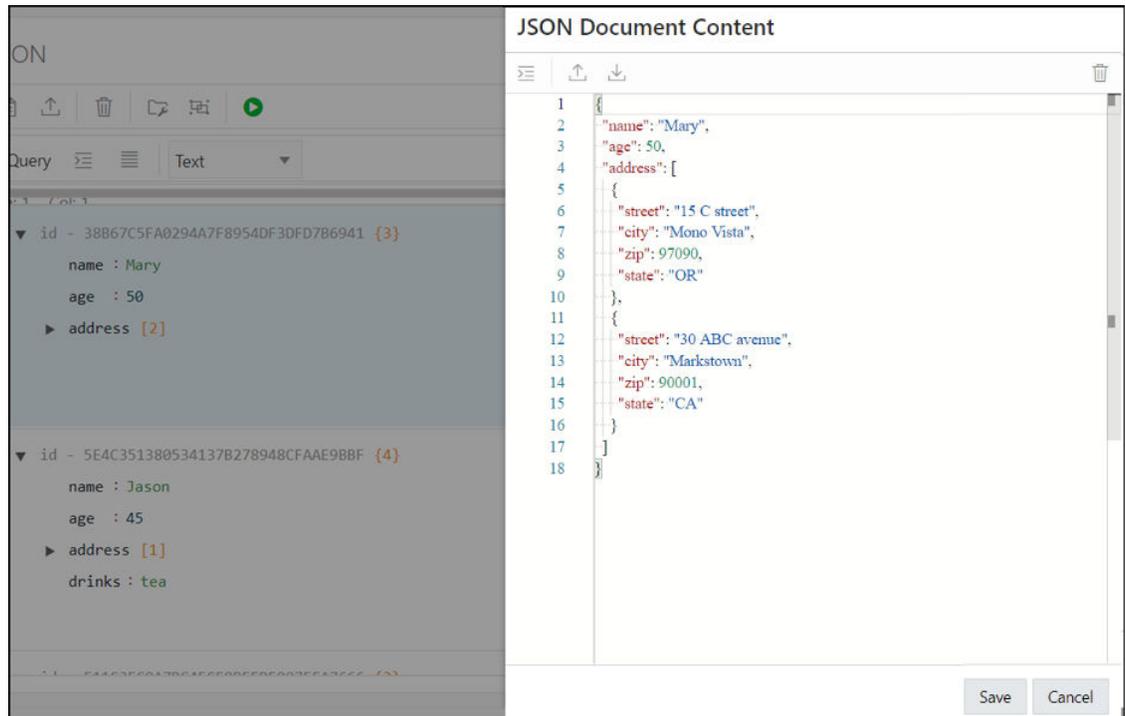
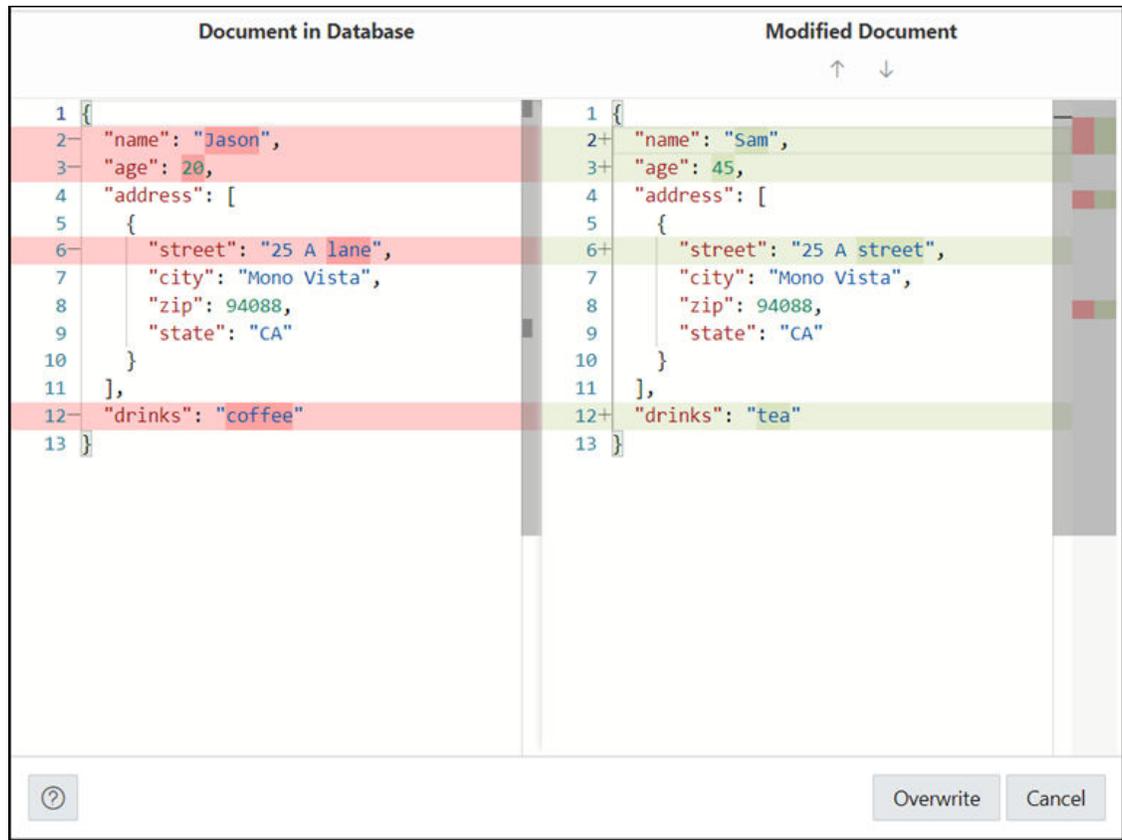


Figure 5-1 Differences in the Document Displayed



About Querying Documents in a Collection

You can search for one or more documents in a collection by using a filter specification or Query-by-Example (QBE).

A QBE is a pattern expressed in JSON. You use it to select, from a collection, the JSON documents whose content matches it, meaning that the condition expressed by the pattern evaluates to true for the content of only those documents. For more information about QBEs and how to use them, see *Overview of SODA QBEs in Oracle AI Database Introduction to Simple Oracle Document Access (SODA)*.

For a specific collection, enter the QBE string in the JSON editor. For example, to select documents where the name is Mary, enter `{"name":"Mary"}` and then click **Run Query**, as shown in [Figure 5-2](#). The results of the query are displayed in the lower right pane.

The editor offers a comprehensive list of commands available through the Command Palette. To open the Command Palette, press **Ctrl+Shift+P**. For a list of keyboard shortcut keys, see [Keyboard Shortcuts](#).

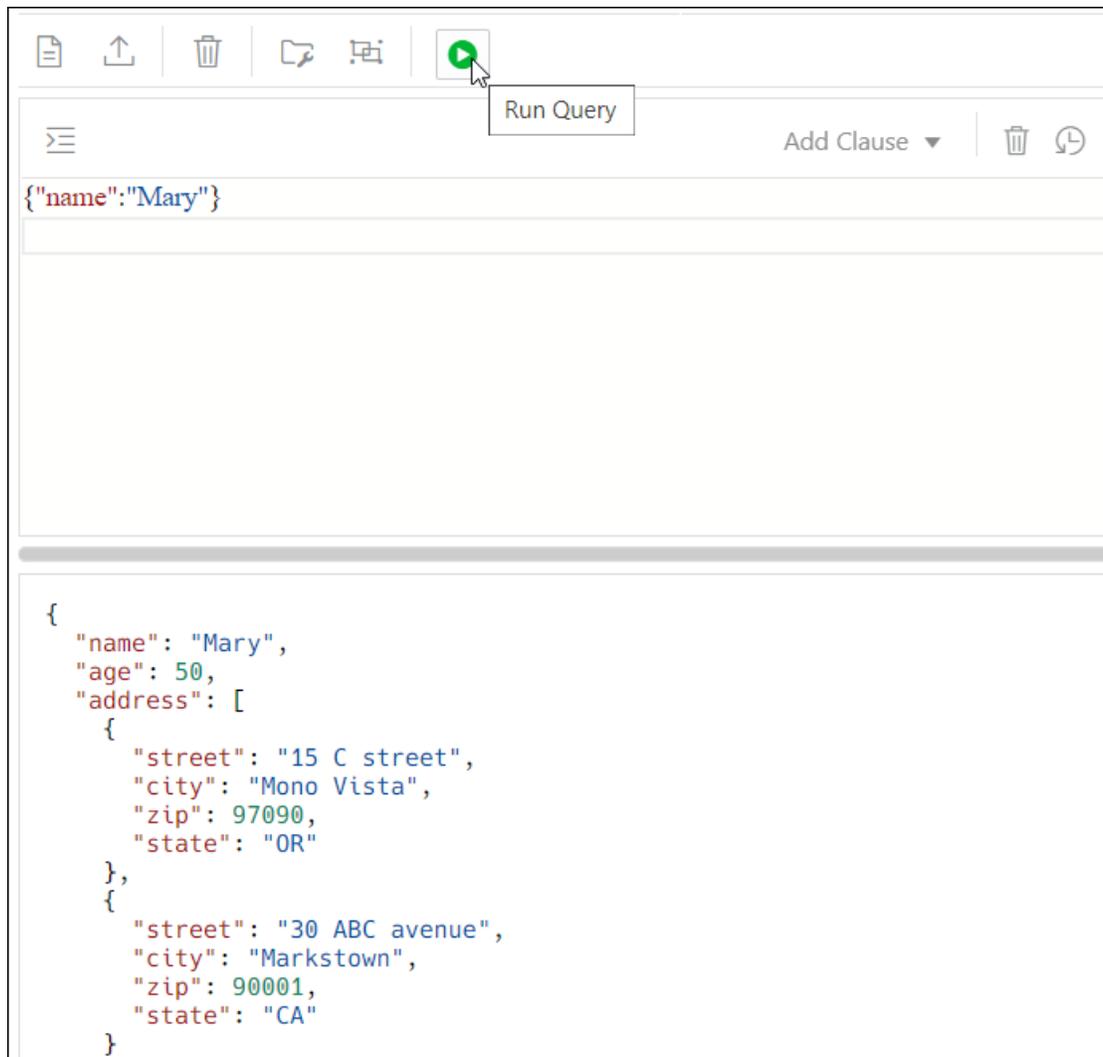
An error in the query is signified by a red dot in the left gutter and a squiggle line beneath the specific text. When you hover over it, you see a pop-up displaying possible fixes for resolving the error.

You can set editor preferences using the **Preferences** option in the top-right user drop-down list on the header. Some of the available options are Theme (Light, Dark and High contrast dark), Font size and family, Tab size, Word wrap, Ruler, Line numbers and so on.

The icons above the editor are:

- **Format JSON data** : Enables indentation and line feeds for the QBE string.
- **Add Clause**: Adds a formatted template of the \$orderby or \$patch clause to the QBE string.
- **Clear** : Clears the current QBE string.
- **History** : Retrieves previous QBE search strings.

Figure 5-2 QBE String in JSON Editor



The QBE expression must be a valid JSON object and can contain \$query and \$orderby or \$patch clauses. The QBE expression is treated as a \$query clause if there are no clauses defined. Starting with the \$query clause, add the \$orderby or \$patch clause later using the **Add Clause** list. The content is transformed and a template for the \$orderby and \$patch clause is provided. You need to set correct values in the templates. See [Using the In-Context Autocomplete Feature in the JSON Editor](#)

For example, this is a simple filtering query:

```

1  {
2    "site_admin": true
3  }

```

After the \$orderby clause is added using Add Clause, you see the following entry:

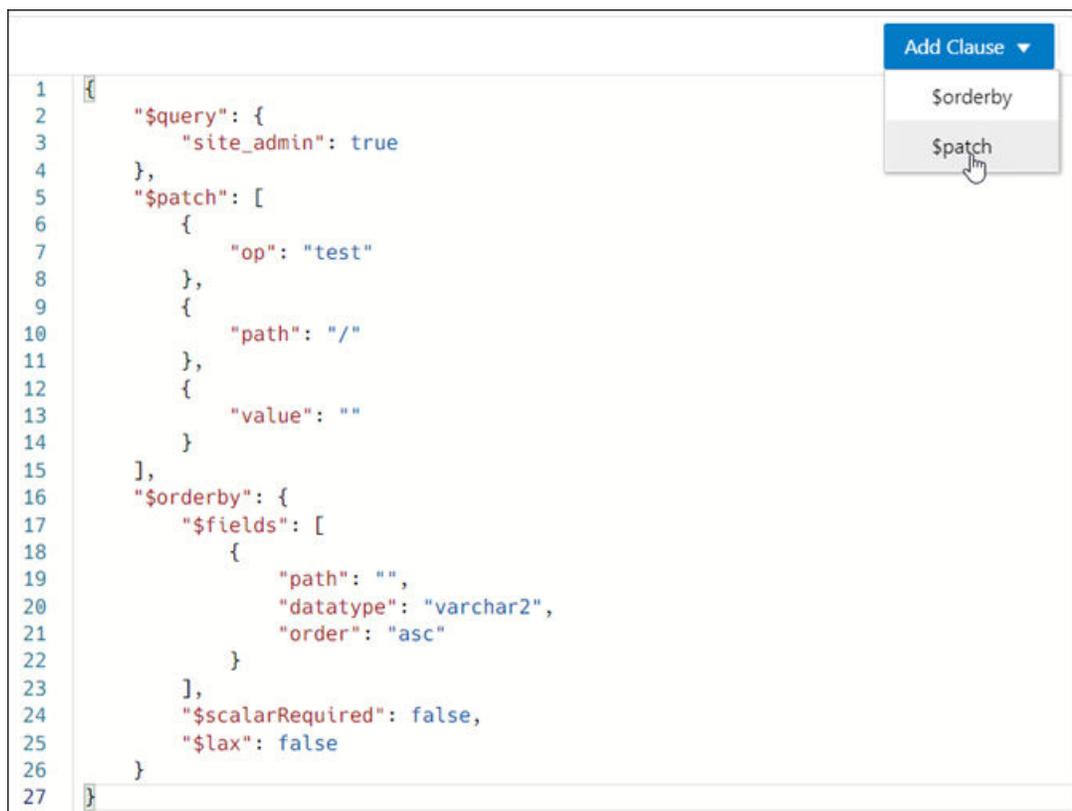


```

1  {
2    "$query": {
3      "site_admin": true
4    },
5    "$orderby": {
6      "$fields": [
7        {
8          "path": "",
9          "datatype": "varchar2",
10         "order": "asc"
11       }
12     ],
13     "$scalarRequired": false,
14     "$lax": false
15   }
16 }

```

After the \$patch clause is added, the query changes to:



```

1  {
2    "$query": {
3      "site_admin": true
4    },
5    "$patch": [
6      {
7        "op": "test"
8      },
9      {
10       "path": "/"
11     },
12     {
13       "value": ""
14     }
15   ],
16   "$orderby": {
17     "$fields": [
18       {
19         "path": "",
20         "datatype": "varchar2",
21         "order": "asc"
22       }
23     ],
24     "$scalarRequired": false,
25     "$lax": false
26   }
27 }

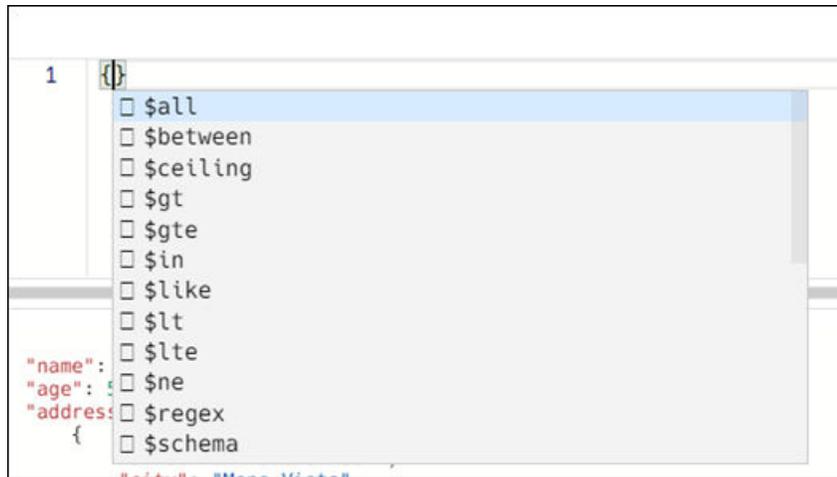
```

- [Using the In-Context Autocomplete Feature in the JSON Editor](#)

If you press **Ctrl+Space**, the editor provides you with a context-aware list of options from which you can select and autocomplete at the insertion point.

Using the In-Context Autocomplete Feature in the JSON Editor

If you press **Ctrl+Space**, the editor provides you with a context-aware list of options from which you can select and autocomplete at the insertion point.



The following types of information appear in the list:

- **Filter comparison clauses**

A template is available for each clause and it is inserted at the cursor position. For example, when you select the `$between` clause from the autocomplete list, you see the following entry in the editor:



where `age` is the property name and `49` and `70` are values. These are parameters that you can edit and they appear highlighted. You can type or use the autocomplete help to edit the property name. Press the **Tab** key to move to the next parameter.

- **Property names from JSON documents in the collection**

There are two sources for property names from JSON documents:

- If a search index is created with support for data guide, then the JSON data guide from the database dictionary is scanned for property names.
- Property names that are collected from viewed or edited documents are presented. It is possible that the property names are a subset from the whole namespace.

Based on the type of property, the related template is inserted and the cursor is positioned at the expected place for insertion. The following is a list of property types, their templates and the corresponding cursor position for each:

- Object

```
"ShippingInstructions.Address": {
  <cursor_here>
}
```

- Array

```
"LineItems": [<cursor_here>]
```

- String

```
"LineItems[*].Part.Description": "<cursor_here>"]
```

- Number

```
"PONumber": 0
```

- Boolean

```
"site_admin": true
```

When the cursor is between double quotes (""") and autocompletion is activated, then only the property name is inserted without templates or additional double quotes.

Creating Indexes for JSON Collections

You can create indexes for JSON collections in the JSON page.

 **See Also**

Indexes for JSON Data in *Oracle Database JSON Developer's Guide*

Open the Indexes Pane

In the JSON left pane, right-click the collection, and select **Indexes**.

The Indexes pane lists the existing indexes for the collection. Select the index row to display more information.

Employees indexes

TABLE JSON

Name	Type	Schema
emp_name	functional	HR

```

{
  "name": "emp_name",
  "schema": "HR",
  "tableName": "Employees",
  "tableSchemaName": "HR",
  "indexNulls": true,
  "unique": false,
  "lax": false,
  "scalarRequired": false,
  "fields": [
    {
      "path": "name",
      "dataType": "VARCHAR2",
      "maxLength": 2000,
      "order": "ASC"
    }
  ],
  "type": "functional"
}

```

The icons at the top are **Add JSON Index**, **Edit JSON Index**, and **Delete JSON Index**.

The properties of the selected index appear in JSON format below the listed indexes. Select **JSON** from the TABLE - JSON option to view all indexes in JSON presentation.

Create an Index

1. Click the + **New JSON Index** icon. The New Index pane appears.
2. Enter the following fields to create an index:

Name: Type:

Unique Index Nulls Path Required

Properties

Enter * to display known properties

Composite index Advanced

Path	
\$.name	<input checked="" type="checkbox"/>
\$.age	<input type="checkbox"/>

- **Name:** Enter a name for the index.
- **Type:** Select the index type from the drop-down list. The different options are Functional, Spatial and Search. Based on the index type selected, the corresponding options appear.

- For a functional type index, the fields to enter are:
 - **Unique:** Select this option to make all indexed values unique.
 - **Index Nulls:** Select this option to use the index in Order By queries.
 - **Path Required:** Select this option if the path must select a scalar value, even a JSON null value.
 - **Properties:** Type the property that you want to index on, or Type * to display all available document properties in the collection. To select a property, select the checkbox in the respective row.

Note

You cannot index properties in arrays.

- **Composite Index:** Select this option if you want to use more than one property.
- **Advanced:** Select this option to change the storage properties of the indexed property. For each property, you can change the type (varchar2, number, date or timestamp), maximum length for indexing (for character properties), and sort order.
- For search index, the options are:
 - **Dataguide off-on:** Select **on** to create JSON data guide for collection.
 - **Text Search off-on:** Select **on** to index all properties in documents to support full-text search based on string equality (every property is treated as string)
 - **Range Search off-on:** Select **on** to support range search when string-range search or temporal search (equality or range) is required.
- Spatial index is used to index GeoJSON geographic data. The selected property should be of GeoJSON type. See [Using GeoJSON Geographic Data](#)

For spatial index, the options are:

- **Path Required:** Select this option if the path must select a value, even if it is a JSON null value.
- **Lax:** Select this option if the targeted field does not need to be present or does not have a GeoJSON geometry object as its value.

Note

You cannot enable **Path Required** and **Lax** at the same time.

3. Click **Create**. A notification is displayed indicating that the index is created and the Indexes pane is populated.

Creating Relational Views of JSON Documents

You can create relational views of JSON documents.

1. In the JSON page, click  **New Collection View** in the right pane.
2. Enter the following fields:
 - **View Name:** Enter a name for the view.

- **Select columns to add:** Click in this field to view the list of available properties and to select the required properties, adding them one at a time. Alternatively, select **Add All** to add all objects and arrays with all their properties. Select one or several properties and move them up or down to position them in the resulting view. Objects and arrays are moved or deleted together with all their properties.

You can edit the Column Name attribute for each column, and for string types, you can also edit the Length attribute. Precision and scale can be set for numeric properties, for example, (10,2).

Path	Type	Length
name	string	5
address.street	string	13
address.city	string	10
address.zip	number	
address.state	string	2

3. Click **Test Query** to review the DDL statements before creating the view.

Click **Create** to create the view, or click **Definition** to return to the previous screen and make changes to the corresponding attributes.

The created view is now available in the SQL page.

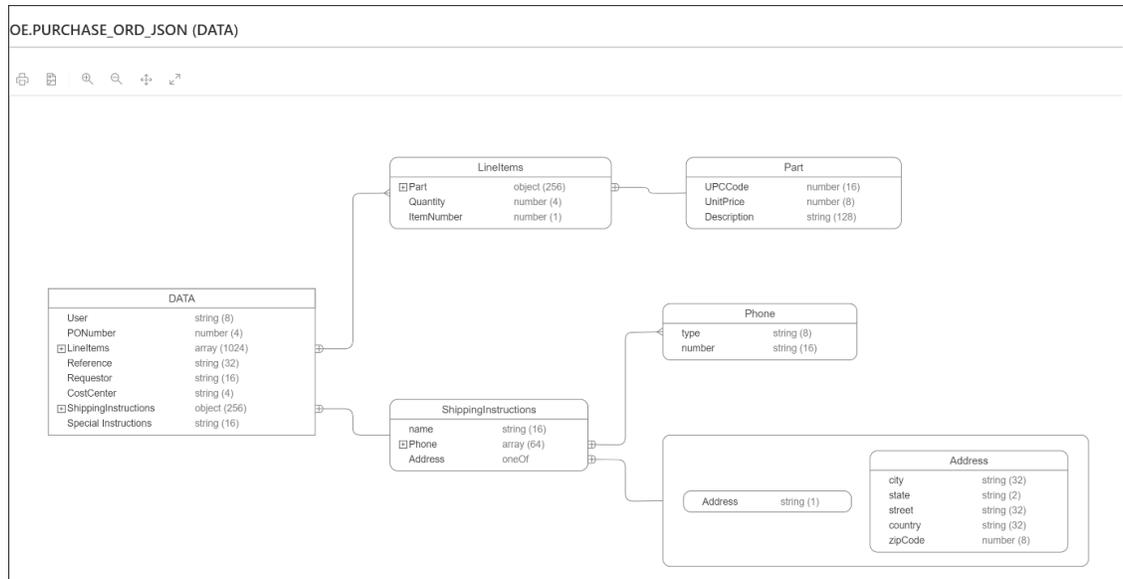
Viewing the JSON Data Guide Diagram for a Collection

The diagram view displays the JSON data guide for a collection as a hierarchical structure using a format similar to entity-relationship diagrams.

The JSON data guide represents the JSON schema for documents that have a column with JSON content.

In the diagram, arrays are presented as one-to-many relationships, contained objects as one-to-one relationships, and "oneOf" constructs as a box that surrounds possible choices.

Click **Diagram**  in the right pane toolbar to display the JSON data guide diagram for a specific collection.



The icons in the toolbar are Print Diagram, Save to SVG format, Zoom In, Zoom Out, Fit Screen and Actual Size.

See Also

JSON Data Guide in *Oracle AI Database JSON Developer's Guide*

6

The Charts and Dashboards Page

Charts enable you to create charts from the database. The chart is constructed using the input SQL command. Dashboards enable you to group charts together to create reports.

To navigate to the Charts and Dashboards page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **Charts**.
- Click **Selector**  to display the navigation menu. Under Development, select **Charts**.

Topics

- [Overview](#)
- [Creating or Editing a Chart](#)
- [Example: Creating and Editing a Chart](#)
- [Creating or Editing a Dashboard](#)
- [Overview](#)
This section describes the layout of the main Charts and Dashboards page.
- [Creating or Editing a Chart](#)
You can create or edit a chart to visualize data using SQL commands.
- [Example: Creating and Editing a Chart](#)
This section provides an example of creating and editing a chart.
- [Creating or Editing a Dashboard](#)
This section describes the steps to create or edit a dashboard.

Overview

This section describes the layout of the main Charts and Dashboards page.

To navigate to the Charts and Dashboards page, do either of the following:

The upper right corner has two icons:

- **Tour**: Provides a guided tour of charts and dashboards highlighting salient features and providing information that is useful if you are new to the interface.
- **Create**: Creates a new chart or dashboard. See [Creating or Editing a Chart](#) and [Creating or Editing a Dashboard](#)

This page consists of two sections:

- **Charts and Dashboards**: Displays the total number of charts and dashboards created. Click the respective card to view the charts or dashboards created in the section below.
- Displays the charts or dashboards created in the default card format.

The actions available for a chart or dashboard are:

- **View Chart/Dashboard**: To view the chart or dashboard on a different page.

- **Edit:** To make changes to the chart or dashboard.
- **Unpublish:** To make the chart or dashboard unavailable for use.
- **Go to module definition:** To show the module definition in the REST Modules page.
- **Delete:** To delete the chart or dashboard.

Creating or Editing a Chart

You can create or edit a chart to visualize data using SQL commands.

To create a new chart, click **Create Chart**.

To edit a chart, click the context menu icon and select **Edit**.

Enter the following details:

1. **Name:** Enter a name for the chart.
2. **Description:** Enter a description for the chart.
3. **Protected by Privilege:** Select an available privilege from the drop-down list to enable only those with this privilege to securely access the chart.
4. **Published:** When enabled, you can share and access the chart externally.

Note

You can also protect a chart by protecting the corresponding module using the REST pages. For more information, see [Editing a Resource Module](#). When a chart is protected, a user must authenticate before viewing the chart.

5. Click **Next**.
6. Enter the SQL statement in the command editor on the left. Ensure that you do not end the SQL statement with a semicolon.

The chart settings are automatically updated in the **Chart Definition** section on the right.

7. Edit/Update the chart settings in the **Chart Definition** section. The fields in the mapping section varies according to the **Chart Type** selected.

When the **Chart Type** selected is one of the following: *Area Chart*, *Bar Chart*, *Line Chart*, or *Line with Area Chart*, the mappings displayed are as follows:

- **Orientation:** Choose between horizontal and vertical orientation types from the drop-down list.
- **X axis label and Y axis label:** Optionally enter labels for X axis and Y axis.
- **Label:** The label is auto-generated from the SQL statement which represents the X axis.
- **Value:** The Value is auto-generated from the SQL statement which represents the Y axis (the values to be plotted on the chart).
- **Refresh Columns:** Refresh the data of all the columns selected in the SQL statement.
- **Coordinate System:** Choose between Cartesian and Polar coordinate systems from the drop-down list.
- **Sorting:** Choose a sorting type (ascending, descending, off) for plotting the values. If selected, the Sorting value overrides the order specified by the `ORDER BY` clause in the

SQL statement. For a large amount of data, Oracle recommends using the `ORDER BY` clause to sort data.

When the **Chart Type** selected is one of the following: *Pie Chart*, *Pyramid Chart*, or *Funnel Chart*, the mappings displayed are as follows:

- **Style:** Select from either 2D or 3D styles.
- **Series ID:** The Series ID is auto-generated from the SQL statement which represents the category to be plotted.
- **Value:** The Value is auto-generated from the SQL statement which represents the values to be plotted for each category.

Preview/Auto-preview: Click preview to view the changes made to the chart. If Auto-preview is checked, the changes to the chart are refreshed automatically.

8. The **Chart** tab at the bottom displays the chart.
9. The **Data** tab at the bottom displays the data resulting from the SQL command.
10. Click **Create** to create the chart, or click **Save** when editing the chart.

Note

Due to browser resource restrictions, the preview of charts is limited to 3000 rows from your SQL query results. However, once you create a chart, all the data is rendered in the standalone view.

Example: Creating and Editing a Chart

This section provides an example of creating and editing a chart.

Creating a Chart

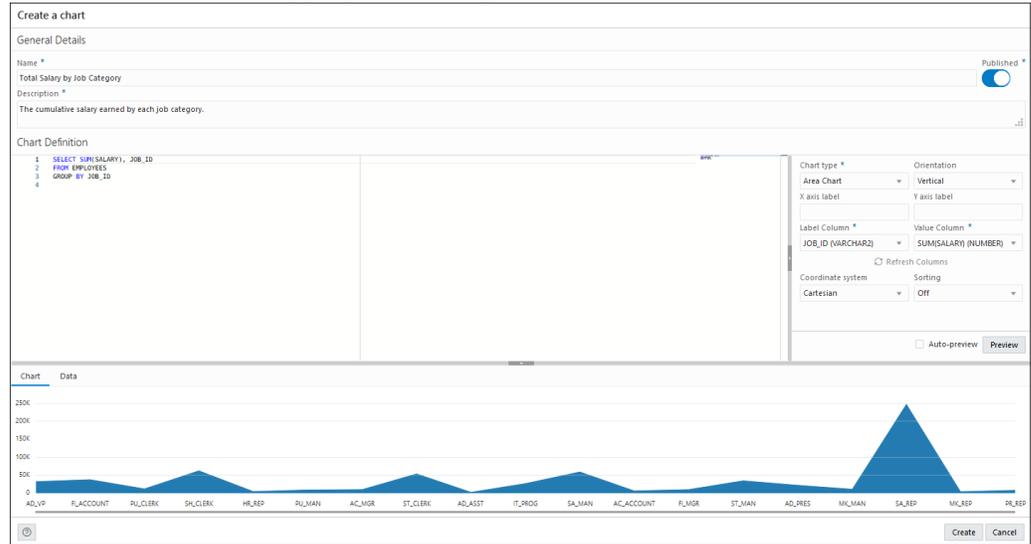
The following example shows how to create a chart to understand the cumulative salary earned by each job category.

1. Navigate to the **Charts** page from Database Actions homepage.
2. Click **Create Chart**.
3. Enter the details of the chart as follows:
 - **Name:** Total Salary by Job Category
 - **Description:** The cumulative salary earned by each job category.
 - **Chart Definition:**
 - **SQL Statement:**

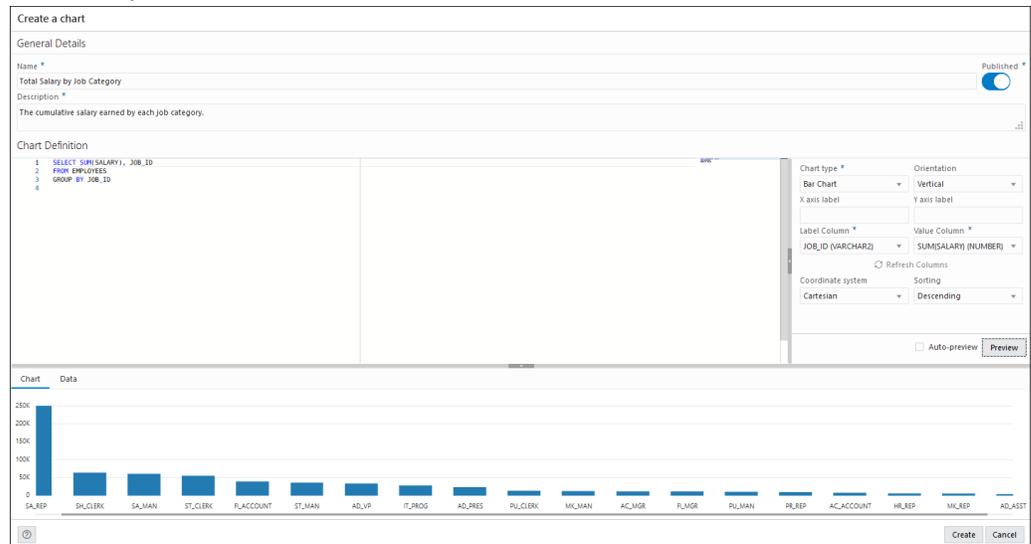
```
SELECT SUM(SALARY), JOB_ID
FROM EMPLOYEES
GROUP BY JOB_ID
```

- **Chart Type:** Area Chart
- **Orientation:** Vertical
- **X axis label:** Job Category

- **Y axis label:** Total Salary
 - **Label:** JOB_ID (VARCHAR2)
 - **Value:** SUM(SALARY) (NUMBER)
 - **Coordinate system:** Cartesian
 - **Sorting:** Off
- Click **Preview** in the Mapping section to preview the chart below.

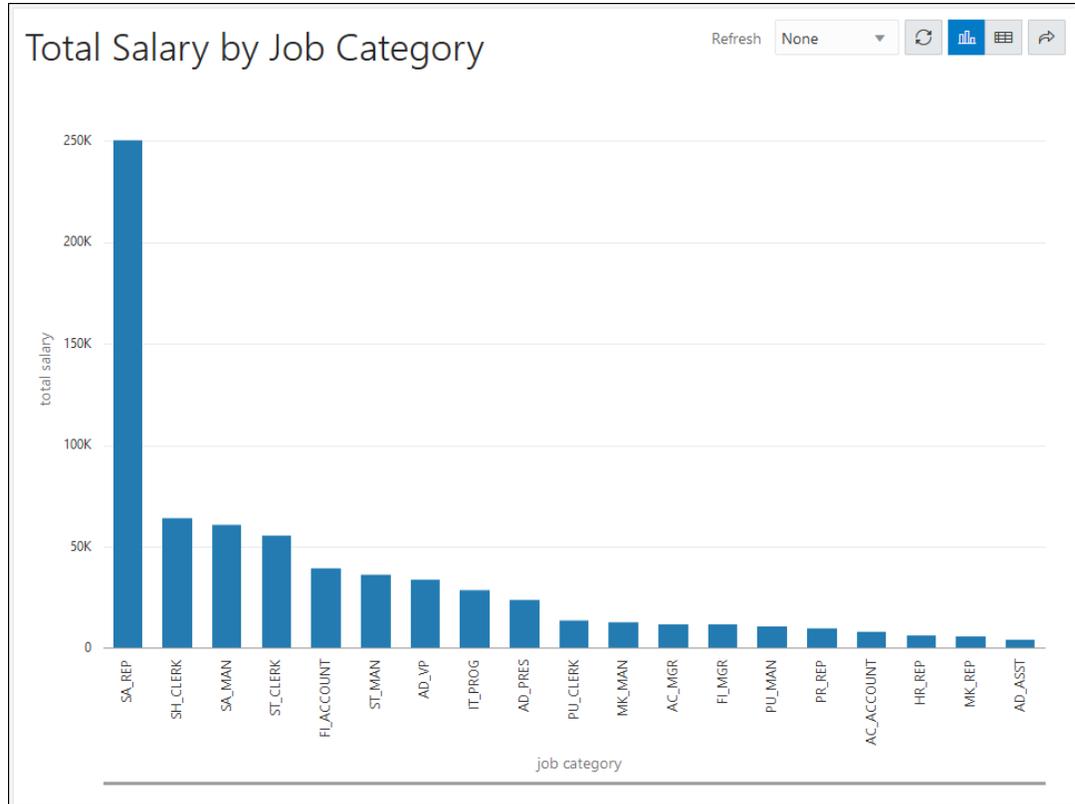


- Change the **Chart Type** to *Bar Chart*, **Sorting** to *Descending* and click **Preview** to view the updated chart.



4. Click **Create** to create the chart. A confirmation notification is displayed and the chart card is visible in the Charts page.

- Click the Selector  and select **View Chart** to view the chart in a new tab.



Editing a Chart

The following example edits the chart created in the previous section.

- Click the Selector  and select **Edit** to edit the chart.

The screenshot shows the 'Charts' management page. At the top, there is a search bar and filters. Below, a chart card is displayed with the title 'Total Salary by Job Category' and a description 'The cumulative salary earned by each job category.' The card includes 'Published' and 'Unprotected' status indicators and a URL. A context menu is open over the card, showing the following options:

- View Chart
- Edit
- Unpublish
- Go to module definition
- Delete

- In the Mapping section, change the **Chart Type** to *Pie Chart* and **Style** to *3D*.

3. Click **Preview** to view the updated chart.

Edit - Total Salary by Job Category

General Details

Name * Total Salary by Job Category Published *

Description * The cumulative salary earned by each job category.

Chart Definition

```

1 SELECT SUM(SALARY), JOB_ID
2 FROM EMPLOYEES
3 GROUP BY JOB_ID
    
```

Chart type * Pie Chart

Series ID Column * JOB_ID (VARCHAR2)

Style 3D

Value Column * SUM(SALARY) (NUMBE...)

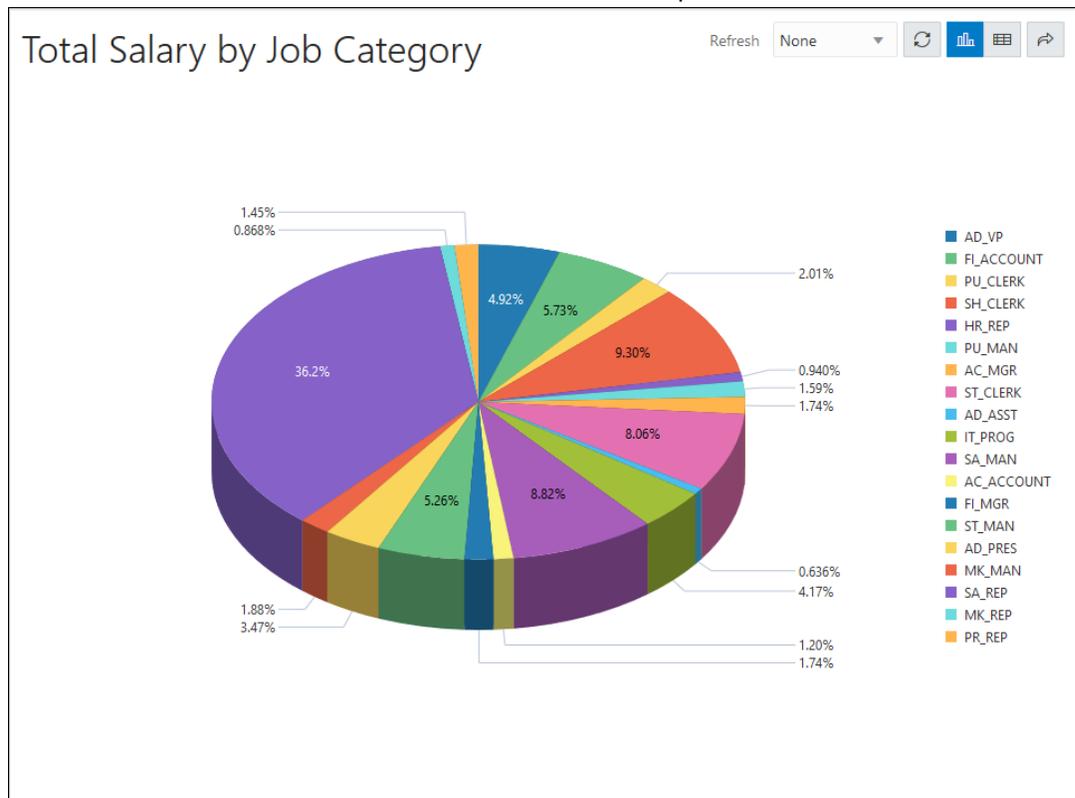
Auto-preview **Preview**

Chart Data

Job Category	Percentage
SH_CLERK	4.9%
HR_REP	36%
AD_VP	5.7%
FL_ACCOUNT	2.0%
PU_CLERK	1.7%
AC_MGR	1.6%
PU_MAN	0.64%
ST_CLERK	1.2%
AD_VP	4.2%
HR_REP	5.3%
AD_VP	1.4%
FL_ACCOUNT	0.87%
PU_CLERK	3.5%
AC_MGR	1.9%
PU_MAN	5.3%

Save **Cancel**

4. Click **Save**.
5. Click the Selector and select **View Chart** to view the updated chart in a new tab.



Creating or Editing a Dashboard

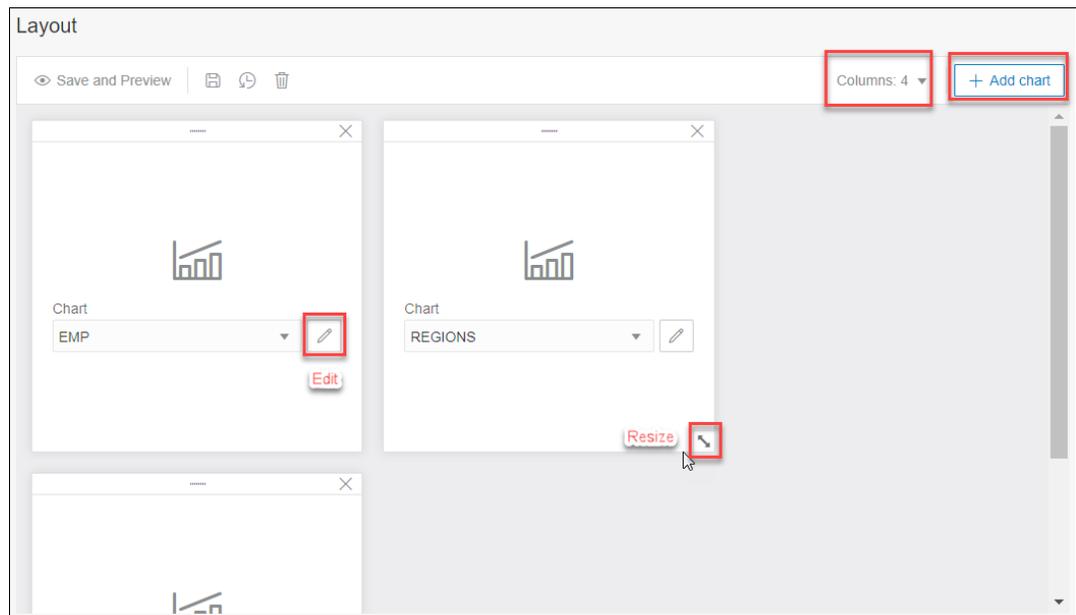
This section describes the steps to create or edit a dashboard.

To create a dashboard, in the Charts and Dashboard page, at the top right, click **Create** and then select **New Dashboard**.

To edit a dashboard, click the Actions icon and select **Edit**.

1. In the Create a Dashboard pane, enter or edit the following fields:
 - **Name:** Enter a name for the dashboard.
 - **Published:** Select this option to make the dashboard available for use.
 - **Description:** Enter a description for the dashboard.
 - **Protected by Privilege:** Select an available privilege from the drop-down list to enable only those with this privilege to securely access the dashboard.
 - **Go to Dashboard details after creation:** Select this option to go to the main Charts and Dashboards page after creating the dashboard.
2. Click **Create** or click **Save**, if editing. The Layout page is displayed
3. In the Layout page, you can add charts and design the layout for the dashboard. Click **Add Chart**.
 - In the **Select Chart** pane, for the **Selection** tab, select the chart to add from the drop-down list.
You can select the **Custom** option and enter a URL for the chart from your schema or from a different schema. The URL must point to the REST Module root, for example, `http://xxxx.oracle.com:1234/ords/hr/sdw/charts/Chart01/`. Also, you need to have the associated privileges to view the chart.

A preview of the selected chart is displayed. You can change the refresh rate of the chart, display the chart in Chart or tabular format, or share the chart by copying the URL.
 - In the **Settings** tab, select **Chart** and **Data Grid** to enable both views for the chart. By default, the Chart view is displayed in Preview.
4. Click **Create**.
The chart is added to the Layout section.



5. Add more charts as needed to the dashboard. The maximum is 3 or 4 based on the number of columns that you select from the Columns drop-down list.
 - You can reorder the charts by dragging them horizontally or vertically to move them to the required position.

- Use the Edit icon on a chart to edit the chart if needed. You can also change the chart by selecting from the drop-down list.
 - Drag the double-sided arrow at the right corner to resize the chart.
6. Click **Save and Preview** to save the layout and display a preview of the dashboard.

The dashboard is displayed on a new page. For each chart, you can change or customize the refresh rate, display the chart in Chart or tabular format, or share the chart by copying the URL.

7

The Scheduling Pages

Note

Available for Oracle Database 12c release 2 and later releases.

Database Actions provides a graphical interface for using the [DBMS_SCHEDULER](#) PL/SQL package to work with Oracle Scheduler objects such as Jobs, Chains, Programs and Schedules. To use the scheduling features, you must first understand the concepts and essential tasks for job scheduling, which are explained in the chapters about Oracle Scheduler concepts and scheduling jobs in the *Oracle Database Administrator's Guide*.

The Scheduling functionality is focused on exploring Oracle Database Scheduler artifacts and provides details for jobs, chains, chain steps and rules, job execution history, programs, schedules, job classes, file watchers, windows and window groups. Depending on the access rights of the logged-in user, information is taken from `ALL_*` or `DBA_*` views.

Note

SQL and PL/SQL editors are used in the user interfaces for Jobs and Programs. Some shortcut keys used in these editors are:

- **Alt+F1**: Invokes accessibility options.
- **Ctrl+M**: Alters the Tab key behavior, from inserting tab to going to the next control on the form.

For more keyboard shortcuts, see [Keyboard Shortcuts](#).

To navigate to the Scheduling pages, do either of the following:

- In the Launchpad page, select the **Development** tab and click **Scheduling**.
- Click **Selector**  to display the navigation menu. Under Development, select **Scheduling**.

Topics

- [Overview](#)
- [Jobs](#)
- [Chains](#)
- [Programs](#)
- [Schedules](#)
- [Objects](#)

- [Overview](#)
The Overview page provides an overview of the activity in the Scheduling pages. It consists of two parts: Objects and Problem Jobs
- [Jobs](#)
- [Chains](#)
- [Programs](#)
- [Schedules](#)
- [Objects](#)

Overview

The Overview page provides an overview of the activity in the Scheduling pages. It consists of two parts: Objects and Problem Jobs

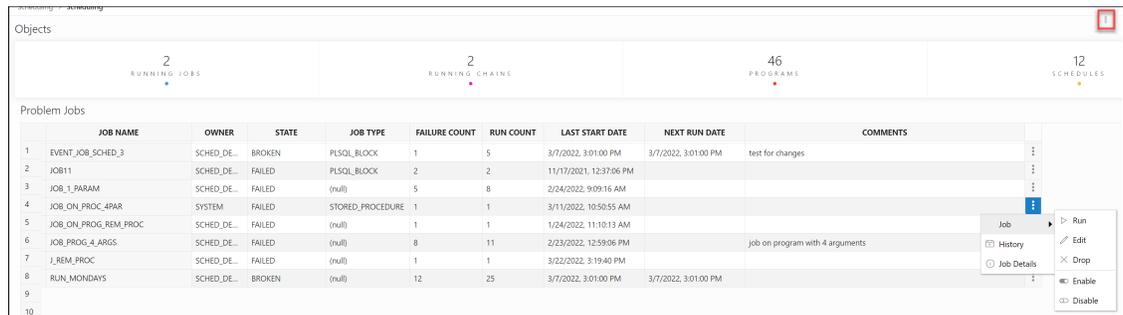
The Objects section provides an overview of the total number of running jobs, running chains, programs and schedules. Click an object tile to go to the corresponding page.

Click  **Scheduler Summary** at the top right to view details such as Running Jobs, Current open window, Log history retain, Default timezone, Event expiry time, File watcher count, repeat interval, and mail server settings.

This Problem Jobs section provides information about jobs with problems in their execution. Listed in tabular format are jobs having one of the following statuses: Failed, Broken, Chain_Stalled, Blocked, Retry Scheduled, Resource_Unavailable. The  **Actions** context menu provides links to the history and job details for each problem job. Other actions that you can perform are run, edit, drop, enable and disable job.

 **Note**

If there are no jobs with problems, the Jobs Summary page appears as the initial page for the Scheduling functionality.



The screenshot shows the Oracle Scheduling Overview page. At the top, there are four summary tiles: '2 RUNNING JOBS', '2 RUNNING CHAINS', '46 PROGRAMS', and '12 SCHEDULES'. Below these is a 'Problem Jobs' section with a table listing jobs that have encountered issues. The table has columns for Job Name, Owner, State, Job Type, Failure Count, Run Count, Last Start Date, Next Run Date, and Comments. A context menu is open over the first job, showing options: Run, Edit, Drop, Job Details, Enable, and Disable.

	JOB NAME	OWNER	STATE	JOB TYPE	FAILURE COUNT	RUN COUNT	LAST START DATE	NEXT RUN DATE	COMMENTS
1	EVENT_JOB_SCHED_3	SCHED_DE...	BROKEN	PLSQL_BLOCK	1	5	3/7/2022, 3:01:00 PM	3/7/2022, 3:01:00 PM	test for changes
2	JOB11	SCHED_DE...	FAILED	PLSQL_BLOCK	2	2	11/17/2021, 12:37:06 PM		
3	JOB_1_PARAM	SCHED_DE...	FAILED	(null)	5	8	2/24/2022, 9:09:16 AM		
4	JOB_ON_PROG_4PAR	SYSTEM	FAILED	STORED_PROCEDURE	1	1	3/11/2022, 10:50:55 AM		
5	JOB_ON_PROG_REM_PROC	SCHED_DE...	FAILED	(null)	1	1	1/24/2022, 11:10:13 AM		
6	JOB_PROG_4ARGS	SCHED_DE...	FAILED	(null)	8	11	2/23/2022, 12:59:06 PM		job on program with 4 arguments
7	J_REM_PROC	SCHED_DE...	FAILED	(null)	1	1	3/22/2022, 3:19:40 PM		
8	RUN_MONDAYS	SCHED_DE...	BROKEN	(null)	12	25	3/7/2022, 3:01:00 PM	3/7/2022, 3:01:00 PM	
9									
10									

Jobs

There are five options available in the Jobs menu: Summary, Running, Forecast, History, Notifications. Each option is described below.

- [Summary](#)
- [Running](#)
- [History](#)
- [Forecast](#)
- [History \(Gantt Chart\)](#)
- [Notifications](#)
- [Create or Edit Job](#)

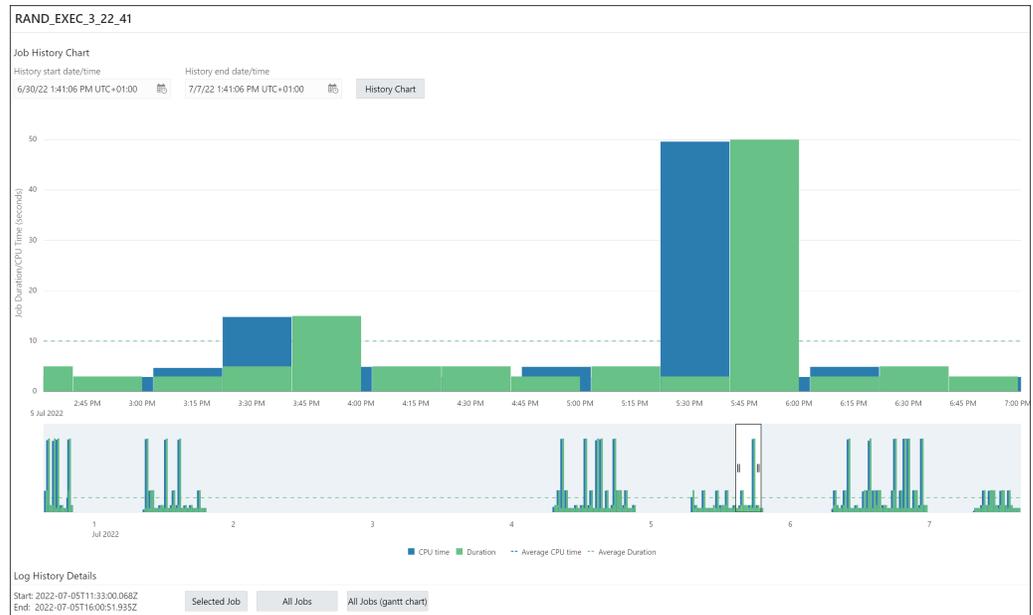
Summary

At the top, the Summary page provides an overview of the total number of Failed, Blocked, Chain stalled, Resource unavailable, Broken, and Retry Scheduled jobs. Click a job tile and the corresponding list of jobs are displayed in tabular format below. Remove the filters to display all scheduled jobs.

To create a job, see [Create or Edit Job](#).

The Actions icon  is available at the end of each job row. Click **Actions** to view the following list of options:

- **Job**
 - **Run:** Runs the specific job.
 - **Edit:** See [Create or Edit Job](#).
 - **Drop:** Drops the specific job.
 - **Job-Enable/Disable:** If enabled is selected, the job is picked up by the Scheduler for processing. The status of the job (enabled or not) is seen in the Job Details page, where the property “enabled” displays TRUE or FALSE.
- **History**
 - **Report:** Provides a history of the job runs in a report format, including log details, status of the run, run duration, errors if any, and so on.
 - **Chart:** Provides a history of the job runs in a visual bar chart representation. Run duration and used CPU time are presented for each execution. Jobs in chains are presented with aggregated information for the whole job. There is an overview scrollbar that enables zoom and scroll by changing the time frame. You can display all details from logs for that time frame for a selected job or for all jobs. Also, a Gantt chart can be shown for all jobs for a selected time frame.



- Job Forecast:** You can execute the job forecast functionality for a single job (available in the Action menu) or for a set of jobs (available on the toolbar for the table). For a set of jobs, filtering can be used to narrow the set of jobs. Use **Ctrl key+click** to select some jobs and then Job Forecast will run for the selected jobs only. If there are no selected jobs, then forecast will run for all listed jobs. Not every job included in the set is included in the final forecast. Only jobs with a defined calendar (repeat interval) are included. You can define the calendar inline or using schedule, window or window group. After the forecast is done, you can select different zoom levels and filter the results by schema.
- Job Details:** Displays job attributes, such as action, job class, type, schedule, and so on. Select **JSON** to view the attributes in JSON format. Depending on the job's definition, details about used objects are provided such as used program, PL/SQL code, procedure, procedure dependencies, program arguments, job arguments, job class, schedule, window, window group together with windows in the group, file watcher, history chart presenting one week of history from last start date.

Running

The Running page displays the list of currently running jobs. The following commands are available: Stop, Edit, History, Job Details.

JOB NAME	OWNER	JOB STYLE	DETACHED	ELAPSED TIME	CPU USED	DESTINATION OWNER	DESTINATION	CREDENTIAL OWNER	CREDENTIAL NAME	LOG ID
1 A_JOB_ON_CH_N222	SCHED_DEMO_USER	REGULAR	FALSE	P270746M55445	null	null	null	null	null	4294967295
2 CHAIN_JOB_1	SCHED_DEMO_USER	REGULAR	FALSE	P8607Z1H59M323295	null	null	null	null	null	4294967291

History

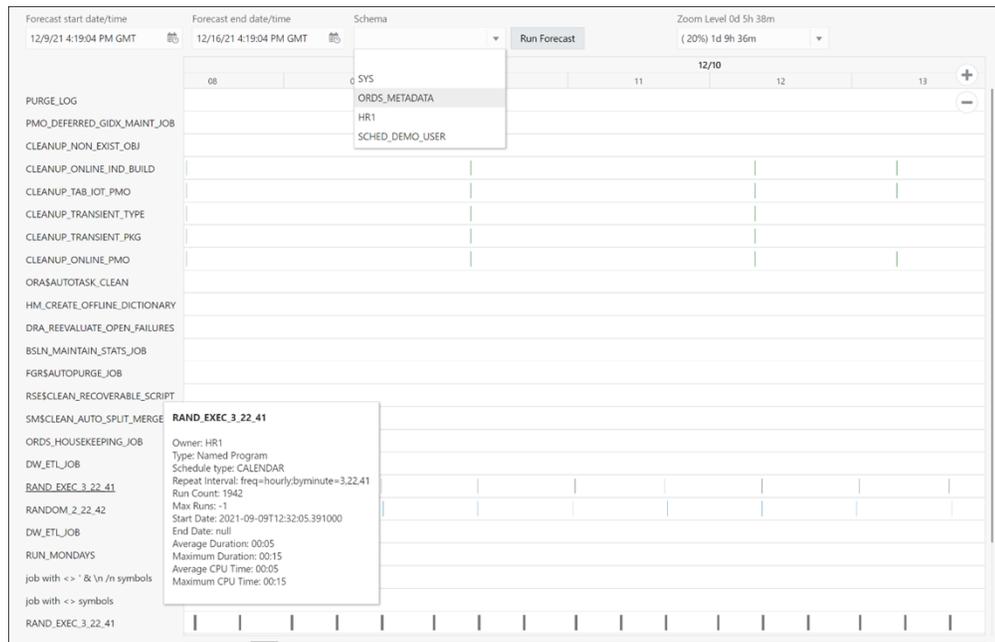
The History page displays log run details for all Scheduler jobs available to the user. You can use the History window and set filters to limit the amount of data. You can filter using delayed

jobs by providing a delay interval. Ordering is supported on the grid by clicking the column header.

Forecast

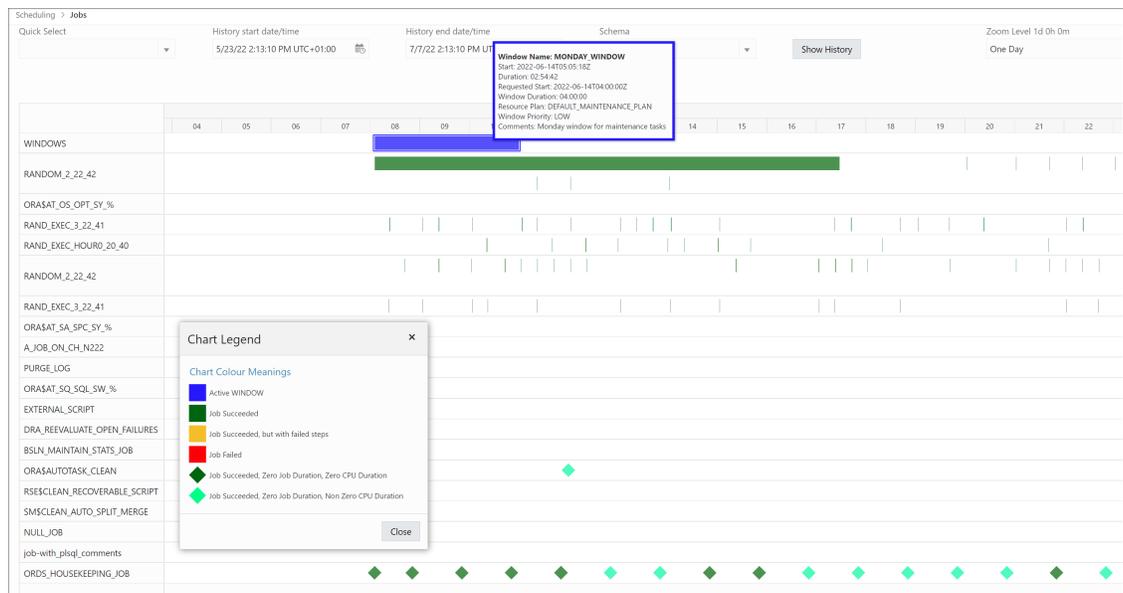
The Forecast page provides the job execution forecast for all available jobs. This operation takes time to finish depending on the number of jobs and forecast interval. Forecast for all available jobs depends on the rights of the connecting user. After execution, you can filter the results based on the schema of the jobs.

The Forecast functionality is also available on the Jobs - Summary page. In that case, it works on the list of available jobs. You can filter the jobs before the forecast functionality is used. If there is a selection of jobs, then the functionality uses only the selected jobs.



History (Gantt Chart)

History represents windows and job execution history in the form of a Gantt chart. The windows activation history is shown on the first row. Jobs are ordered in a descending order based on the maximum used CPU time. Details are shown for each window activation and job execution. The job summary is available when hovering over the label of a job's row.



Notifications

You can create notifications only if an email server is set for the scheduler.

The Notifications page enables you to view, create, edit and delete notifications related to job events. Also, you can see email server details by clicking the **Notifications Email Server** icon at the top right of the page.

For each message, you can specify job, recipient email addresses and sender (or no sender), and you can modify the subject and body of the message and set the filter condition. If multiple recipients and events are provided, then Oracle Scheduler creates a separate notification record for each combination <recipient,event>. You can edit the content of a set of notifications created in this way by using **Edit Aggregated**, which appears in the context menu of each notification. **Edit** edits the content of a single notification.

Notifications can be filtered by job name, job owners, recipients and events. You can select and remove some notifications by clicking the **Remove Notifications** icon. If there are no selected notifications (use **Ctrl+click** to deselect a single notification), then the Remove Notifications dialog appears enabling removal of all notifications for the selected job, or job and recipients, or job and events, or job and events and recipients.

Remove Notifications

<div style="border-bottom: 1px solid gray; padding: 5px;">Properties</div> <div style="padding: 5px;">SQL</div>	<div style="margin-bottom: 10px;"> Schema <input style="width: 100%;" type="text" value="SCHEM_DEMO_USER"/> </div> <div style="margin-bottom: 10px;"> Job Name * <input style="width: 100%;" type="text" value="DW_ETL_JOB"/> </div> <div style="margin-bottom: 10px;"> Select Recipients <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 5%;"></th> <th style="width: 90%;">Recipient</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;"><input type="checkbox"/></td> <td>joel@abc.com</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;"><input type="checkbox"/></td> <td>smith@abc.com</td> </tr> </tbody> </table> </div> <div> Select Events <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 5%;"></th> <th style="width: 90%;">Event</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;"><input type="checkbox"/></td> <td>JOB_DISABLED</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;"><input type="checkbox"/></td> <td>JOB_FAILED</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;"><input type="checkbox"/></td> <td>JOB_SCH_LIM_REACHED</td> </tr> </tbody> </table> </div>			Recipient	1	<input type="checkbox"/>	joel@abc.com	2	<input type="checkbox"/>	smith@abc.com			Event	1	<input type="checkbox"/>	JOB_DISABLED	2	<input type="checkbox"/>	JOB_FAILED	3	<input type="checkbox"/>	JOB_SCH_LIM_REACHED
		Recipient																				
1	<input type="checkbox"/>	joel@abc.com																				
2	<input type="checkbox"/>	smith@abc.com																				
		Event																				
1	<input type="checkbox"/>	JOB_DISABLED																				
2	<input type="checkbox"/>	JOB_FAILED																				
3	<input type="checkbox"/>	JOB_SCH_LIM_REACHED																				

?
Remove
Cancel

Create or Edit Job

This section describes how to create a new Oracle Scheduler job or edit an existing job.

To create a job, Database Actions internally uses the `DBMS_SCHEDULER.CREATE_JOB` procedure, which is documented in *Oracle Database PL/SQL Packages and Types Reference*.

1. In the Jobs page, at the top right, click **Create Job**.
2. In Job Properties, enter the following fields:

Details

- **Enabled:** If this option is specified, validity checks are made and the job is created enabled if all the checks are successful. If this option is not specified, the job is not created enabled.
- **Name:** Name of the job.
- **Description:** Optional text string that can be used to describe the job.
- **Type:** Type of object to be executed by the job: PL/SQL Block, Chain, Stored Procedure, Named Program, or Script. Additional controls appear for Chain, Stored Procedure and Named Program enabling you to select related objects.

For Stored Procedure, only procedures with IN parameters are listed, procedures with IN OUT or OUT parameters are not permitted.

Schema level and package level procedures are listed for the selected schema but you can directly type the procedure name (or `package_name.procedure_name`) in the field.

- **Class:** Name of the job class to which this job belongs.

Execution Mode

- **Mode:** When to execute the job: **Immediate** (immediately on creation, and once only), **Once** (once, at a specified time), **Repeating**, **Queue**, **File Watcher**, **Schedule** (using a named schedule object), **Window** and **Window group**. If you specify anything other than Immediate, you are prompted for additional information.
For Repeating, you can manually define the repeat interval or click the pencil icon to select the date, frequency, weekday, and interval values. Some clauses of Oracle calendar syntax (include, exclude, intersect, periods and by period) are not supported and a warning is displayed when the edit icon is clicked.

Destination

- **Local** (local system), **Remote** (the database destination for a remote database job, or external destination for a remote external job), or **Multiple** (the job runs on all destinations associated with the provided destination group). Depending on what destination you selected for the job, select the local credential, the remote credential and destination, or the destination group.

Properties

- **Auto Drop:** Determines whether the job is to be automatically dropped after it has completed or has been automatically disabled.
- **Restart on failure:** Determines whether the job can be restarted in case of failure.
- **Restart on recovery:** Determines whether to restart the job in case of database failure.
- **Store Output:** If enabled, then for job runs that are logged, all job output and error messages are stored in the *_JOB_RUN_DETAILS views. If disabled, then the output and messages are not stored.
- **Follow Default Time Zone:** Determines whether if the job start date is null, then when the default time zone scheduler attribute is changed, the Scheduler recomputes the next run date and time for this job so that it is in accordance with the new time zone.
- **Allow Runs in Restricted Mode:** If enabled, the job is permitted to run when the database is in restricted mode, provided that the job owner is permitted to log in during this mode.
- **Stop on Window Close:** If the schedule of a job is a window or a window group, enabling this option causes the job to stop once the associated window is closed, and disabling causes the job to continue after the window closes. (Note that if the job is allowed to continue, its resource allocation will probably change because closing a window generally also implies a change in resource plans.)
- **Instance Stickiness:** This attribute should only be used for a database running in an Oracle Real Application Clusters (Oracle RAC) environment. By default, it is enabled. Jobs start running on the instance with the lightest load and the Scheduler thereafter attempts to run on the instance that it last ran on. If that instance is either down or so overloaded that it does not start new jobs for a significant period of time, another instance runs the job. If the interval between runs is large, instance_stickiness is ignored and the job is handled as if it were a non-sticky job. If instance_stickiness is disabled, each instance of the job runs on the first instance available.
- **Parallel Instances:** For an event-based job, determines what happens if an event is raised and the event-based job that processes that event is already running. If disabled, it causes the new event to be ignored. If enabled, it causes an instance of the job to be started for every instance of the event, and each job instance is a lightweight job so multiple instances of the same event-based job can run in parallel.

- **Job Style:** Style of the job being created: **REGULAR** (regular job) or **LIGHTWEIGHT** (lightweight job). A lightweight must reference a program object. Use lightweight jobs when you have many short-duration jobs that run frequently. Under certain circumstances, using lightweight jobs can deliver a small performance gain.
- **Job Priority:** The priority of this job relative to other jobs in the same class as this job. If multiple jobs within a class are scheduled to be executed at the same time, the job priority determines the order in which jobs from that class are picked up for execution by the job coordinator. It can be a value from 1 through 5, with 1 being the first to be picked up for job execution.
- **Logging Level:** Determines how much information is logged:
DBMS_SCHEDULER.LOGGING_OFF (no logging),
DBMS_SCHEDULER.LOGGING_FAILED_RUNS (only jobs that failed, with the reason for failure), DBMS_SCHEDULER.LOGGING_RUNS (all runs of each job in this class),
or DBMS_SCHEDULER.LOGGING_FULL (all operations performed on all jobs).

However, if the job class has a higher (more detailed) logging level than the level specified for the job, the job class logging level is used.
- **Instance ID:** In an Oracle Real Application Clusters environment., the instance ID of the instance that the job must run on.
- **Max Runs:** The maximum number of consecutive scheduled runs of the job.
- **Max Failures:** The number of times a job can fail on consecutive scheduled runs before it is automatically disabled.
- **Raise Events:** Determines at what stages of the job execution to raise events.
- **Max Run Duration:** Maximum amount of time that the job should be allowed to run. Its data type is INTERVAL DAY TO SECOND. If this attribute is set to a nonzero and non-null value, and job duration exceeds this value, the Scheduler raises an event of type JOB_OVER_MAX_DUR. It is then up to your event handler to decide whether or not to allow the job to continue.
- **Schedule Limit:** Maximum delay time between scheduled and actual job start before a program run is canceled.
- **Reset to Defaults:** Resets all properties to their default values.

NLS

Enables you to set the NLS-related property required for this job. To enter a value, type in to the related field for the Value column.

For a new job, parameters are taken from the database session.

3. In the **DDL** pane, you can review and save the SQL statements that are generated when creating or editing the job.
 - For a new job, click **CREATE** to view the generated DDL statements.
 - When you edit a job, click **UPDATE** to view the generated ALTER statements.

When you are finished, click **Apply**.

The **Output** pane displays the results of the DDL commands. If there are any errors, go to the corresponding pane, fix the errors, and run the commands again. You can save the content to a file.

Chains

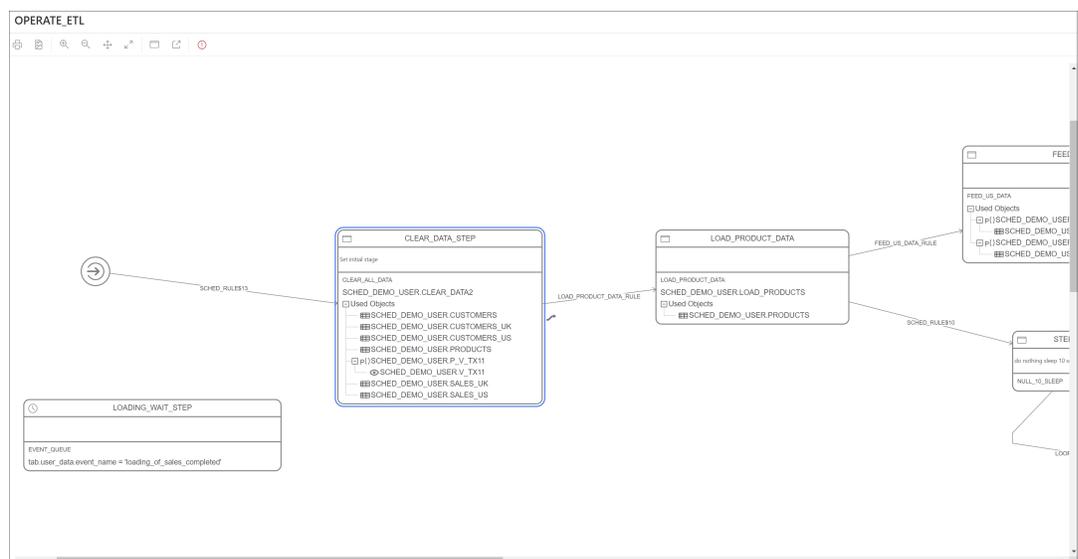
There are four options available in the Chains menu: Summary, Running Chains, Steps, Rules. Each option is described below.

Summary

The Summary page displays all the chains owned by the current user. The details are presented in a tabular format. At the end of each row is the Actions icon . Click **Actions** to view the list of options for the chain. Click a chain in the table and details about steps and rules appear below the Chains table. There is a **Drop Step** or **Drop Rule** option available in the context menu for a step or rule.

The actions available for a chain are:

- **Run:** Runs the specific chain.
- **Add Step:** Creates a new step for the selected chain.
- **Add Rule:** Creates a new rule for the selected chain.
- **Edit:** Edits the chain properties.
- **Drop:** Drops the chain
- **Chain-Enable/Disable:** Enables or disables the chain.
- **Show Diagram:** Displays the job steps of a chain in a visual diagram format.



Double-click a step or rule in the diagram to see more details. You can directly create steps and rules on the diagram using the **Add Step** and **Add Rule** icons.

The following dependencies are shown on each step:

- Event schedule: The name of the schedule.
- Event queue: The name of the queue and condition.
- Program using procedure or PL/SQL block: Tables, views, procedures functions and packages used - two levels of dependencies.

You can also create a rule by connecting two steps:

- Click on a step and the link icon appears.

Running Chains

The Running Chains page displays the execution of the steps of the currently running chains.

The following commands are available for each row in the report:

- For Chain, the commands are Edit, Drop, Enable, Disable, and Evaluate Chain.

Evaluate Chain invokes the `DBMS_SCHEDULER.EVALUATE_RUNNING_CHAIN` procedure. It forces the reevaluation of the rules of a running chain to trigger any rules for which the conditions have been satisfied.

- Other chain-related command are Show Diagram, Analyze Chain, Step Details, and Edit Step.

Steps

The Steps page provides details of each step in all chains.

Rules

The Rules page provides details of each rule in all chains.

- [Create or Edit Chain](#)
- [Add or Edit Step](#)
- [Add or Edit Rule](#)

Create or Edit Chain

This section describes how to create or edit a chain.

To create a chain, Database Actions internally uses the `DBMS_SCHEDULER.CREATE_CHAIN` procedure.

1. In the Chains page, at the top right, click **Create Chain**.
2. In Properties, enter the following fields:
 - **Name:** Name to assign to the chain.
 - **Evaluation Interval:** **NULL** reevaluates of the rules of a running chain only when the job starts and when a step completes. A non-NULL value causes rule evaluations to also occur periodically at the specified interval.
 - **Enabled:** Enables the chain. (Causes the `DBMS_SCHEDULER.ENABLE` procedure to be called after the chain is created.)
3. Click **Create** to create the chain.

Click **Create and Open** to create the chain and open in Diagram mode.

Add or Edit Step

This section describes how to add a step for a chain.

1. In the context menu for a chain, select **Add Step**.
2. In the Create Step slider, enter the following fields:
 - **Step Name:** Name of the step.

- **Type:** Select **Program**, **Chain**, **Event** or **Queue** to run during the step. Additionally, select the name of the program, chain, event or queue in the corresponding field.
 - **Pause:** Select this option to pause the step.
 - **Skip:** Select this option to skip the step.
 - **Restart on Failure:** Select this option to restart the step after database failure.
 - **Restart on Recovery:** Select this option to restart the step after database recovery.
3. Click **Create**.

The new step is displayed for the chain in the main Chains page.

Add or Edit Rule

This section describes how to add a rule for a chain.

1. In the context menu for a chain, select **Add Rule**.
2. In the Create Rule slider, enter the following fields:
 - **Rule Name:** Name of the rule being created.
 - **Condition:** Enter the condition to evaluate.
 - **Action:** Action to perform if condition is met.
 - **Comments:** Optional comments describing the rule.

See Also

[DEFINE_CHAIN_RULE Procedure](#) in *Oracle Database PL/SQL Packages and Types Reference*

Programs

The Programs page displays the list of created programs.

The Actions menu for a program consists of the following options: Edit, Drop, Enable or Disable the program and View Program Details, including used procedure and procedure dependencies.

The screenshot shows the Oracle Cloud console interface. On the left, there is a table of programs with columns: Detached, Schedule Limit, Priority, Weight, Max Runs, Max Failures, and Max Run Durations. The table contains 19 rows, all with 'FALSE' in the Detached column and '3' in the Priority column. On the right, the 'RANDOM_EXEC_TIME' program properties are displayed, showing the PL/SQL code for the procedure.

Program ID	Detached	Schedule Limit	Priority	Weight	Max Runs	Max Failures	Max Run Durations
1	FALSE	(null)	3	1	(null)	(null)	(null)
2	FALSE	(null)	3	1	(null)	(null)	(null)
3	FALSE	(null)	3	1	(null)	(null)	(null)
4	FALSE	(null)	3	1	(null)	(null)	(null)
5	FALSE	(null)	3	1	(null)	(null)	(null)
6	FALSE	(null)	3	1	(null)	(null)	(null)
7	FALSE	(null)	3	1	(null)	(null)	(null)
8	FALSE	(null)	3	1	(null)	(null)	(null)
9	FALSE	(null)	3	1	(null)	(null)	(null)
10	FALSE	(null)	3	1	(null)	(null)	(null)
11	FALSE	(null)	3	1	(null)	(null)	(null)
12	FALSE	(null)	3	1	(null)	(null)	(null)
13	FALSE	(null)	3	1	(null)	(null)	(null)
14	FALSE	(null)	3	1	(null)	(null)	(null)
15	FALSE	(null)	3	1	(null)	(null)	(null)
16	FALSE	(null)	3	1	(null)	(null)	(null)
17	FALSE	(null)	3	1	(null)	(null)	(null)
18	FALSE	(null)	3	1	(null)	(null)	(null)
19	FALSE	(null)	3	1	(null)	(null)	(null)

```

PROCEDURE RANDOM_EXEC AS
  CURR_TIMES TIMESTAMP WITH TIME ZONE := SYSTIMESTAMP;
  CURR_TIMES NUMBER := DBMS_UTILITY.GET_TIME;
  END_TIME TIMESTAMP;
  GEN NUMBER(2);
BEGIN
  END_TIME := CURR_TIMES + INTERVAL '3' SECOND;
  GEN := DBMS_RANDOM.VALUE(1, 10);
  IF GEN IN (1, 2, 3, 7) THEN
    END_TIME := CURR_TIMES + INTERVAL '15' SECOND;
  ELSIF GEN IN (4, 5, 6, 8) THEN
    END_TIME := CURR_TIMES + INTERVAL '5' SECOND;
  END IF;
  DBMS_OUTPUT.PUT_LINE(GEN);
  LOOP
    EXIT WHEN CURRENT_TIMESTAMP >= END_TIME;
  END LOOP;
END RANDOM_EXEC;

```

- [Create or Edit Program](#)
This section describes how to create or edit an Oracle Scheduler program.

Create or Edit Program

This section describes how to create or edit an Oracle Scheduler program.

To create a program, Database Actions internally uses the `DBMS_SCHEDULER.CREATE_PROGRAM` procedure.

1. In the Programs page, at the top right, click **Create Program**.
2. In Program Properties, enter the following fields:

Details tab

- **Name:** Name of the program. The name has to be unique in the SQL namespace. For example, a program cannot have the same name as a table in a schema.
- **Enabled:** If this option is specified, validity checks will be made and the program will be created enabled if all the checks are successful. If this option is not specified, the program is not created enabled.
- **Description:** Optional text string that can be used to describe the program.
- **Type**
 - **PL/SQL Block:** The program is a PL/SQL block. Job or program arguments are not supported when the job or program type is `PLSQL_BLOCK`. In this case, the number of arguments must be 0. Enter or paste in the complete PL/SQL code, or edit the existing code.
 - **Stored Procedure:** The program is a PL/SQL or Java stored procedure, or an external C subprogram. Only procedures, not functions with return values, are supported. PL/SQL procedures with IN OUT or OUT arguments are not supported.

Schema: Schema of the stored procedure. If not specified, the schema of the job is assumed.

Procedure: Name of the stored procedure.

Arguments: For each argument, name, data type, default value, and whether it is an input, output, or input/output argument.

- **Script:** The program is a SQL Script (SQL*Plus statements), Backup Script (RMAN commands), or External Script (operating system commands). Enter or paste the script text in the box.

Properties tab

Enables you to set program properties. For most properties the default is null, but you can check the box to specify a value.

- **Detached:** Enabled if the program is a detached job. Use a detached job to start a script or application that runs in a separate process, independently and asynchronously to the Scheduler. A detached job typically starts another process and then exits. Upon exit (when the job action is completed) a detached job remains in the running state. The running state indicates that the asynchronous process that the job started is still active. When the asynchronous process finishes its work, it must connect to the database and call DBMS_SCHEDULER.END_DETACHED_JOB_RUN, which ends the job.
 - **Max Runs:** Maximum number of runs before the program is marked as completed.
 - **Max Failures:** Maximum number of failures tolerated before the program is marked as broken.
 - **Max Run Duration:** Maximum run duration of the program.
 - **Schedule Limit:** Maximum delay time between scheduled and actual job start before a program run is canceled.
3. In the **DDL** pane, you can review and save the SQL statements that are generated.
 - For a new program, click **CREATE** to view the generated DDL statements.
 - When you edit a program, click **UPDATE** to view the generated ALTER statements.

When you are finished, click **Apply**.
 4. The **Output** pane displays the results of the DDL commands. If there are any errors, go to the corresponding pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

Schedules

The Schedules page displays the list of available schedules.

The Actions menu for a schedule has the following options: Viewing schedule details, Editing or Dropping the schedule.

The screenshot shows the Oracle Cloud Schedules page. On the left, there is a table listing schedules. On the right, the details for the schedule 'BSLN_MAINTAIN_STATS_SCHED' are displayed in a JSON format.

End Date	Repeat Interval	Event Queue Owner	Event Queue Name
(null)	FREQ=WEEKLY	(null)	(null)
(null)	freq=daily,byhours=3,byminute=0,bysecond	(null)	(null)
(null)	FREQ=MINUTELY,INTERVAL=10	(null)	(null)
(null)	FREQ=DAILY, BYHOUR=02, BYMINUTE=0	(null)	(null)

```

{
  "owner": "SYS",
  "schedule_name": "BSLN_MAINTAIN_STATS_SCHED",
  "schedule_type": "CALENDAR",
  "start_date": "2019-04-23T00:00:00.000000",
  "repeat_interval": "FREQ=WEEKLY",
  "event_queue_owner": null,
  "event_queue_name": null,
  "event_condition": null,
  "file_watcher_owner": null,
  "file_watcher_name": null,
  "end_date": null,
  "comments": "Pre-defined schedule for computing moving window baseline statistics"
}

```

- [Create or Edit Schedule](#)

Create or Edit Schedule

This section describes how to create or edit Oracle Scheduler schedule.

To create a schedule, Database Actions internally uses the `DBMS_SCHEDULER.CREATE_SCHEDULE` procedure.

1. In the Schedules page, at the top right, click **Create Schedule**.
2. In Program Properties, enter the following fields:

Properties tab

- **Name:** Name of the schedule. The name has to be unique in the SQL namespace. For example, a schedule cannot have the same name as a table in a schema.
 - **Description:** Optional text string that can be used to describe the schedule.
 - **Mode:** Specify when jobs that use this schedule are to run: **Repeating** (specify the repeat interval, start date, and end date), **Queue** (specify the queue name, agent, condition, start date, and end date), or **File Watcher** (specify the file watcher object name, condition, start date, and end date).
3. In the **DDL** pane, you can review and save the SQL statements that are generated.
 - For a new schedule, click **CREATE** to view the generated DDL statements.
 - When you edit a schedule, click **UPDATE** to view the generated ALTER statements.

When you are finished, click **Apply**.

4. The **Output** pane displays the results of the DDL commands. If there are any errors, go to the corresponding pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

Objects

The Objects menu in Scheduling provides links to the following four pages: Job Classes, File Watchers, Windows and Window Groups

Job Classes

A job class is an Oracle Scheduler object that enables the Scheduler administrator to group jobs for logical purposes, such as to assign the same set of attribute values to member jobs, to set service affinity for member jobs, to set resource allocation for member jobs, or to group jobs for prioritization.

Note

Creating a job class requires the `MANAGE_SCHEDULER` system privilege. Access to `SYS.DBA_RSRC_CONSUMER_GROUPS` and `SYS.DBA_SERVICES` dictionary views is required in order UI to provide all details and choices.

The Job Classes page displays the job classes created in a tabular format. You can perform the following actions:

- **Create:** To create a job class, see [Create Job Class](#).

- **Edit:** To edit a job class, select **Edit** from the Actions menu at the end of a row. For a description of the fields, see [Create Job Class](#).
- **Drop:** To drop a job class, select **Drop** from the Actions menu.

File Watchers

A file watcher is an Oracle Scheduler object that defines the location, name, and other properties of a file whose arrival on a system causes the Scheduler to start a job. You create a file watcher and then create any number of event-based jobs or event schedules that reference the file watcher. When the file watcher detects the arrival of the designated file, it raises a file arrival event. The job started by the file arrival event can retrieve the event message to learn about the newly arrived file.

The File Watchers page displays the file watchers created in a tabular format. You can perform the following actions:

- **Create:** To create a file watcher, see [Create File Watcher](#).
- **Edit:** To edit a file watcher, select **Edit** from the Actions menu at the end of a row. For a description of the fields, see [Create File Watcher](#).
- **Drop:** To drop a file watcher, select **Drop** from the Actions menu.
- **Show DDL:** Displays the Data Definition Language statements for the file watcher. To show DDL, select this option from the Actions menu.

Windows

A window is an Oracle Scheduler object that can be used to automatically start jobs or to change resource allocation among jobs during various time periods of the day, week, and so on.

Note

Creating a window object requires the `MANAGE SCHEDULER` system privilege. Access to `SYS.DBA_RSRC_PLANS` dictionary view is required to provide all details and choices.

The Windows page displays windows created in a tabular format. You can perform the following actions:

- **Create:** To create a window, see [Create Window](#).
- **Edit:** To edit a window, select **Edit** from the Actions menu at the end of a row. For a description of the fields, see [Create Window](#).
- **Drop:** To drop a window, select **Drop** from the Actions menu.
- **Show DDL:** Displays the Data Definition Language statements for the window. To show DDL, select this option from the Actions menu.

Window Groups

A window group is an Oracle Scheduler object that is a list of Oracle Scheduler Windows. Scheduler jobs that are scheduled to be run in a window group will be activated in that time span and using that resource plan for all windows in the group

The Window Groups page displays windows created in a tabular format. You can perform the following actions:

- **Create:** To create a window group, see [Create Window Group](#).
- **Edit:** To edit a window group, select **Edit** from the Actions menu at the end of a row. For a description of the fields, see [Create Window Group](#).

- **Drop:** To drop a window group, select **Drop** from the Actions menu.
- **Show DDL:** Displays the Data Definition Language statements for the window group. To show DDL, select this option from the Actions menu.
- [Create Job Class](#)
- [Create File Watcher](#)
- [Create Window](#)
- [Create Window Group](#)

Create Job Class

To create a job class:

1. In the Job Class page, click **Create Job Class**.
2. Enter the following fields:
 - **Name:** Name of the job class.
 - **Description:** Optional text string that can be used to describe the job class.
 - **Logging Level:** Specifies how much information is written to the job log:
 - **RUNS:** Detailed information for all runs of each job in this class.
 - **FULL:** RUNS plus all operations performed on all jobs in this class.
 - **OFF:** No logging.
 - **Log Retention Period (days):** Number of days that job log entries for jobs in this class are retained. The range of valid values is 0 through 999. If set to 0, no history is kept. If NULL (the default), retention days are set by the `log_history` Scheduler attribute.
 - **Service Name:** The database service that the jobs in this class will have affinity to. If no service is specified, the job class will belong to the default service, which means it will have no service affinity and any one of the database instances within the cluster might run the job.
 - **Resource Consumer Group:** Resource consumer group this class is associated with. If no resource consumer group is specified, the job class is associated with the default resource consumer group.
3. Click **Create**.

Create File Watcher

To create a file watcher:

1. In the File Watchers page, click **Create File Watcher**.
2. Enter the following fields:
 - **Enabled:** Enables the file watcher. (Causes the `DBMS_SCHEDULER.ENABLE` procedure to be called after the file watcher is created.)
 - **Name:** Name of the file watcher.
 - **Description:** Optional descriptive text.
 - **Destination:** Name of an external destination. You create an external destination by registering a remote Scheduler agent with the database. The view

ALL_SCHEDULER_EXTERNAL_DESTS lists valid external destination names. If this parameter is null, the file watcher is created on the local host.

- **Credential Name:** Name of a valid credential object. The file watcher uses the credential to authenticate itself with the host operating system to access the watched-for file. The file watcher owner must have EXECUTE privileges on the credential.
- **Directory Path:** Directory in which the file is expected to arrive. The single wildcard '?' at the beginning of the path denotes the Oracle home path. For example, '?/rdbms/log' denotes the rdbms/log subdirectory of the Oracle home directory.
- **File Name:** Name of the file to look for. Two wildcards are permitted anywhere in the file name: '?' denotes any single character, and '*' denotes zero or more characters.
- **Min File Size:** Minimum size in bytes that the file must be before the file watcher considers the file found.
- **Steady State Duration:** Minimum time interval that the file must remain unchanged before the file watcher considers the file found. Cannot exceed one hour. If null, an internal value is used. The minimum value is 10 seconds. Oracle recommends similar steady state duration values for all file watchers for efficient file watcher job operation. Also, the repeat interval of the file watcher schedule must be equal or greater than the steady state duration value.

3. Click **Create**.

Create Window

To create a window:

1. In the Windows page, click **Create Window**.
2. Enter the following fields:
 - **Name:** Name of the window.
 - **Description:** Optional descriptive text.
 - **Enabled:** Enables the window. (Causes the DBMS_SCHEDULER.ENABLE procedure to be called after the window is created.)
 - **Resource Plan:** The resource plan that automatically activates when the window opens. When the window closes, the system switches to the appropriate resource plan, which is usually the plan that was in effect before the window opened, but can also be the plan of a different window. Only one resource plan can be associated with a window. If null, the resource plan in effect when the window opens stays in effect for the duration of the window. If an empty string, the resource manager is disabled for the duration of the window. If the window is open and the resource plan is dropped, then the resource allocation for the duration of the window is not affected.
 - **Duration:** The length of time that the window stays open. Can range from one minute to 99 days.
 - **Priority:** Relevant when two windows overlap. Because only one window can be in effect at one time, the window priority determines which window opens. The two possible values for this attribute are HIGH and LOW. A high priority window has precedence over a low priority window, therefore, the low priority window does not open if it overlaps a high priority window.
 - **Execution Mode:** Repeating Or Schedule:
 - For Repeating, specify the **Repeat Interval** and optionally the **Start Date** and **End Date**.

- For `Schedule`, select the name of the **Schedule** to be used.
3. Click **Create**.

Create Window Group

To create a window group:

1. In the Window Groups page, click **Create Window Group**.
2. Enter the following fields:
 - **Name**: Name of the window group.
 - **Description**: Optional comments.
 - **Enabled**: Enables the window group. (Causes the `DBMS_SCHEDULER.ENABLE` procedure to be called after the window group is created.)
 - **Available Windows**: Lists all Scheduler windows.
 - **Selected Windows**: List windows to be added to the window group. Use the arrow icons to move selected windows or all windows from one list to the other.
3. Click **Create**.

8

The Liquibase Page

The Liquibase page displays information of all database deployments made in the current schema. A deployment consists of changesets, which is a list of sequential changes to be applied to the database.

To navigate to the Liquibase page, do either of the following:

- In the Launchpad page, select the **Development** tab and click **Liquibase**.
- Click **Selector**  to display the navigation menu. Under **Development**, select **Liquibase**.

If you have not made any deployments into the schema and you access the Liquibase page, no data appears.

Topics:

- [Generate a Deployment](#)
- [About the Liquibase User Interface](#)
- [Generate a Deployment](#)
- [About the Liquibase User Interface](#)

Generate a Deployment

First, you have to make changes in the current schema such as creating a table, procedure or any database object.

Then, create a folder that will contain the changesets for the deployment. It is a good practice to create a different folder for each deployment. This ensures that you have information of all the deployments, in case you want to roll back to a previous version.

In the following example, Oracle SQLcl is used to deploy the changes.

1. Open the terminal and navigate to the path of the folder that you will use to save the changes made to the database (changelog files).
2. After you navigate to the correct path, log in to the schema you want to capture with the following command:

```
sqlcl <username>/<password>@<host>:<port>/<servicename>
```

3. You can do one of the following:
 - Deploy the entire schema with the following command:

```
lb genschema
```

This command creates a controller.xml file that includes all the changesets for the schema.

- Deploy a specific object using the following command:

```
lb genobject -type <object_type> -name <object_name>
```

This command generates the changelogs (which are XML files) that contain the changes made to the database in the current folder.

4. Log in to the schema where you want to add these changes (in this example, XYZ) with the following command:

```
sqlcl xyz/<password>@<host>:<port>/<servicename>
```

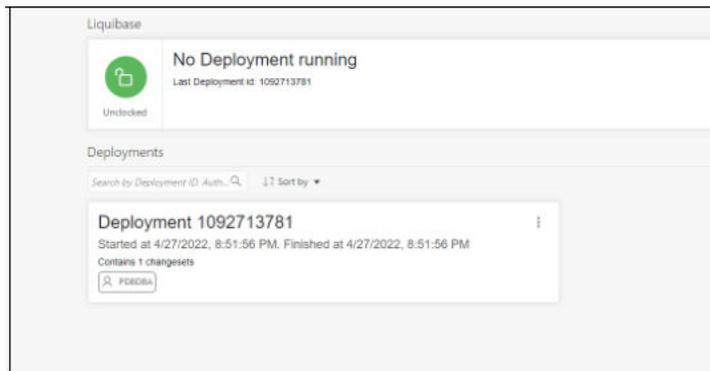
5. Run the following command:

```
lb update -changelog controller.xml (or the file name of your changelog)
```

6. Log in to Database Actions on the schema where you deployed the changes and the Liquibase page displays information about the deployment.

About the Liquibase User Interface

In the Liquibase page, the card at the top displays the Liquibase status. The deployments are displayed as cards below, as shown in the following figure.



Liquibase Status

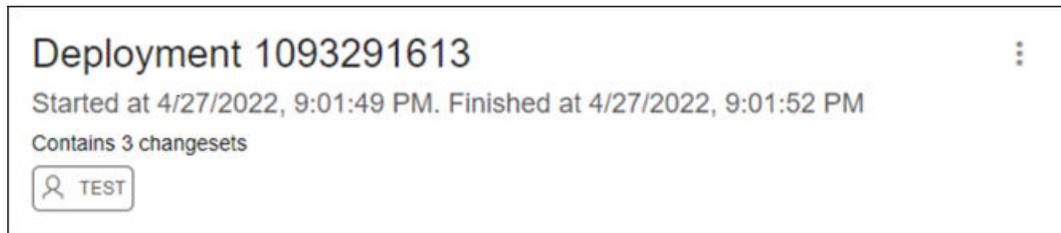
Liquibase runs only one deployment at a time. When there is no deployment running, the top Liquibase card indicates that Liquibase is currently unlocked and displays the ID of the last deployment.

If there is a deployment running at the time, the card indicates that Liquibase is locked, and displays the name of the resource blocking it and the time at which the deployment started.



Deployments

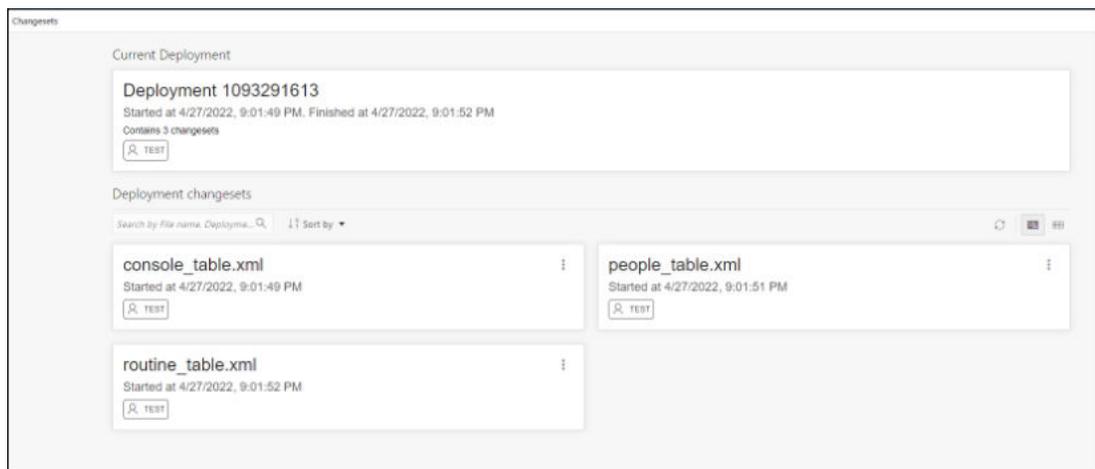
The card for each deployment provides details such as the schema, deployment id, start and complete deployment date and time, and number of changesets run in the deployment. To open a specific deployment page, click the title of the card, or click **Open Deployment Changes** in the context menu.



Changesets

In a deployment page, the changesets that have been run for that specific deployment is displayed below. The Current Deployment card at the top shows the deployment to which the changesets belong.

Each changeset card displays the changeset name, the date and time when the changeset started deploying, and the schema where the change was deployed.



The context menu for a changeset consists of the following options:

- **Previous Object State XML:** Displays an XML showing the state of the object (table, procedure) before the changeset was applied. If the object did not exist previously on the database (example, a table that was created in the deployment), the following message is displayed: `xml code not available....`
- **Show Executed SQL:** Displays the SQL code that was executed when the changeset was deployed.

Part II

Monitoring

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

This part provides information about the monitoring features available with Database Actions.

Topics:

- [The Monitoring Pages](#)
- [The Monitoring Pages](#)

9

The Monitoring Pages

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Monitoring menu provides access to several pages to view the performance and other characteristics of your database.

- [The Monitoring Overview Page](#)
- [The Performance Hub Page](#)
- [The Alerts Page](#)
- [The Sessions Page](#)
- [The Storage Page](#)
- [The Parameters Page](#)
- [The Database Dashboard Page](#)
- [The AWR Page](#)
- [The Monitoring Overview Page](#)
- [The Performance Hub Page](#)
- [The Alerts Page](#)
- [The Sessions Page](#)
- [The Storage Page](#)
- [The Parameters Page](#)
- [The Database Dashboard Page](#)
- [The AWR Page](#)

The Monitoring Overview Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Monitoring Overview page displays general information about the database.

To navigate to the Overview page, click **Selector**  to see the navigation menu and then select **Monitoring**.

- **Used Online Database Storage:** Displays how much storage is being used by the database. You can click the title to open the Storage page.
- **Users:** Displays how many user accounts are in the open, locked and expiring statuses. You can hover over one of the statuses to see a list of the user accounts with that status.
- **Sessions:** Displays the status of open sessions in the database. You can click the title to open the Sessions page.
- **Waits:** Displays how many wait events are occurring in the database for various reasons. You can click the title to open the Performance Hub page.

The Performance Hub Page

Note

Available only if you signed in to an Oracle Autonomous Database as a user with administrator rights.

The Performance Hub page shows performance data for a time period you specify.

To navigate to the Performance Hub page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **Performance Hub**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **Performance Hub**.

Note

The Performance Hub page is available in the following user interface languages: French, Japanese, Korean, Traditional Chinese, and Simplified Chinese. If you change the language to German, Spanish, Italian, or Portuguese in Preferences, the Performance Hub page reverts to English.

The Performance Hub page consists of these parts:

- **Time Range Area:** Use the controls in time range area at the top of the page to specify the time period for which you want to view performance data.
- **ASH Analytics Tab:** Use this tab to explore ASH (Active Session History) information across a variety of different dimensions for the specified time period.
- **SQL Monitoring Tab:** Use this tab to view the top 100 SQL statement executions by different dimensions for the specified time period, and to view details of SQL statement executions you select.

See [Performance Hub](#) in the Database service documentation for more information.

The Alerts Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Alerts page is a chronological log of messages and errors and is commonly used to learn whether the background processes have encountered errors. You can review the alert log periodically to verify that your database system is operating normally.

To navigate to the Alerts page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **Alerts**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **Alerts**.

The alert log includes the following:

- Nondefault initialization parameters used at startup
- Administrative operations, such as STARTUP, SHUTDOWN, ARCHIVE LOG, RECOVER, and CREATE/ALTER/ DROP DATABASE/TABLESPACE
- Messages and errors relating to the functions of certain background processes, such as LGWR
- Internal errors (ORA-600), block corruption errors (ORA-1578), and deadlock errors (ORA-60)

Click Refresh  at the top right of the page to refresh the data.

You can search for a specific value in the log by selecting the display column in the first drop-down list, selecting the condition in the second drop-down list, entering the search value in the box, and clicking the search icon.

In the display table, if you right-click the header row, you see:

Columns: Enables you to select columns to show or hide.

Sort: Displays a dialog box for selecting columns to sort by. For each column, you can specify ascending or descending order, and you can specify that null values be displayed first.

If you right-click any other part of the display table, you see:

Count Rows: Displays the number of rows in the table.

Single Record View: Enables you to view data for a table or view, one record at a time.

Copy: Enables you to copy data from a cell or a row or a range of rows. To copy from more than one row, select the rows you want to copy, right-click by pressing the SHIFT or CTRL key, and select **Copy**.

The Sessions Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Sessions page shows information about all currently open sessions in the database.

To navigate to the Sessions page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **Sessions**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **Sessions**.

The data is automatically refreshed at intervals ranging from 10 seconds to 2 minutes. You can also refresh the data by clicking Refresh  at the top right of the screen.

The table shows summarized data about each open session. Select a session in the table to see more detailed data in the Session Details table below, such as the last SQL statement, explain plan, waits, contention, and so on. You can use the Column, Operator and Value fields to search for the required sessions.

In the display table, if you right-click the header row, you see:

Columns: Enables you to select columns to show or hide.

Sort: Displays a dialog box for selecting columns to sort by. For each column, you can specify ascending or descending order, and you can specify that null values be displayed first.

If you right-click any other part of the display table, you see:

Count Rows: Displays the number of rows in the table.

Single Record View: Enables you to view data for a table or view, one record at a time.

Copy: Enables you to copy data from a cell or a row or a range of rows. To copy from more than one row, select the rows you want to copy by pressing the SHIFT or CTRL key, right-click and select **Copy**.

The Storage Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Storage page shows the storage used based on the current allocation of tablespaces along with additional drill-down capabilities to view segments.

To navigate to the Storage page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **Storage**.

- Click **Selector**  to display the navigation menu. Under Monitoring, select **Storage**.

You can refine the list of segments shown by using the filter feature. Click **View Datafiles** to view the datafiles in each tablespace.

You can view tablespace and segment space usage.

To view space usage information

1. From the Database drop-down menu, click **Storage**.

The Storage page displays. If the Oracle database is version 12c or later, the Storage page shows the used and allocated storage space for tablespaces in any pluggable database. If the Oracle database is version 11g, the Storage page shows the used and allocated space for the entire database.

2. You can click a tablespace to view its storage information. An interactive report appears, showing the segments that exist within the tablespace. Most segments are user objects, and they include tables, LOBs, and indexes.
3. On the Segments page, you can refine the list of segments shown by using the filter feature.

For example, you can search for all the segments for a specific owner (schema) by selecting OWNER from the first drop-down list, entering the owner (schema) name in the box, and clicking the search icon.

The Parameters Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The Parameters pages displays initialization parameters, which are used to configure the database instance, including memory structures, and define locations for database files.

To navigate to the Parameters page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **Parameters**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **Parameters**.

Values for initialization parameters are stored in a text-based initialization parameter file (PFILE) or binary server parameter file (SPFILE). The initialization parameter file is read at database instance startup.

Click Refresh  at the top right of the page to refresh the data.

To perform a search, enter values in the search criteria columns and click the search icon to locate the initialization parameter.

In the display table, if you right-click the header row, you see:

Columns: Enables you to select columns to show or hide.

Sort: Displays a dialog box for selecting columns to sort by. For each column, you can specify ascending or descending order, and you can specify that null values be displayed first.

If you right-click any other part of the display table, you see:

Count Rows: Displays the number of rows in the table.

Single Record View: Enables you to view data for a table or view, one record at a time.

Copy: Enables you to copy data from a cell or a row or a range of rows. To copy data from more than one row, select the rows you want to copy by pressing Shift or Ctrl, right-click and select **Copy**.

The Database Dashboard Page

Note

Available only if you signed in to an Oracle Autonomous Database as a user with administrator rights.

The Database Dashboard page provides information about the performance of an Autonomous Database.

To navigate to the Database Dashboard page, do either of the following:

- In the Launchpad page, click **Database Dashboard**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **Database Dashboard**.

The Database Dashboard page consists of the following tabs, which show real-time and historical information about the utilization of an Autonomous Database:

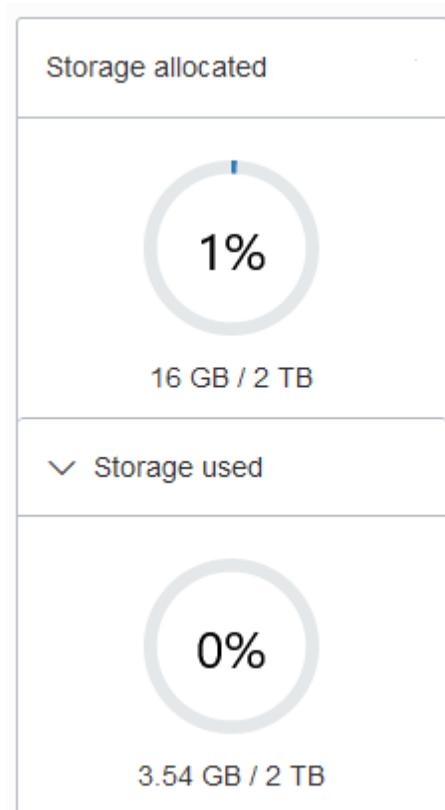
- Overview
- Monitor
- [Database Dashboard Overview](#)
The **Overview** tab shows real-time and historical information about the Autonomous AI Database utilization.
- [Database Dashboard Activity](#)
The **Monitor** tab shows real-time and historical information about the Autonomous AI Database performance data, activity, and utilization.

Database Dashboard Overview

The **Overview** tab shows real-time and historical information about the Autonomous AI Database utilization.

The charts shown on this page include:

- **Storage:** This chart shows the provisioned, allocated, and used storage. The chart indicates what percentage of the space is currently in-use.



Provisioned storage is the amount of storage you select when you provision the instance or when you modify storage by scaling storage.

Storage allocated is the amount of storage physically allocated to all data tablespaces and temporary tablespaces and includes the free space in these tablespaces. This does not include storage for the sample schemas.

Storage used is the amount of storage actually used in all data and temporary tablespaces. This does not include storage for the sample schemas. The storage used is the storage in the Autonomous AI Database as follows:

- Storage used by all database objects. Note: the chart does not include storage for the sample schemas as they do not count against your storage.
- Storage for files users put in the file system.
- Storage used by temporary tablespaces.
- Used storage excludes the free space in the data and temporary tablespaces.

By default the chart does not show the used storage. Select **Storage used** to expand the chart to see used storage (the values are calculated when you open the chart).

For an Autonomous JSON Database the chart shows an additional field showing the percentage of storage used that is not storing JSON documents.

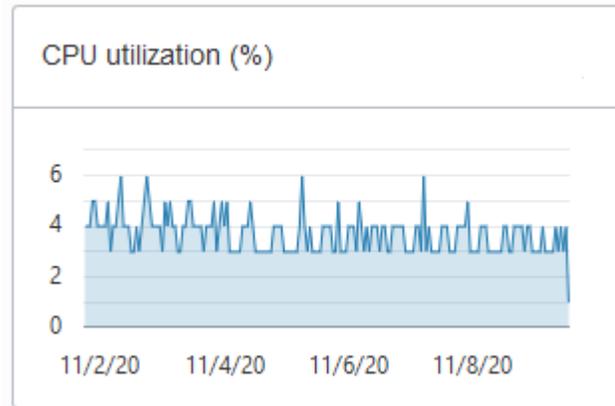
Note

If you drop an object, the space continues to be consumed until you empty the recycle bin. See [Purging Objects in the Recycle Bin](#) for more information.

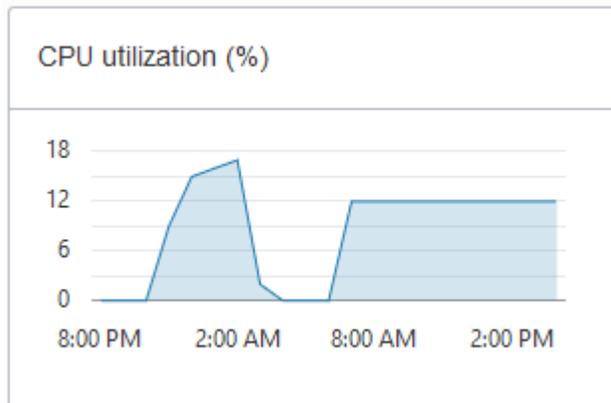
See Use Sample Data Sets in Autonomous Database for information on sample schemas SH and SSB.

- **CPU utilization (%) for ECPU Compute Model:** This chart shows the historical CPU utilization of the service:
 - Compute auto scaling disabled: this chart shows hourly data. A data point shows the average CPU utilization for that hour. For example, a data point at 10:00 shows the average CPU utilization for 9:00-10:00.

The utilization percentage is reported with respect to the number of ECPUs the database is allowed to use. For example, if the database has four (4) ECPUs, the percentage in this graph is based on 4 ECPUs.

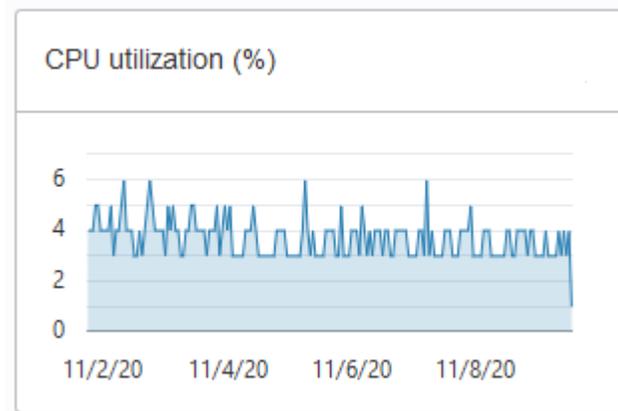


- Compute auto scaling enabled: For databases with compute auto scaling enabled the utilization percentage is reported with respect to the maximum number of ECPUs the database is allowed to use, which is three times the number of ECPUs. For example, if the database has four (4) ECPUs with auto scaling enabled, the percentage in this graph is based on 12 ECPUs.

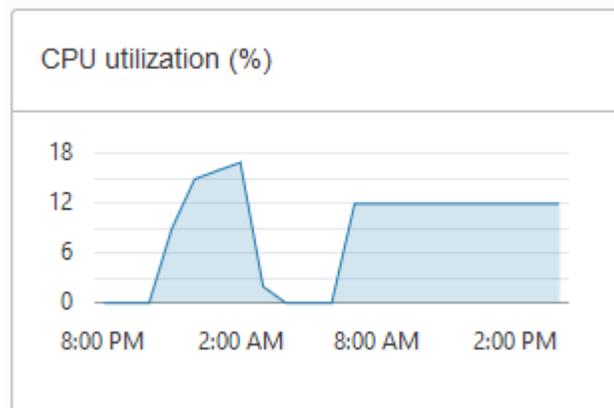


- **CPU utilization (%) for OCPU Compute Model:** This chart shows the historical CPU utilization of the service:
 - Compute auto scaling disabled: this chart shows hourly data. A data point shows the average CPU utilization for that hour. For example, a data point at 10:00 shows the average CPU utilization for 9:00-10:00.

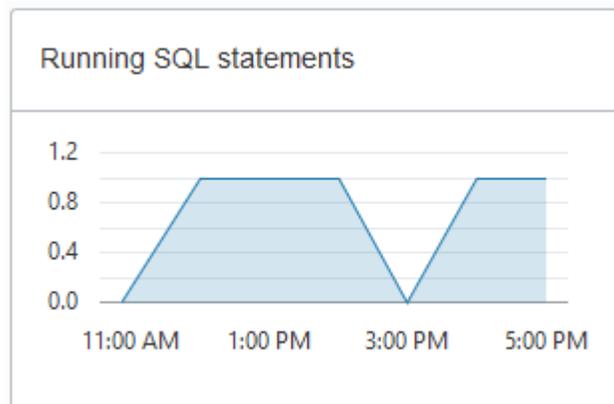
The utilization percentage is reported with respect to the number of OCPUs the database is allowed to use. For example, if the database has four (4) OCPUs, the percentage in this graph is based on 4 OCPUs.



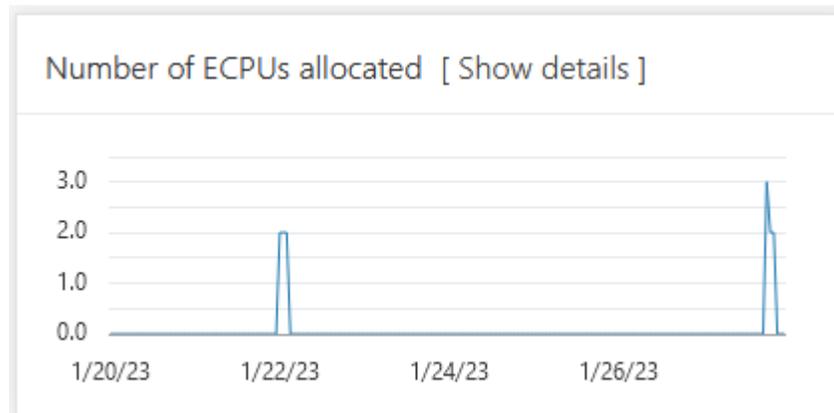
- **Compute auto scaling enabled:** For databases with compute auto scaling enabled the utilization percentage is reported with respect to the maximum number of OCPUs the database is allowed to use. For example, if the database has four (4) OCPUs with auto scaling enabled, the percentage in this graph is based on 12 OCPUs.



- **Running SQL statements:** This chart shows the average number of running SQL statements historically. This chart shows hourly data. A data point shows the running SQL statements for that hour. For example, a data point at 10:00 shows the average number of running SQL statements for 9:00-10:00.



- **Number of ECPUs allocated (only shown for ECPU Compute Model):**



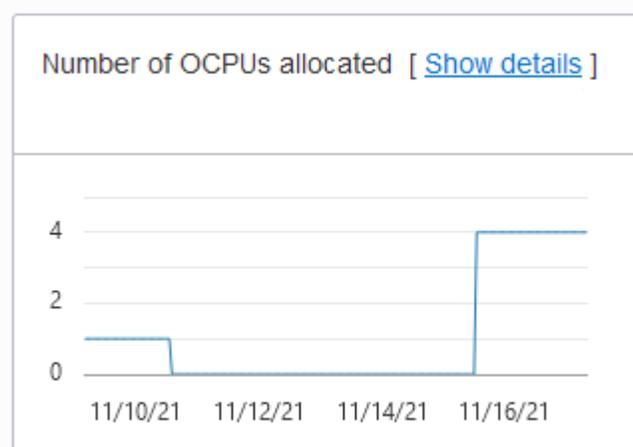
Notes for display results:

- Compute auto scaling disabled: For databases with compute auto scaling disabled, for each hour the chart shows the number of ECPUs allocated to the database if the database is open for at least some part of the hour.
- Compute auto scaling enabled: For databases with compute auto scaling enabled, for each hour the chart shows the average number of ECPUs used during that hour if that value is higher than the number of ECPUs provisioned. If the number of ECPUs used is not higher than the number of ECPUs provisioned, then the chart shows the number of ECPUs allocated for that hour.
- Stopped Database: If the database was stopped for the full hour the chart shows 0 ECPUs allocated for that hour.

Click **Show details** for more information, including the number of ECPUs allocated to the database and to external resources, and the total allocated ECPUs.

The Show details view includes separate values for database ECPU usage and external resource ECPU usage. External resources include: Cloud SQL, Graph, OML4PY, and others. The Total ECPUs are the total number of ECPUs in use on the Autonomous AI Database. The external ECPUs value shows how external ECPUs contribute to the total ECPU usage.

- **Number of OCPUs allocated (only shown for OCPU Compute Model):**



Notes for display results:

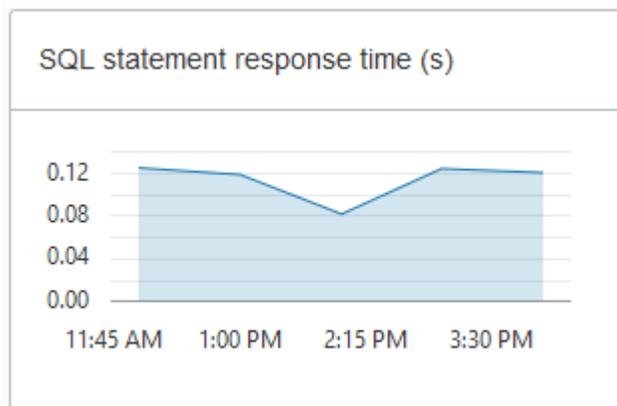
- Compute auto scaling disabled: For databases with compute auto scaling disabled, for each hour the chart shows the number of OCPUs allocated to the database if the database is open for at least some part of the hour.

- Compute auto scaling enabled: For databases with compute auto scaling enabled, for each hour the chart shows the average number of OCPUs used during that hour if that value is higher than the number of OCPUs provisioned. If the number of OCPUs used is not higher than the number of OCPUs provisioned, then the chart shows the number of OCPUs allocated for that hour.
- Stopped Database: If the database was stopped for the full hour the chart shows 0 OCPUs allocated for that hour.

Click **Show details** for more information, including the number of OCPUs allocated to the database and to external resources, and the total allocated OCPUs.

The Show details view includes separate values for database OCPU usage and external resource OCPU usage. External resources include: Cloud SQL, Graph, OML4PY, and others. The Total OCPUs are the total number of OCPUs in use on the Autonomous AI Database. The external OCPUs value shows how external OCPUs contribute to the total OCPU usage.

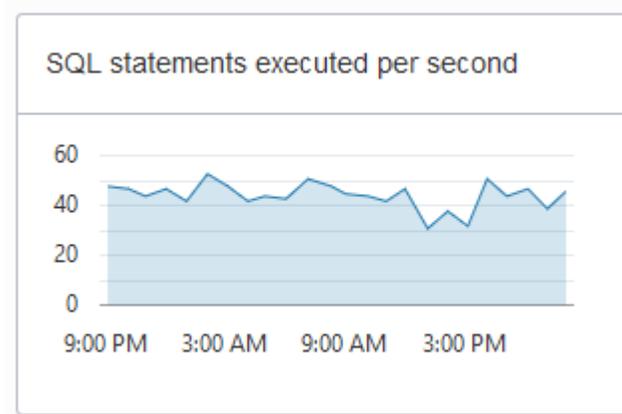
- **SQL statement response time (s):** This chart shows the average response time, in seconds, of SQL statements historically. This chart shows hourly data. A data point shows the average SQL statement response time for that hour. For example, a data point at 10:00 shows the average SQL statement response time, in seconds, for the hour from 9:00-10:00.



- **SQL statements executed per second**

Note

Database Dashboard does not show this chart when the Autonomous AI Database instance workload type is **Data Warehouse**.



The default retention period for performance data is thirty (30) days. The CPU utilization, running statements, and average SQL response time charts show data for the last eight (8) days by default.

You can change the retention period by modifying the Automatic Workload Repository retention setting with the PL/SQL procedure

`DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS()`. The maximum retention you can set is 30 days. See *Oracle AI Database PL/SQL Packages and Types Reference*.

If you need to store more performance data you can use the AWR Hub. See [Analyze Automatic Workload Repository \(AWR\) Performance Data](#) for more information.

Database Dashboard Activity

The **Monitor** tab shows real-time and historical information about the Autonomous AI Database performance data, activity, and utilization.

Note

The default view in the **Monitor** tab is real-time. This view shows performance data for the last hour.

The charts on this page are:

- **Database Activity**

This chart shows the average number of sessions in the database using CPU or waiting on a wait event. See *Oracle AI Database Reference* for more information on wait events.

- **CPU Utilization** (with ECPU compute model)

This chart shows the CPU utilization of each consumer group. The utilization percentage is reported with respect to the number of ECPUs the database is allowed to use. For example, if the database has four (4) ECPUs, the percentage in this graph is based on 4 ECPUs.

For databases with compute auto scaling enabled the utilization percentage is reported with respect to the maximum number of ECPUs the database is allowed to use, which is three times the number of ECPUs. For example, if the database has four (4) ECPUs with auto scaling enabled, the percentage in this graph is based on 12 ECPUs.

See Manage Concurrency and Priorities on Autonomous Database for detailed information on consumer groups.

- **CPU Utilization** (with OCPU compute model)

This chart shows the CPU utilization of each consumer group. The utilization percentage is reported with respect to the number of OCPUs the database is allowed to use. For example, if the database has four (4) OCPUs, the percentage in this graph is based on 4 OCPUs.

For databases with compute auto scaling enabled the utilization percentage is reported with respect to the maximum number of OCPUs the database is allowed to use. For example, if the database has four (4) OCPUs with auto scaling enabled, the percentage in this graph is based on 12 OCPUs.

See Manage Concurrency and Priorities on Autonomous Database for detailed information on consumer groups.

- **Running Statements**

This chart shows the average number of running SQL statements in each consumer group.

See Manage Concurrency and Priorities on Autonomous Database for detailed information on consumer groups.

- **Queued Statements**

This chart shows the average number of queued SQL statements in each consumer group.

See Manage Concurrency and Priorities on Autonomous Database for detailed information on consumer groups.

To see earlier data click **Time period**. The default retention period for performance data is thirty (30) days. By default in the Time Period view the charts show information for the last eight (8) days.

In the time period view you can use the calendar to look at a specific time period in the past 30 days. You can also use the time slider to change the period for which performance data is shown.

Note

The retention time can be changed by changing the Automatic Workload Repository retention setting with the PL/SQL procedure `DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS`. Be aware that increasing the retention time results in more storage usage for performance data. See *Oracle AI Database PL/SQL Packages and Types Reference*.

The AWR Page

Note

Available only if you signed in to an Oracle Autonomous Database on dedicated Exadata infrastructure as a user with administrator rights.

The AWR page in Database Actions enables you to generate, view and download Automatic Workload Repository (AWR) reports.

To navigate to the AWR page, do either of the following:

- In the Launchpad page, select the **Monitoring** tab and click **AWR**.
- Click **Selector**  to display the navigation menu. Under Monitoring, select **AWR**.

The Automatic Workload Repository collects, processes, and maintains performance statistics for the database. The gathered data can be displayed in reports and views.

To generate or view a report in the AWR page, select the range of the time period required using the **Start ID** and **End ID** fields. The snapshot dropdown list for each field is sorted in descending order and starts with the most recent database snapshot.

Note

The **Generate Report** button is available only when the value in the Start ID field is lower than the value in the End ID field.

Monitoring > AWR Report

Start ID: 3 End ID: 4 [Generate Report](#) [Download Report](#)

WORKLOAD REPOSITORY PDB report (root snapshots)

DB Name	DB Id	Unique Name	Role	Edition	Release	RAC	CDB
DB213C	285474074	DB213C	PRIMARY	EE	21.0.0.0.0	NO	YES
Instance	Inst Num	Startup Time	User Name	System Data Visible			
DB213C	1	06-Jun-22 06:09	PDBDBA	YES			
Container DB Id	Container Name	Open Time					
345142159	DB213P	06-Jun-22 06:09					
Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)		
db213	Linux x86 64-bit	48	24	2	500		
	Snap Id	Snap Time	Sessions	Cursors/Session			
Begin Snap:	3	06-Jun-22 07:58:33	36				
End Snap:	4	06-Jun-22 08:58:47	29				

Part III

Data Tools

This part provides information about the following topics:

Topics:

- [The Data Pump Page](#)
- [The Data Load Page](#)
- [The Table AI Assist Tool](#)
- [The Catalog Tool](#)
- [The Data Transforms Page](#)
- [The Data Insights Page](#)
- [The Data Analysis Tool](#)
- [The Data Marketplace tool](#)
- [The Data Share Tool](#)
- [The Jobs Feature](#)
- [The Data Studio Overview Page](#)

The Data Studio comprises of the Data Load, the Data Analysis, the Data Insights, the Catalog tool, the Table AI Assist tool, the Data Marketplace tool, Jobs feature and the Data Share tool.
- [Python Data Studio API for Oracle Autonomous Database](#)

The Python Data Studio Application Programming Interface (API) for Oracle Autonomous Database services provides the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to Oracle Autonomous Database.
- [The Data Load Page](#)

Use the Data Load dashboard page to make more data available to your Oracle Autonomous Database. You can load data from files or databases, from links to external databases or cloud storage files, or from a live feed of data from cloud storage.
- [The Catalog Tool](#)

The Catalog Tool allows you browse, search and discover data in, and connected to, the Oracle Autonomous Database.
- [The Data Transforms Page](#)

Data Transforms allows you to design data transformations in the form of data loads, data flows, and workflows, without requiring you to write any code.
- [The Data Insights Page](#)

The Data Insights page displays information about patterns and anomalies in the data of entities in your Oracle Autonomous Database.
- [The Data Analysis Tool](#)

The Data Analysis tool enables you to create Analytic Views with multidimensional metadata.

- [The Data Marketplace Tool](#)
Data Marketplace provides an innovative platform to access and share datasets effortlessly.
- [The Table AI Assist Tool](#)
The **Table AI Assist Tool** enables you to add, remove, and rename columns in a table using AI assistance. It enables you to augment or correct data found in tables in your local schema using natural language AI prompts. For example, prepare data for analytics by adding columns for days of the week or reporting quarters or deriving the distance between two sets of geographical co-ordinates, and more.
- [The Jobs Feature](#)
The Jobs feature automates the tasks run by the Data Studio suite of tools. It can be scheduled to run at specific intervals or triggered by certain events.
- [The Data Share Tool](#)
The Data Share tool allows you to share Oracle data and metadata with other databases and non-database tools. The Data Share page displays information about the different types of shares available in the Oracle Autonomous Database.

The Data Studio Overview Page

The Data Studio comprises of the Data Load, the Data Analysis, the Data Insights, the Catalog tool, the Table AI Assist tool, the Data Marketplace tool, Jobs feature and the Data Share tool.

The Data Studio Tools enables you to load data from cloud and other diverse sources, analyzes it and gain insights from it. You can share the result of the analysis with other users. It is a one-stop application of your analytics tool from multiple data sources. This tool makes sure that there is seamless transition between different applications. The multiple ways of navigation do not impact the progress of your work. For instance, if you are working on data analysis and decide you need some additional data, you can navigate to the Data Load page, bring in the new contents, and return to your analysis in progress. Select **Data Studio** tab from the Launchpad.

Data Studio Overview page

Click the **Data Studio Overview** menu from the **Data Studio** to view the Data Studio Overview page consists of four sections:

1. A navigation pane which consists of the following tools in the menu: Data Load, Data Analysis, Data Insights, the Catalog and the Data Share tools.
2. A widget with slides which displays information on the tools and their purpose. For example, the Data Load tool loads data from files on your computer or from cloud storage. Click < or > button to navigate back and forth the slides.
3. The right pane toolbar displays the help links to the Data Studio tool documentation.
4. A Recent Objects section which displays recently updated or created objects. Each entry has a context menu available at the end of the row. Click on the entries to view the details of the selected object.

Load your data from diverse sources such as files, databases and cloud storage, which are all consolidated in a single location. Some of the cloud data warehouses the Data Studio tool supports are Oracle Cloud Infrastructure (OCI), Amazon S3, Microsoft Azure Blob Storage and Google Cloud Storage.

Once your data is available, analyze it, generate insights, and create reports.

On the right of the Data Studio Overview page, refer to the Getting Started panel to know more about this tool.

Note

You will lose your data once you click the Database Actions in the header or click the selector icon.

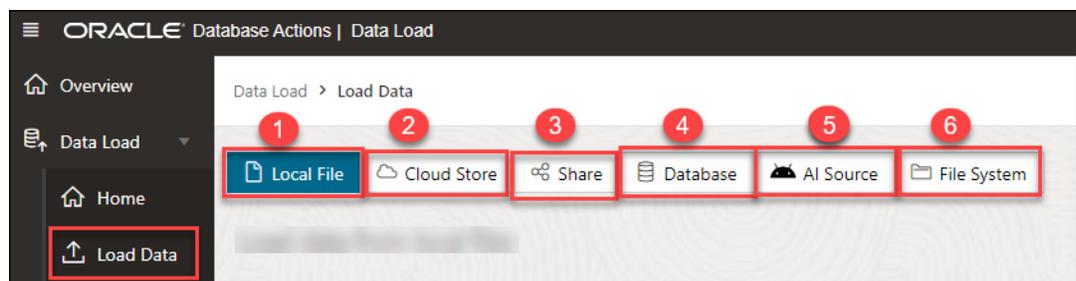
Let us run analysis with data from different sources and generate insights from it.

Load Data

Select the **Data Load** menu from the navigation pane in the Data Studio Overview page to load data from files on your local device, from remote databases, or from cloud storage buckets. Clicking Data Load opens Data Load Dashboard page.

The Data Load menu has the following submenus:

- **Home:** This menu takes you to the home page of the Data Load tool in the Launchpad. You can perform any of the following activities from the submenus such as Load local file, Load cloud store, Load Database Tables, Link Cloud Store, Link Database Tables, Create Live feed and search for cloud location and then select the Home menu. This does not break the continuity of the action you perform. Refer to [The Data Load Page](#) for more details.
- **Load Data:** Click **Load Data** to view the following options to load data:



- **Load Local File:** This option enables us to load data from files on your local device. Refer to the [Loading Data From Local Files](#) section for more details.
- **Load Cloud Store:** This menu enables you to load data from a cloud store location to a table in your Autonomous AI Database. Refer to the [Loading Data from Cloud Storage](#) section for more details.
- **Load Share:** This menu enables you to load data from Shares where you can select tables from a Share. See [Loading Data from a Share](#).
- **Load Database:** This menu enables you to load data from tables in another database into your Oracle Autonomous AI Database. Refer to the [Loading Data from other Databases](#) section for more details.
- **Load AI Source:** This menu enables you to load data from an AI source. See [Loading Data from AI](#).
- **Load File System:** This menu enables you to load data from file system directories to your Autonomous AI Database.
- **Link Data:** Click **Link Data** to view the following options of linking data:



- **Link Cloud Store:** This menu enables you to link to data in cloud storage buckets. Refer to the [Linking to Objects in Cloud Storage](#) section for more details.
- **Link Share:** This menu enables you to link a share with your Autonomous AI Database. See [Linking to Share](#).
- **Linking Data Catalog :** This menu enables you to link Data Catalog to your Autonomous AI Database. See [Linking Data Catalog](#).
- **Link Database:** This menu enables you to link to data in tables in another database from your Oracle Autonomous AI Database. Refer to the [Linking to Other Databases](#) section for more details.
- **Linking File System:** This menu enables you to link file system directories to your Autonomous AI Database. See [Linking to File System](#).
- **Live Feed:** This menu enables us to create a live table feed on demand, on a schedule, or as the result of a notification. Refer to the [Feeding Data](#) section for more details.
- **Connections:** You can view all the existing connections from this option. This option enables you to edit or delete a cloud storage link and manage catalogs and shares. You can also create new cloud storage links. Refer to the [Managing Connections](#) chapter for more details.

After you load the data using the data load job, it becomes available in the Data Load page. You can later inspect, review and delete the table on clicking the **Actions** icon. You could also reload the cart from the Reload Cart button to reload cart using the current cart in case you want to make changes to the table. This data will be the source data of the Analytic View. Use breadcrumbs at the top of the page to navigate back to the home page.

Follow the steps described in [The Data Load Page](#) chapter in this book to load data from Oracle Cloud Infrastructure to your Autonomous AI Database.

Data Analysis

Analytic views provide metadata that highlight the parts of your data that are most important to your organization. By creating an AV, you can improve the value of automatic insights and the analysis of your data because the system has this metadata.

Select the **Data Analysis** menu from the navigation pane in the Data Studio Overview page to create Analytic Views and analyze the data from the Analytic View. The table you loaded using the Load Data feature will be used to create the Analytic View. You can select both hierarchies and measures from Analytic Views. Analytic Views utilize diagrams, symbols, and text, that represents how the data flows and the connection in between them. The Data Analysis tool helps you find the structural errors in the Analytic View you create. It spots anything from null values in a join column to duplicated fields. You can develop a visual representation of your analysis in the form of tables, Pivot tables and bar charts. You can also export the Analytic View to Tableau or Power BI for better visualization.

Create Analytic Views by referring to [The Data Analysis Tool](#) chapter in this book.

The **Related Insights** panel in the right works with the measures associated with the Analytic View to visualize data in the form of bar charts.

Table AI Assist

Select the **Table AI Assist** menu from the navigation pane in the Data Studio Overview page to effortlessly enhance your local data tables with the power of AI. You can use natural language prompts to augment and correct your information with this tool.

Refer to [The Table AI Assist Tool](#) chapter of this book for more information on the Table AI Assist page.

Data Insights

Select the **Data Insights** menu from the navigation pane in the Data Studio Overview page to generate insights automatically. The insights appear in the Data Insights dashboard as a series of bar charts.

By selecting the user, Analytic View, and the column name, you can view the data points of the actual values that deviate considerably from the forecast values. This type of predictive analytics will enable you to understand a pattern better and make better decisions. Then you can act on your data to optimize it or to improve its performance.

For example, consider that the fact table for the insights records values about different companies, and the measures of the fact table are Value (USD), Acquisition date and Acquisition year. Then you can generate insights for the Value measure. The dashboard will have a series of charts labeled Acquisition date and Acquisition year. Let's say you want to view the value of your company **A**. Click the chart whose sale you want to display for a particular range of Acquisition year. Clicking a chart displays more details about it.

Generate automatic insights by referring to [The Data Insights Page](#) in this book.

Below is the chart that displays the values and insights.



According to the insights generated by the tool, you notice that your company is lagging from the expected value by a significant number. In order to lead you need to change a few factors that affect the sales of your organization.

The insights that Data Insights generates for the analytic view are more useful than those for a table because of the additional metadata that an analytic view provides.

Catalog

Select the **Catalog** menu from the navigation pane in the Data Studio Overview page to view information about the upstream dependencies of the entity, how the entity was created and how it is linked to other entities.

This page lists details about the database objects such as cloud storage links and tables you create from the Data Analysis tool and the Data load tool. It also displays lists of the database dictionary objects such as tables, columns, and Analytic Views.

Use the Catalog page to locate objects in the catalog and perform tasks specific to those objects.

Refer to [The Catalog Tool](#) chapter of this book for more information on the Catalog page.

Data Marketplace

Select the **Marketplace** menu from the navigation pane in the Data Studio Overview page to easily exchange metadata from data providers to data consumers, allowing easy data discovery and use.

Refer to [The Data Marketplace Tool](#) chapter of this book for more information on the Data Marketplace.

Data Share

Select the **Data Share** menu from the navigation pane in the Data Studio Overview page to provide data shares and consume data shares.

See [The Data Share Tool](#)

Jobs feature

Select the **Jobs** feature menu from the navigation pane in the Data Studio Overview page to automate the tasks performed by Data Studio suite of tools. With this feature, you can create jobs that can load data, augment table, make REST calls and run PL/SQL procedures.

Refer to [The Jobs Feature](#) chapter of this book for more information on the Jobs feature.

11

Python Data Studio API for Oracle Autonomous Database

The Python Data Studio Application Programming Interface (API) for Oracle Autonomous Database services provides the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to Oracle Autonomous Database.

Get Account Information

The account creation email from the Autonomous Database service contains the user name and password for your Oracle Autonomous Database account. If you don't have this information, then contact your Autonomous Database administrator.

See [Manage Users and User Roles on Autonomous Database - Connecting with Database Actions](#) for more information.

Note

The package will be available in OML notebooks.

Topics:

- [Working with Python Data Studio API for Autonomous Database](#)
- [Working with Python Data Studio API for Autonomous Database](#)
Python Data Studio API package works both in OML notebook and as a standalone application.

Working with Python Data Studio API for Autonomous Database

Python Data Studio API package works both in OML notebook and as a standalone application.

You can view the REST API endpoints hosted on Oracle Autonomous Database. These endpoints allow you to store Machine Learning models along with their metadata, and create scoring endpoints for the model.

Connecting to Autonomous Database

To connect to the Autonomous Database via your username and password, run the following command:

```
import adp
adp_instance = adp.login('<url>', '<username>', '<password>')
```

Here are the parameters and their descriptions:

`url`: This specifies the Database Actions host. For example, `https://abcdefg.oraclecloudapps.com`.

`username`: This specifies the schema name.

`password`: This specifies the password to the schema.

Connecting inside OML Notebooks

Login inside OML notebooks by entering the following command:

```
import adp
adp_instance = adp.connect()
```

The return value of the login method is an instance of class that contains all Autonomous Data Platform (ADP) functions.

The `Adp` class instance consists of the following inner classes:

- `Ingest`
- `Analytics`
- `Insight`
- `Miscellaneous`
- `Data Share`

Refer to the following functions in the subsequent chapter.

Topics:

- [Ingest functions](#)
- [Analytic View functions](#)
- [Insight functions](#)
- [Miscellaneous functions](#)
- [Data Share Functions](#)
- [Ingest Functions](#)
The `Adp.Ingest` class is used for loading data from different sources into tables and views.
- [Analytic View Functions](#)
The class **Adp.Analytics** provides the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to Analytic Views tool of Data Studio Suite of tools. If the requested analytic view does not exist (except create), all functions return message `Analytic view does not exist`.
- [Insight Functions](#)
Insight functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to the Insight tool in Data Studio suite of tools. Generation of insight is performed in background and requires checking for the completion.
- [Miscellaneous functions](#)
Miscellaneous functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface.
- [Data Share Functions](#)
The Python Data Studio Application Programming Interface (API) for Data Share functions provides the syntax and descriptions of the classes, methods, attributes, and parameters

of the application programming interface to the Data Share tool of Data Studio suite of tools.

- [Catalog Functions](#)

The Oracle Data Studio Catalog or Catalog is a multi-catalog tool that provides a way to search for data and other objects in your currently connected Autonomous Database, and also in a wide range of other connected systems.

Ingest Functions

The `Adp.Ingest` class is used for loading data from different sources into tables and views.

The Autonomous Database Load Data API for Python allows the following operations:

- Copy tables from the Database Link to tables or views of the current schema
- Copy objects from the Cloud Storage Link to tables or external tables of the current schema
- Create table with data from json

These operations create the corresponding Ingest Job

Database Link operations

Database Link functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to the Database Link.

Get Consumer Group Procedure

This function receives list of consumer groups to run your SQL or PL/SQL code. The values match the database services available when connecting to the database. This feature is available only if you have the `EXECUTE` privilege on the `CS_SESSION` package.

Syntax

```
Ingest.get_consumer_groups()
```

Example

In this example, you get the list of the consumer groups:

```
adp.Ingest.get_consumer_groups() //Output: ['LOW', 'MEDIUM', 'HIGH']
```

Get Database Links Procedure

Return list of available database links (to create database link see Database Links [Database Links](#) in "Oracle® Database Database Administrator's Guide 21c").

```
Ingest.get_database_links(owner)
```

Parameters:

- **owner:** This field displays the owner of Database Links. If this field is missing, the tool uses the current schema owner.

Example

In this example, you get the list of the Database Links:

```
adp.Ingest.get_database_links()

//Output
{"nodes":
[
  {"label": "*****_*****_SH.REGRESS.RDBMS.DEV.US.ORACLE.COM",
    "type": "DB_LINK",

    "id": "\"ADPTEST\".\"DB_LINK\".\"*****_*****_SH.REGRESS.RDBMS.DEV.US.
ORACLE.COM\"",
    "data": {
      "name": "*****_*****_SH.REGRESS.RDBMS.DEV.US.ORACLE.COM",
      "namespace": "DB_LINK",

      "path": "\"DB_LINK\".\"*****_*****_SH.REGRESS.RDBMS.DEV.US.ORACLE.CO
M\"",
      "schema": "ADPTEST",
      "application": "DATABASE",
      "created": "2024-05-02T08:30:15Z",
      "updated": "2024-05-02T08:30:15Z"
    }
  }
], "links": []}
```

Get Tables in the Database Link Procedure

Return list of tables and views from the database link.

```
Ingest.get_db_link_owner_tables(db_link)
```

Parameters:

- **db_link**: This name of the Database Link.

Example

In this example, you get the list of the objects in a specified Database Link:

```
adp.Ingest.get_db_link_owner_tables("*****_*****_SH.REGRESS.RDBMS.
DEV.US.ORACLE.COM")

//Output
[
  "APEX_DG_DATASET_ROWS": {
    "dbLink": "*****_*****_SH.REGRESS.RDBMS.DEV.US.ORACLE.COM",
    "owner": "APEX_230200",
    "tableName": "APEX_DG_DATASET_ROWS",
    "numRows": 3,
    "avgRowLen": 54
  }, ...
]
```

Copy Table from Database Link Procedure

Create table from the specified tables into the current database.

Syntax

```
Ingest.copy_tables_from_db_link(tables, consumer_group)
```

Parameters:

- **tables:** Array of tables description. Each description has 4 fields.
- **owner:** This field displays the owner of the Database Link. If this field is missing, the tool uses the current schema owner.
- **table_name:** the name of the table in the Database Link. This field is mandatory.
- **db_link:** The name of the Database Link. This field is mandatory.
- **target_table_name:** The name of the target table. If this field is missing, the tool uses the source table name.
- **consumer_group:** the consumer group. Default value is "LOW".

Examples

In this example, you can create table based on the table in the Database Link:

```
adp.Ingest.copy_tables_from_db_link([{'owner': 'SH', 'tableName': 'PRODUCTS',
'dbLink': 'PHOENIX119757_ORDS_SH.REGRESS.RDBMS.DEV.US.ORACLE.COM', 'targetTableName': 'PRODUCT'}], 'HIGH')
```

```
//Output
```

```
[
  {
    'schema': 'SH',
    'tableName': 'PRODUCTS',
    'targetTableName': 'PRODUCTS',
    'name': '*****_*****_SH.REGRESS.RDBMS.DEV.US.ORACLE.COM',
    'rowsCopied': 766
  }
]
```

The method returns the list of tables you create. The dictionary of the output is the same as in input parameters, and additional value is inserted row count.

Link Table from Database Link Procedure

Create view from the specified tables into the current database. Take into account that the result string has no number of rows.

```
Ingest.link_tables_from_db_link(tables, consumer_group)
```

Parameters:

- **tables:** Array of tables description. Each description has 4 fields
 - **owner:** This field displays the owner of the Database Link. If this field is missing, the tool uses the current schema owner.
 - **table_name:** the name of the table in the Database Link. This field is mandatory.

- **db_link**: The name of the Database Link. This field is mandatory.
- **target_table_name**: The name of the target table. If this field is missing, the tool uses the source table name.
- **consumer_group**: the consumer group. Default value is "LOW".

Example

In this example, you can create table based on the table in the Database Link:

```
adp.Ingest.link_tables_from_db_link([[{'owner': 'SH', 'tableName': 'PROMOTIONS',
'dbLink': '***** _***** _SH.REGRESS.RDBMS.DEV.US.ORACLE.COM',
'targetTableName': 'PROMOTIONS'}]], 'HIGH')
```

//Output

```
[
  {
    'schema': 'SH',
    'tableName': 'PROMOTIONS',
    'targetTableName': 'PROMOTIONS',
    'name': '***** _***** _SH.REGRESS.RDBMS.DEV.US.ORACLE.COM' }
]
```

Cloud Storage Link operations

Cloud Storage Link functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to the Cloud Storage Link tool of Data Studio suite of tools.

These methods allows managing Credentials and Cloud Storage Link, and import cloud objects to tables or external tables of the current schema.

Get List of Credentials Procedure

Return list of available Credentials

```
Ingest.get_credential_list()
```

The above function returns a list of Credentials.

Example

In this example, you can get all Credentials:

```
adp.Ingest.get_credential_list()

{
  "items": [
    {
      "credential_name": "PROVIDER_DEMO_DELTA_SHARING_111$SHARE_CRED",
      "username": "BEARER_TOKEN",
      "windows_domain": null,
      "comments": "{\"comments\": \"Created via
DBMS_CLOUD.create_credential\", \"user_comments\": \"\"}"
    },
    ...
  ]
}
```

```
    ]
}
```

Create Credential Procedure

Create Simple Credential. For Microsoft Azure cloud use is an account name, and password is an access key, for Amazon user is an access key ID, and password is secret access key, for Google user is HMAC access key, and password is HMAC access secret.

Syntax

```
Ingest.create_credential(credential_name, user, password)
```

Parameters:

- **credential_name**: the name of a new credential
- **user** and **password** depend on Cloud system: For Microsoft Azure cloud **user** is an account name, and **password** is an access key, for Amazon **user** is an access key ID, and **password** is secret access key, for Google **user** is HMAC access key, and **password** is HMAC access secret.

Example

In this example, you can create a Credential with specified user and password:

```
adp.Ingest.create_credential('TEST', 'ADMIN', 'PASSWORD')
```

Create OCID Credential Procedure

Create credentials using OCI Signing Keys.

Syntax

```
Ingest.create_ocid_credential(credential_name, user_ocid, tenancy_ocid,
                              private_key, fingerprint)
```

This function creates credentials using OCI Signing Keys.

Parameters:

- **credential_name**: The name of the credential.
- **user_ocid**: The OCID of the the user.
- **tenancy_ocid**: The OCID of the tenancy.
- **private_key**: The private key in the RSA key pair. The private key spans over multiple lines. Ensure to replace all the newline characters with the space character and use the resulting key.
- **fingerprint**: The fingerprint for the RSA key pair that you're using to access OCI.

Example

```
adp.Ingest.create_ocid_credential('OCI_NATIVE_CRED',
'ocidl.user.ocl..aaaaaaatfn77fe3fxux3o5lego7glqjejrzjsqsrs64f4jsjrhbsk5qzndq'
, 'ocidl.user.ocl..aaaaaaatfn77fe3fxux3o5lego7glqjejrzjsqsrs64f4jsjrhbsk5qzndq'
,
'
```

```
'MIIeogIBAAKCAQEAsbNPOYEKxM5h0DF+qXmie6ddo95BhlSMSIxRRS01JEMPeSta0C7WEg7g8SOSz
hIroCkgOqDzkcyXnk4Bl0dn5Wm/BYpdAtTXk0sln2DH... ',
'4f:0c:d6:b7:f2:43:3c:08:df:62:e3:b2:27:2e:3c:7a')
```

Drop Credential Procedure

Drop credential with the specified name.

```
Ingest.drop_credential(credential_name)
```

Parameters:

- **credential_name:** The name of the credential.

Example

In this example, you can drop Credential with the name 'TEST':

```
adp.Ingest.drop_credential('TEST')
```

Get Cloud Storage Links List Procedure

Syntax

```
Ingest.get_cloud_storage_link_list(owner)
```

Parameters:

- **owner:** This field displays the owner of the Database Link. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can the list of Cloud Storage Links :

```
Ingest.get_cloud_storage_link_list()
```

```
// Output
```

```
{
  "nodes": [
    {
      "label": "AA",
      "type": "CLOUD_STORAGE_LINK",
      "id": "\"ADMIN\".\"STORAGE_LINK\".\"AA\"",
      "data": {
        "name": "AA",
        "entityID": 28897,
        "namespace": "STORAGE_LINK",
        "path": "\"STORAGE_LINK\".\"AA\"",
        "schema": "ADMIN",
        "application": "CLOUD",
        "created": "2023-01-30T21:42:17Z",
        "updated": "2023-01-30T21:42:17Z",
        "catalog": "LOCAL"
      }
    }
  ]
}
```

```
    }  
  },  
  ...  
  ]  
}
```

Create Cloud Storage Link Procedure

Create Cloud Storage Link Procedure based on Cloud Storage URI and credentials. Credential may be skipped if storage link is a public bucket.

Syntax

```
Ingest.create_cloud_storage_link(storage_link_name, uri, credential_name,  
                                description)
```

Parameters:

- **storage_link_name**: The name for the cloud storage link. This field is mandatory.
- **uri**: The URL Path (URI) except for the file name. This field is mandatory.
- **credential_name**: The credential name must conform to Oracle object naming conventions.
- **description**: a description for the link. If description is missing, the tool uses **storage_link_name** instead.

Example

In this example, you can create the Cloud Storage Link with the name 'TEST'

```
Ingest.create_cloud_storage_link('TEST', 'https://.../test-bucket/o/', None,  
'OCI Storage Link')
```

Drop Cloud Storage Link Procedure

Drop Cloud Storage Link based on storage link name.

```
Ingest.drop_cloud_storage_link(storage_link_name)
```

Parameters:

- **storage_link_name**: The name for the cloud storage link.

Example

In this example, you can drop Cloud Storage Link with the name 'TEST'

```
Ingest.drop_cloud_storage_link('TEST')
```

Get Cloud Storage Objects Procedure

Get the list of objects in Cloud Storage Link.

Syntax

```
Ingest.get_cloud_objects(storage_link,owner)
```

This function returns the list of objects in cloud storage link.

Parameters:

- **storage_link:** The name for the cloud storage link.
- **owner:** This field displays the owner of the Database Link. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can list of objects in Cloud Storage Link with the name 'TEST'

```
Ingest.get_cloud_objects('TEST')

// Output
{
  "nodes": [
    {
      "label": "users/1623813236395.parquet",
      "type": "CLOUD_OBJECT",
      "id": "\"ADMIN\".\"STORAGE_LINK\".\"TEST\".\"OBJECT\".\"users/1623813236395.parquet\"",
      "data": {
        "name": "users/1623813236395.parquet",
        "namespace": "OBJECT",
        "path": "\"STORAGE_LINK\".\"TEST\".\"OBJECT\".\"users/1623813236395.parquet\"",
        "schema": "ADMIN",
        "annotation": {
          "bytes": 1835742,
          "checksum": "8c215516638037427850b03f0e111850",
          "isFolder": false,
          "fileName": "1623813236395.parquet",
          "uri": "https://.../test-bucket/o/users/1623813236395.parquet"
        },
        "application": "CLOUD",
        "created": "2022-08-02T14:21:26Z",
        "updated": "2022-08-02T14:21:26Z",
        "catalog": "LOCAL"
      }
    },
    ...
  ]
}
```

This returns the list of available consumer groups.

Copy Cloud Objects Procedure

Copy cloud objects from the Cloud Storage Link to the tables in the current schema

```
Ingest.copy_cloud_objects(objects, consumer_group)
```

Parameters:

- **objects:** Array of tables description. Each description has 4 fields.
 - **storageLink:** The name of the Cloud Storage Link. This field is mandatory.
 - **objectName:** the name of the file in the Cloud Storage Link. See **name** field in results of **get_cloud_object** method. This field is mandatory.
 - **targetTableName:** The name of the target table. If this field is missing, the tool uses the filename as the table name.
- **consumer_group:** the consumer group. Default value is "LOW".

Example

In this example, you can create table and load data from the object in Cloud Storage Link.

```
adp.Ingest.copy_cloud_objects([{'storageLink': 'TEST', 'objectName': 'users/
testData.csv', 'targetTableName': 'TESTDATA'}, 'HIGH'])
```

```
output:
[
  {
    "storageLink": "TEST",
    "targetTableName": "TESTDATA",
    "objectName": "users/testData.csv",
    "rowsCopied": 588
  }
]
```

Link Cloud Objects Procedure

Create external tables based on cloud objects from the Cloud Storage Link.

```
Ingest.link_cloud_objects(objects, consumer_group)
```

Parameters:

- **objects:** Array of tables description. Each description has 4 fields.
 - **storageLink:** The name of the Cloud Storage Link. This field is mandatory.
 - **objectName:** the name of the file in the Cloud Storage Link. See **name** field in results of **get_cloud_object** method. This field is mandatory.
 - **targetTableName:** The name of the target table. If this field is missing, the tool uses the filename as the table name.
- **consumer_group:** the consumer group. Default value is "LOW".

Example

In this example, you can create external table from specified object in Cloud Storage Link.

```
adp.Ingest.link_cloud_objects([{'storageLink': 'TEST', 'objectName': 'users/
testData.csv', 'targetTableName': 'TESTDATA'}, 'HIGH'])
```

```
output:
[
  {
```

```

        "storageLink": "TEST",
        "targetTableName": "TESTDATA",
        "objectName": "users/testData.csv",
        "rowsCopied": 588
    }
]

```

Create Table from json Data Procedure

```
Ingest.load_data(tables)
```

Parameters:

- **tables** is the list of dictionaries with the following fields:
 - **content**: Is the content of data, it is a dictionary with key as column name and values as a list of values of the column.
 - **targetTableName**: The name of table.

Example

In this example, you can create table and fill it with data from content:

```

content = {
    "Year_id": [11,12,13,14,15],
    "Year_name": ["CY2011", "CY2012", "CY2013", "CY2014", "CY2015"],
    "Year_end_date": ["31-DEC-11", "31-DEC-12", "31-DEC-13", "31-DEC-14", "31-DEC-15"],
    "Quarter_id": [211,212,213,214,215],
    "Quarter_name": ["Q2CY2011", "Q2CY2012", "Q2CY2013", "Q2CY2014", "Q2CY2015"],
    "Quarter_end_date": ["30-JUN-11", "30-JUN-12", "30-JUN-13", "30-JUN-14", "30-JUN-15"]
}

content_list=[{"content": content, "targetTableName": "TestLoad"}]

adp.Ingest.load_data(content_list)

//Output

[{'fileName': 'TESTLOAD', 'targetTableName': 'TESTLOAD', 'rowsCopied': 6}]

```

Analytic View Functions

The class **Adp.Analytics** provides the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to Analytic Views tool of Data Studio Suite of tools. If the requested analytic view does not exist (except create), all functions return message `Analytic view does not exist`.

List of Analytic View Names Procedure

Returns list of Analytic View names.

```
Analytics.get_list(owner)
```

Parameters:

- **owner:** This field displays the owner of the Database Link. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can the list of Analytic Views:

```
Analytics.get_list()
```

```
{
  "nodes": [
    {
      "label": "ANALYTIC_VIEW1",
      "type": "ANALYTIC_VIEW",
      "id": "\"ADMIN\".\"DB\".\"ANALYTIC_VIEW1\"",
      "data": {
        "name": "ANALYTIC_VIEW1",
        "namespace": "DB",
        "path": "\"DB\".\"ANALYTIC_VIEW1\"",
        "schema": "ADMIN",
        "classifications": {
          "CAPTION": "Analytic View1",
          "DESCRIPTION": "Analytic View1"
        },
        "application": "DATABASE",
        "created": "2021-06-09T14:19:06Z",
        "updated": "2021-06-15T10:23:07Z"
      }
    }, ...
  ]
}
```

Create the Analytic View Procedure

Create the Analytic View based on list of dimension tables and list of measures.

Syntax

```
Analytics.create(fact_table, dimensions, measures, av_name,
                owner)
```

Parameters:

- **fact_table:** The name of fact table.
- **dimensions** is the list of dimensions table names. If dimension table has no joins with any keys in fact table, throws the error.
- **measures:** This field specifies the list of the fact table columns that will be used for aggregation.

- **av_name:** This field is an Analytic View name, if this parameter is missing, use default analytic view name based on name of the fact table: "<fact_table>_AV".
- **owner:** This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can create the Analytic View based on four dimensions and two measures:

```
tables = ["CHANNELS", "PRODUCTS", "PROMOTIONS", "TIMES"]
measures = [ 'UNIT_COST', 'UNIT_PRICE' ]
adp.Analytics.create('COSTS', tables, measures)
```

```
// Output
```

```
success
```

Create the Analytic View Automatically Procedure

Create the Analytic View based on fact table. Measures are selected automatically based on the columns of the fact table. Dimensions are selected from suitable dimensions tables.

The function uses default analytic view name based on name of the fact table: "<fact_table>_AV".

Syntax

```
Analytics.create_auto(fact_table, skip_dimensions,
                      owner)
```

Parameters:

- **fact_table:** The name of fact table.
- **skip_dimensions:** If the field value is True, columns of the fact table are selected as dimensions. The default value is False.
- **owner:** This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can create the Analytic View:

```
Analytics.create_auto('COSTS', False) // Output success
```

Delete the Analytic view Procedure

Delete the Analytic view with specified name

Syntax

```
Analytics.drop(model_name, delete_objects)
```

Parameters:

- **model_name**: The name of the Analytic View.
- **delete_objects**: If the field value is True, all related objects (dimensions and measures) will be dropped. The default value is False.

Example

In this example, you can drop the Analytic View:

```
adp.Analytics.drop('COSTS_AV', True)
```

Compile the Analytic View Procedure

Compile the Analytic View. It optimizes the query to efficiently retrieve data from the fact tables.

Syntax

```
Analytics.compile(av_name, owner)
```

Parameters:

- **av_name**: The name of the Analytic View.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can compile the Analytic View:

```
adp.Analytics.compile('COSTS_AV')
```

Get Measures List Procedure

Get the list of measures from the specified Analytic View.

Syntax

```
Analytics.get_measures_list(av_name, owner)
```

Parameters:

- **av_name**: The name of the Analytic View.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get measure list of the Analytic View:

```
adp.Analytics.get_measures_list('COSTS_AV')
```

```
// Output
```

```
{
  "nodes": [
    {
      "label": "UNIT_COST",
```

```

    "type": "MEASURE",
    "id": "\"ADMIN\".\"DB\".\"COSTS_AV\".\"MEASURE\".\"UNIT_COST\"",
    "data": {
      "name": "UNIT_COST",
      "namespace": "MEASURE",
      "path": "\"DB\".\"COSTS_AV\".\"MEASURE\".\"UNIT_COST\"",
      "schema": "ADMIN",
      "application": "DATABASE",
      "created": "2022-08-16T17:04:08Z",
      "updated": "2022-08-16T17:04:08Z"
    }
  }, ...
]
}

```

Get Preview Procedure

Return the metadata of the Analytic View (hierarchies (name and caption) and measures), aggregated data, and sql query for selecting data from the Analytic View.

Syntax

```
Analytics.get_data_preview(entity_name, owner)
```

Parameters:

- **entity_name**: The name of the Analytic View.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get preview data of the Analytic View:

```
adp.Analytics.get_data_preview('COSTS_AV')
```

```
// Output
```

```

{
  "av": "COSTS_AV",
  "caption": null,
  "description": null,
  "metadata": {
    "hierarchies": [
      {
        "name": "CHANNEL_ID",
        "caption": "CHANNEL_ID"
      },
      {
        "name": "PROD_ID",
        "caption": "PROD_ID"
      }
    ],
    "measures": [
      {
        "name": "UNIT_COST",

```

```
        "caption": "UNIT_COST"
    },
    {
        "name": "UNIT_PRICE",
        "caption": "UNIT_PRICE"
    }
]
},
"data": [...],
"SQL": "..."
}
```

Get Metadata Procedure

Return detailed metadata of the Analytic View.

Syntax

```
Analytics.get_metadata(av_name, owner)
```

Parameters:

- **av_name**: The name of the Analytic View.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Return detailed metadata of the Analytic View in the schema of owner (if is missing or set to None, use the current schema).

Example

In this example, you can get metadata of the Analytic View:

```
adp.Analytics.get_metadata('COSTS_AV')
```

Dimension Names Procedure

Returns array of dimension names of the Analytic View

Syntax

```
Analytics.get_dimension_names(av_name)
```

Parameters:

- **av_name**: The name of the Analytic View.

Example

In this example, you can get dimension names of the Analytic View:

```
adp.Analytics.get_dimension_names('COSTS_AV')
```

Fact Table Name Procedure

Returns the name of the fact table of the Analytic View.

Syntax

```
Analytics.get_fact_table_name(av_name)
```

Parameters:

- **av_name**: the name of the Analytic View.

Example

In this example, you can get name of the fact table of the Analytic View:

```
adp.Analytics.get_fact_table_name('COSTS_AV')
```

Get Error Classes From Dimensions Procedure

Return the information about errors in specified dimension during analyzing the Analytic View. This method is used for checking that the dimension of the Analytic View has no errors. In the case of error `ERROR_COUNT` is not 0, and `errorData` is not empty, and contains `ERROR_MESSAGE` (Text representation of the error), `ERROR_NUMBER`(Oracle error code).

Syntax

```
Analytics.get_error_classes_from_dim(av_name,  
                                     dimension)
```

Parameters:

- **av_name**: the name of the Analytic View.
- **dimension**: dimension name.

Example

In this example, you can get error classes name of the dimensions of the Analytic View:

```
adp.Analytics.get_error_classes_from_dim('COSTS_AV', 'CHANNELS')
```

Get Error Classes From Fact Table Procedure

Return the information about errors in specified fact table during analyzing the Analytic View. This method is used for checking that the fact table of the Analytic View has no errors. In the case of error `ERROR_COUNT` is not 0, and `errorData` is not empty, and contains `ERROR_MESSAGE` (Text representation of the error) and `ERROR_NUMBER` (Oracle error code).

Syntax

```
Analytics.get_error_classes_from_fact_tab(av_name,  
                                          fact_tab)
```

Parameters:

- **av_name**: The name of the Analytic View.
- **fact_table**: The name of fact table.

Example

In this example, you can get error classes of the name of the fact table of the Analytic View:

```
adp.Analytics.get_error_classes_from_fact_tab('COSTS_AV', 'COSTS')
```

Return the information about errors in specified fact table during analyzing the Analytic View. This method is used for checking that the fact table of the Analytic View has no errors. In the case of error `ERROR_COUNT` is not 0, and `errorData` is not empty, and contains `ERROR_MESSAGE` (Text representation of the error) and `ERROR_NUMBER` (Oracle error code).

Quality Report Procedure

Return the information about errors in analytic view specified by `av_name`. This method checks fact table and dimensions of the Analytic View. In the case of error text representation of the errors is included into report.

Syntax

```
Analytics.quality_report(av_name)
```

Parameters:

- **av_name:** The name of the Analytic View.

Example

In this example, you can get quality report of the Analytic View:

```
adp.Analytics.quality_report('COSTS_AV')

// Output

[
  "Fact table COSTS has no errors",
  "Dimension COSTS_AV_CHANNELS_AD has no errors",
  "Dimension COSTS_AV_PRODUCTS_AD has no errors",
  "Dimension COSTS_AV_PROMOTIONS_AD has no errors",
  "Dimension COSTS_AV_TIMES_AD has no errors"
]
```

Get Data Procedure

Returns the aggregated data from the Analytic View. The return value is the list of query results.

Syntax

```
Analytics.get_data(levels, column_names, entity_name, hierarchies,
                  measures, where_condition, owner)
```

Parameters:

- **levels:** The level of the Analytic View.
- **column_names:** The array of columns.
- **entity_name:** The name of the Analytic View.

- **hierarchies**: The array of selected hierarchies.
- **measures**: The array of measures.
- **where_condition**: is the array of conditions, each condition has 4 fields: **hierarchy**, **column**, **operator**, and **value**.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get quality report of the Analytic View:

```
adp.Analytics.get_data(True,
    ["COSTS_AV_PROD_HIER_PROD_ID_ATTR",
     "UNIT_COST",
     "COSTS_AV_PROD_HIER_DEPTH"], 'COSTS_AV',
    ["COSTS_AV_PROD_ID_HIER"],
    ["UNIT_COST", "UNIT_PRICE"],
    [{"hierarchy": "COSTS_AV_PROD_ID_HIER", "column": "LEVEL_NAME",
     "operator": "=", "value": "ALL:PROD_ID"}])

//Output

[{"UNIT_COST":808685.68}, {"UNIT_COST":213170.06}, {"UNIT_COST":16778.02}, ...],
```

SQL Query Procedure

Return SQL Query that is used in `get_data`.

Syntax

```
Analytics.get_sql(levels, column_names, entity_name, hierarchies,
    measures, where_condition, owner)
```

Parameters:

- **levels**: The level of the Analytic View.
- **column_names**: The array of columns.
- **entity_name**: The name of the Analytic View.
- **hierarchies**: The array of selected hierarchies.
- **measures**: The array of measures.
- **where_condition**: is the array of conditions, each condition has 4 fields: **hierarchy**, **column**, **operator**, and **value**.
- **owner**: This field displays the owner of the Analytic View. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get SQL for aggregated data of the Analytic View:

```
adp.Analytics.get_sql(True,
    ["COSTS_AV_PROD_HIER_PROD_ID_ATTR",
     "UNIT_COST",
     "COSTS_AV_PROD_HIER_DEPTH"], 'COSTS_AV',
```

```

        ["COSTS_AV_PROD_ID_HIER"],
        ["UNIT_COST", "UNIT_PRICE"],
        [{"hierarchy": "COSTS_AV_PROD_ID_HIER", "column": "LEVEL_NAME",
"operator": "=", "value": "ALL:PROD_ID"}]
    )
}

//Output

SELECT
    "MEASURES"."UNIT_COST" AS "UNIT_COST"
FROM "ADMIN"."COSTS_AV" HIERARCHIES(
    "PROD_ID"."PROD_ID" )
WHERE
    (
        (
            "PROD_ID"."PROD_ID"."LEVEL_NAME" IN ('ALL', 'PROD_ID')
        )
    )
)

```

Insight Functions

Insight functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to the Insight tool in Data Studio suite of tools. Generation of insight is performed in background and requires checking for the completion.

Get Request List Procedure

Returns the list of request names and their parameters

Syntax

```
Insight.get_request_list(owner)
```

Parameters:

- **owner:** This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can list jobs of insights:

```
adp.Insight.get_request_list()
```

```
// Output
```

```

[
  ...
  {
    "owner": "ADMIN",
    "request_name": "REQUEST_INSIGHT_1",
    "request_type": "TABLE_INSIGHT",
    "source_object": "SALES",
    "source_owner": "ADMIN",
    "request_metadata": "{ \"targets\": [ \"ACTUAL\" ], \"appName\": \"INSIGHTS\" }",
    "created": "2022-10-31T08:03:26.634Z",
  }
]

```

```

    "updated": "2022-10-31T08:03:26.634Z",
    "request_job_setting": "{\\"EXTRACTION\\":{\\"selectionRule\\":
{\\"value\\":\\"MAX_DIFFERENCE\\"},\\"fittedShareConvergence\\":
{\\"value\\":\\"LAX\\"},\\"maxNumExtractionDims\\":
{\\"value\\":10000},\\"insightValueType\\":{\\"value\\":\\"MEMBER\\"},\\"excludeZero\\":
{\\"value\\":\\"YES\\"},\\"minXAxisCount\\":
{\\"value\\":2}},\\"sessionId\\":null,\\"applicationName\\":\\"INSIGHTS\\",\\"insightTy
pes\\":[\\"FITTED_SHARE_COMPARISON\\"]}",
    "max_num_insights": 20
  },
  ...
]

```

Get Insight List Procedure

Returns the list of insights. Each insight has the following fields: **insight_name**, **visualization_id**, **insight_column**, **insight_value**, **insight_dimension**, **dimension**

Syntax

```
Insight.get_insights_list(request_name)
```

Parameters:

- **request_name**: The name of the request.

Example

In this example, you can list insights:

```
adp.Insight.get_insights_list('REQUEST_INSIGHT_1')
```

```

// Output:
[
  {
    "insight_name": "INSIGHT_62",
    "visualization_id": 199,
    "insight_column": "ACTUAL",
    "insight_value": "COUNTRY.2",
    "insight_dimension": "COUNTRY",
    "dimension": "FORECST"
  },...
]

```

Drop Insight Procedure

Drop insight request

Syntax

```
Insight.drop(request_name)
```

Parameters:

- **request_name**: The name of the request.

Example

In this example, you can drop the request:

```
adp.Insight.drop('REQUEST_INSIGHT_1')
```

Generate Insight Procedure

Generate insight of the measure for the Analytic view.

Syntax

```
Insight.generate(av_name, measure, job_owner,  
                 source_owner)
```

Parameters:

- **av_name**: The name of the Analytic View.
- **measure**: The measure.
- **job_owner**: This field displays the owner of the job. If this field is missing, the tool uses the current schema owner.
- **source_owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can generate insights for the Analytic View:

```
adp.Insight.generate('COSTS_AV', 'UNIT_COST')
```

```
// Output
```

```
{  
  "object_owner": "ADMIN",  
  "request_name": "REQUEST_INSIGHT_1",  
  "max_insight_count": 20,  
  "source_object": "COSTS_AV",  
  "source_owner": "COSTS_AV",  
  "request_metadata": "{ \"targets\":  
[\"UNIT_COST\"], \"appName\": \"INSIGHTS\" }",  
  "default_job_settings": "{ \"insightTypes\": [\"FITTED_SHARE_COMPARISON\"] }",  
  "new_request": "true"  
}
```

Get Job Status

Get the status of the job specified by Request Name. The generation of insights is finished when field **insight_job_status** is COMPLETED.

Syntax

```
Insight.get_job_status(request_name, owner)
```

Parameters:

- **request_name**: The name of Request.
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can display job status:

```

adp.Insight.get_job_status('REQUEST_INSIGHT_1')

// Output
{
  "insight_job_status": "COMPLETED",
  "progress_msg": " ",
  "error_msg": null
}

```

Get Graph Details Procedure

Returns json dictionary of data that may be plotted.

Results consists of two fields:

item contains overall information from the insight: **description**, **XAXIS**, **measure**, **insight_type_label**, **visualization_condition**, **source_object**

query contains information for plotting and has three lists: **labels**, **actuals** and **estimates**

Syntax

```

Insight.get_graph_details(name, id, count, query_manipulation,
                          owner)

```

Parameters:

- **name**: The name of the Insight.
- **id**: The visualization id.
- **count**: number of displayed values
- **query_manipulation**: If it is not None, add cursor type and grand totals fields to the query.
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get data to plot the graph:

```

adp.Insight.get_graph_details("INSIGHT_62", 199, 5, True)

```

```

Output:
{
  "items": {
    "description": "COUNTRY.2",
    "XAXIS": "FORECST",
    "measure": "ACTUAL",
    "insight_type_label": "Expected",
    "visualization_condition": "COUNTRY.2",
    "source_object": "CO"
  },
  "query": {
    "labels": [

```

```

        "01:[66.7-1590]",
        "02:[829-2350]",
        "04:[2350-3870]",
        "06:[3870-5390]",
        "07:[4630-6150]"
    ],
    "actuals": [
        2200,
        6200,
        24600,
        26600,
        14200
    ],
    "estimates": [
        2671.105013588167,
        4830.820533674098,
        7135.506976989193,
        24772.963909661587,
        22121.48462234561
    ]
}
}
}

```

Miscellaneous functions

Miscellaneous functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface.

Miscellaneous functions provide the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface.

List of Tables Procedure

Return list of tables.

Syntax

```
Misc.list_tables(owner)
```

Parameters:

- **owner:** This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get list of tables:

```
adp.Misc.list_tables()
```

```
// Output
```

```
{
  "nodes": [
    {
      "label": "COSTS",
      "type": "TABLE",

```

```

    "id": "\"ADMIN\".\"DB\".\"COSTS\"",
    "data": {
      "name": "COSTS",
      "namespace": "DB",
      "path": "\"DB\".\"COSTS\"",
      "schema": "ADMIN",
      "annotation": {
        "numRows": 82112,
        "status": "VALID",
        "blocks": 370,
        "avgRowLen": 26,
        "partitioned": "NO",
        "external": "NO",
        "sharded": "N",
        "tablespace": "DATA",
        "hasSensitiveColumn": "NO",
        "compression": "DISABLED"
      },
      "application": "DATABASE",
      "created": "2021-06-09T09:16:49Z",
      "updated": "2021-06-09T09:16:49Z",
      "catalog": "LOCAL"
    }
  }, ...
]
}

```

List of Views Procedure

Return list of views.

Syntax

```
Misc.list_views(owner)
```

Parameters:

- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can get the list of views:

```

adp.Misc.list_views()

// Output
{
  "nodes": [
    {
      "label": "COSTS_AVVIEW",
      "type": "VIEW",
      "id": "\"ADMIN\".\"DB\".\"COSTS_AVVIEW\"",
      "data": {
        "name": "COSTS_AVVIEW",
        "namespace": "DB",
        "path": "\"DB\".\"COSTS_AVVIEW\"",

```

```
        "schema": "ADMIN",
        "application": "DATABASE",
        "created": "2024-11-21T10:00:16Z",
        "updated": "2024-11-21T10:00:16Z",
        "catalog": "LOCAL"
    }, ...
]
}
```

Drop Table Procedure

Drop the specified table.

Syntax

```
Misc.drop_table(table_name)
```

Parameters:

- **table_name**: This field displays the name of the table to drop.

Example

In this example, you can drop table 'TEST':

```
adp.Misc.drop_table('TEST')
```

Drop View Procedure

Drop the specified view.

Syntax

```
Misc.drop_view(view_name)
```

Parameters:

- **view_name**: This field displays the name of the view to drop.

Example

In this example, you can drop view 'VIEW_TEST':

```
adp.Misc.drop_view('VIEW_TEST')
```

Drop Tables Procedure

Drop the specified tables

Syntax

```
Misc.drop_tables(table_names)
```

Parameters:

- **table_names**: This field displays the array of names of the table to drop.

Example

In this example, you can drop table 'TEST':

```
dp.Misc.drop_tables(['TEST'])
```

Drop Views Procedure

Drop the specified views.

Syntax

```
Misc.drop_views(view_names)
```

Parameters:

- **view_names:** This field displays the array of names of the views to drop.

Example

In this example, you can drop views 'VIEW_TEST':

Get Entity DDL Procedure

Generate DDL for creating or replacing entity (TABLE or VIEW)

Syntax

```
Misc.get_entity_ddl(entity_type, entity_name,  
                    owner)
```

Parameters:

- **entity_type:** This field displays the type of the entity. It can be 'TABLE' or 'VIEW'
- **entity_name:** This field displays the name of the entity.
- **owner:** This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can view DDL of table 'TEST':

```
adp.Misc.get_entity_ddl('TABLE', 'TEST')
```

Run Query Procedure

Return the results of query.

Syntax

```
Misc.run_query(statement, offset, limit, asof)
```

Parameters:

- **statement:** This field displays SQL statement to run
- **offset:** This field displays the offset to displayed record. If this field is missing, the tool uses 0 as offset.
- **limit:** This field displays maximum number of records to display. If this field is missing, the tool displays all records.

- **asof**: AS OF statement to display historic data. If this field is missing, the tool does not use this feature.

Example

In this example, you can drop the view:

```
adp.Misc.run_query('DROP VIEW "VIEW_TEST"')
```

Global Search Procedure

Return the result of searching databases, tables and another artifacts from the Data Studio.

Syntax

```
Misc.global_search(search_string, rowstart, numrow, sortBy,  
                  hide_system_tables, hide_private_tables, resultapp,  
                  resultannotation)
```

Parameters:

- **search_string**: string to search. It should be in the form "(owner: ADPTEST) (type: TABLE) (application: DATABASE)"
- **rowstart**: first index of searching
- **numrow**: number of returned rows
- **sortBy**: list of sort, e.g. [{"column": "entity_name", "direction": "asc"}]
- **hideSystemTables**: If this field is missing, system tables should be searched too.
- **hidePrivateTables**: If this field is missing, system tables should be searched too.
- **resultapp**:
- **resultannotation**:

Example

In this example, you can drop the view:

```
adp.Misc.global_search('( owner: ADMIN ) ( type: TABLE ) ( application:  
DATABASE )', 1, 20001, [], TRUE, True, resultapp="ADPINS")
```

Get Table Columns Procedure

This function returns the list of the columns information of the table

Syntax

```
Misc.get_table_columns(table_name, owner, limit,  
                      offset)
```

Parameters:

- **table_name**: This field displays the name of the table.
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.
- **limit**: The maximum number of columns to display. The default value is 256.

- **offset**: the offset of the columns. The default value is 0.

Example

In this example, you can view columns of table 'PERSONS':

```
adp.Misc.get_table_columns('PERSONS')
```

Output:

```
[
  {
    "column_name": "PERSON_ID",
    "data_type": "NUMBER",
    "nullable": "No",
    "data_default": "\"ADMIN\".\"ISEQ$$_329201\".nextval",
    "column_id": 1,
    "comments": null
  },
  {
    "column_name": "FIRST_NAME",
    "data_type": "VARCHAR2(50 BYTE)",
    "nullable": "No",
    "data_default": null,
    "column_id": 2,
    "comments": null
  },
  {
    "column_name": "LAST_NAME",
    "data_type": "VARCHAR2(50 BYTE)",
    "nullable": "No",
    "data_default": null,
    "column_id": 3,
    "comments": null
  }
]
```

Get Table Constraints Procedure

This function returns the list of the constraints of the table.

Syntax

```
Misc.get_table_constraints(table_name, owner, limit,
                           offset)
```

Parameters:

- **table_name**: This field displays the name of the table.
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.
- **limit**: The maximum number of columns to display. The default value is 256.
- **offset**: the offset of the columns. The default value is 0.

Example

In this example, you can view constraints of table 'PERSONS':

```
adp.Misc.get_table_constraints('PERSONS')
```

Output:

```
[
  {
    "constraint_name": "SYS_C0056209",
    "constraint_type": "Check",
    "search_condition": "\"PERSON_ID\" IS NOT NULL",
    ...
  }
]
```

Insert Row Procedure

The function inserts row into the table

Syntax

```
Misc.insert_row(table_name, data, mapping, owner)
```

Parameters:

- **table_name**: This field displays the name of the table.
- **data**: values of the row to insert into table as a form {"data_column" : value,...}.
- **mapping**: Map data column names in **data** with column names of the table. If this field is missing, the tool uses column names of the table
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can view insert a row to the table 'PERSONS', without mapping and using mapping:

```
adp.Misc.insert_row('PERSONS', {"PERSON_ID":1, "FIRST_NAME":"John",
"LAST_NAME":"Smith"})
```

```
adp.Misc.insert_row('PERSONS', {"ID":1, "FIRST":"John", "LAST":"Smith"},
{"ID":"PERSON_ID", "FIRST":"FIRST_NAME", "LAST":"LAST_NAME"})
```

Update Row Procedure

The function update existing row in the table.

Syntax

```
Misc.update_row(table_name, data, where_col, mapping,
                owner)
```

Parameters:

- **table_name**: This field displays the name of the table.
- **data**: values of the row to insert into table as a form {"data_column" : value,...}.
- **where_col**: name of the column that will be used in where clause. The value of this column should be in **data**
- **mapping**: Map data column names in **data** with column names of the table. If this field is missing, the tool uses column names of the table
- **owner**: This field displays the owner of the object. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can view insert a row to the table 'PERSONS', without mapping and using mapping:

```
adp.Misc.update_row('PERSONS', {"PERSON_ID":1, "FIRST_NAME":"Adam"},
'PERSON_ID')
```

```
adp.Misc.update_row('PERSONS', {"ID":1, "FIRST":"Adam"}, 'ID',
{"ID":"PERSON_ID", "FIRST":"FIRST_NAME"})
```

Data Share Functions

The Python Data Studio Application Programming Interface (API) for Data Share functions provides the syntax and descriptions of the classes, methods, attributes, and parameters of the application programming interface to the Data Share tool of Data Studio suite of tools.

Data Studio Data Share API functions

The `Adp.Share` class is used for working with data shares. This class contains a set of data sharing functions that allows manage data shares, recipients etc.

Prerequisites

You must first create an ADP instance.

The functions have string return value and string parameters.

Create Share Procedure

This function creates Data Share based on object storage date and associate them with recipients.

Syntax

```
Share.create_share(name, objects, storage_link_name, publish_job_details,
owner, type,
storage_link_owner, description)
```

The following are it's parameters and descriptions:

- **name**: The name of the Data Share. This field is mandatory.
- **objects**: The Data Share objects dictionary consists of the following fields:

- [{"schemaName" : "schemaNameValue", "objects": [{"objectName": "objectNameValue", "objectOwner": "objectOwnerValue", "shareObjectName" : "shareObjectNameValue"}, ...], ...}]

This field is mandatory.

- **storage_link_name:** This field displays the Data Share cloud storage link name.
- **publish_job_details:** This field displays the Data Share publish job details dictionary. If this field is missing {"enabled" : False} is used.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.
- **type:** The type of the Data Share. If this field is missing, the tool uses the `VERSIONED` field.
- **storage_link_owner:** The Data Share Cloud Storage link owner. If this field is missing, the tool uses the current schema owner.
- **description:** This field displays the data share description. If this field is missing, the tool uses an empty string.

Example

In this example, you can create a Data Share:

```
name = 'share_py_api_7'
desc = 'desc_share_py_api_7'
link = '"food_cred"'
share_objects = json.dumps([{"schemaName" : "ADMIN",
                             "objects" : [{"objectName"      : "IRIS",
                                           "objectOwner"     : "ADMIN",
                                           "shareObjectName" : "IRIS_SHARE" },
                                           {"objectName"      : "DEMO_FOOD_SAMPLE",
                                           "objectOwner"     : "ADMIN",
                                           "shareObjectName" : "DEMO_FOOD_SAMPLE_SHARE"}
                                           ]}]])
job = json.dumps({"enabled" : False})

adp.Share.create_share(
    name           = name,
    description    = desc,
    storage_link_name = link,
    objects        = share_objects,
    publish_job_details = job)

//Output:
{"createdShare":true,"notifyRecipients":false,"shareId":40295,"sharetablename_
":"DEMO_FOOD_SAMPLE_SHARE" }
```

Update Data Share Objects Procedure

This function updates the Data Share objects.

Syntax

```
Share.update_share_objects(name, objects, publish_job_details, owner,
                           description)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **objects:** The Data Share objects dictionary consists of the following fields:
 - [{"schemaName": "schemaNameValue", "objects": [{"objectName": "objectNameValue", "objectOwner": "objectOwnerValue", "shareObjectName": "shareObjectNameValue"}, ...]}, ...]

This field is mandatory.
- **publish_job_details:** This field displays the Data Share publish job details dictionary. If this field is missing {"enabled": False} is used.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.
- **description:** This field displays the data share description. If this field is missing, the tool uses an empty string.

Example

In this example, you can update a Data Share object:

```
name = 'share_py_api_7'
objects = json.dumps([{"schemaName" : "ADMIN",
                      "objects" : [{"objectName"      : "WINE",
                                   "objectOwner"     : "ADMIN",
                                   "shareObjectName"  : "WINE_SHARE"}
                                ]}])

adp.Share.update_share_objects(name = name, objects = objects)
```

```
//Output:
'{"sharetablename_": "WINE_SHARE"}'
```

Delete Data Share Procedure

This function deletes the Data Share.

Syntax

```
Share.delete_share(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can delete a Data Share object:

```
adp.Share.delete_share('share_py_api_7_1')
```

```
//Output:  
'{"deletedShare":true}'
```

Rename Data Share Procedure

This function renames the Data Share.

Syntax

```
Share.rename_share(name, new_name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **new_name:** The new name you want for the Data Share. This field is mandatory.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can rename a Data Share:

```
adp.Share.rename_share (name = 'share_py_api_7', new_name =  
'share_py_api_7_1')
```

```
'{ "status": true }'
```

Publish Data Share Procedure

This function publishes Data Share.

Syntax

```
Share.publish_share(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can publish a Data Share:

```
adp.Share.publish_share(name='share_py_api_7_1')
```

```
//Output:  
'{"publishShare":true}'
```

Unpublish Data Share

This function publishes Data Share.

Syntax

```
Share.unpublish_share(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can unpublish a Data Share:

```
adp.Share.unpublish_share(name = 'share_py_api_7_1')
```

```
//Output:  
'{"unpublishShare":true}'
```

Get Share Procedure

This function receives Data Share details.

Syntax

```
Share.get_share(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Data Share. This field is mandatory.
- **owner:** This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive Data Share details:

```
s = adp.Share.get_share(name = 'share_py_api_7_1')  
json.loads(s)
```

```
//Output:
{'id': 40415,
 'name': 'SHARE_PY_API_7_1',
 'owner': 'ADMIN',
 'description': 'desc_share_py_api_7',
 'publicDescription': 'desc_share_py_api_7',
 'type': 'VERSIONED',
 'version': None,
 'tablesCount': 3,
 'recipientsCount': 0,
 'lastExportTime': None,
 'objects': [{ 'id': 40416,
  'schemaName': 'ADMIN',
  'objects': [{ 'id': 40435,
   'shareObjectName': 'WINE_SHARE',
   'objectName': 'WINE',
   'objectOwner': 'ADMIN',
   'objectType': 'TABLE'}],
 'id': 40417,
  'shareObjectName': 'IRIS_SHARE',
  'objectName': 'IRIS',
  'objectOwner': 'ADMIN',
  'objectType': 'TABLE'}],
 'id': 40418,
  'shareObjectName': 'DEMO_FOOD_SAMPLE_SHARE',
  'objectName': 'DEMO_FOOD_SAMPLE',
  'objectOwner': 'ADMIN',
  'objectType': 'TABLE'}]}],
 'recipients': None,
 'storageLinkName': 'food_cred',
 'schedule': { 'enabled': 'FALSE',
  'startDate': None,
  'endDate': None,
  'repeatInterval': None}}
```

Get Shares Procedure

This function receives Data Share list based on the owner of the Data Share.

Syntax

```
Share.get_shares(owner)
```

The following are its parameters and descriptions:

- **owner**: This field displays the data share owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive the Data Share list:

```
s = adp.Share.get_shares()
json.loads(s)
```

```
//Output:
[{'id': 40415,
  'name': 'SHARE_PY_API_7_1',
  'owner': 'ADMIN',
  'description': 'desc_share_py_api_7',
  'type': 'VERSIONED',
  'tablesCount': 3,
  'version': None}]
```

You can also view the topics that covers the procedures listed below:

- [Recipient Data Share Functions](#)
- [Provider Data Share Functions](#)
- [Recipient Data Share Functions](#)
The procedures listed here covers all the functions required by a Recipient of a Data Share.
- [Provider Data Share Functions](#)
The procedures listed here covers all the functions required by the provider of a Data Share.

Recipient Data Share Functions

The procedures listed here covers all the functions required by a Recipient of a Data Share.

Create Recipient Procedure

This function creates a Data Share Recipient.

Syntax

```
Share.create_recipient(name, email, description, owner, shares,
                      token_lifetime)
```

The following are it's parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **email:** This field displays the mail address of the Recipient.
- **description:** This field displays the description of the Recipient. If this field is missing, the tool uses an empty string.
- **owner:** This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.
- **shares:** You can view the following fields in the recipient shares dictionary: "[{"name", "shareNameValue"},...]". If a parameter is missing, the tool uses an empty dictionary "[]".
- **token_lifetime:** The tool uses recipient token lifetime in this format 'D HH:MM:SS'. If this field is missing, the tool uses '0 00:60:00'.

Example

In this example, you can create a Data Share Recipient:

```
adp.Share.create_recipient(  
    name = 'test_py_api_recipient_1',  
    email = 'test@gmail.com'  
)
```

```
//Output:  
'{"createRecipient":true,"name":"TEST_PY_API_RECIPIENT_1"}'
```

Update Recipient Share Procedure

This function updates a Data Share Recipient.

Syntax

```
Share.update_recipient_shares(name, shares, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **owner:** This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.
- **shares:** You can view the following fields in the recipient shares dictionary: "[{"name","shareNameValue"},...]". If a parameter is missing, the tool uses an empty dictionary "[]".

Example

In this example, you can update a Data Share Recipient:

```
recipient_name = 'test_py_api_recipient_1'  
shares = json.dumps([{"name":'share_py_api_7'}])  
adp.Share.update_recipient_shares(name = recipient_name, shares = shares)
```

```
//Output:  
'{"status" : true}'
```

Delete Recipient Procedure

This function deletes a Data Share Recipient.

Syntax

```
Share.delete_recipient(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **owner:** This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can delete a Data Share Recipient:

```
adp.Share.delete_recipient (name = 'renamed_test_py_api_recipient_2')
```

```
//Output:  
'{ "status": true }'
```

Rename Recipient Procedure

This function renames a Data Share Recipient.

Syntax

```
Share.rename_recipient(name, new_name, owner)
```

The following are its parameters and descriptions:

- **name**: The name of the Recipient. This field is mandatory.
- **new_name**: The name of the Recipient you want. This field is mandatory.
- **owner**: This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can rename a Data Share Recipient:

```
adp.Share.rename_recipient(name = 'test_py_api_recipient_1', new_name =  
'test_py_api_recipient_2')
```

```
//Output:  
'{ "status": true }'
```

Get Recipient Procedure

This function receives details of the Data Share Recipient.

Syntax

```
Share.get_recipient(name, owner)
```

The following are its parameters and descriptions:

- **name**: The name of the Recipient. This field is mandatory.
- **owner**: This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive details of a Data Share Recipient:

```
s = adp.Share.get_recipient('test_py_api_recipient_2')
json.loads(s)
```

```
//Output:
{'name': 'TEST_PY_API_RECIPIENT_2',
 'id': 40436,
 'type': 'DELTA_SHARING',
 'description': None,
 'email': 'test@gmail.com',
 'clientId': 'xxxxxxxxxxxxx..',
 'clientSecret': 'xxxxxxxxxxxxx..',
 'tokenDuration': None,
 'shares': [{'name': 'SHARE_PY_API_7',
              'id': 40419,
              'description': 'desc_share_py_api_7',
              'tablesCount': 2}],
 'created': '2024-11-12T15:34:08',
 'updated': '2024-11-12T15:36:13'}
```

Get Recipients Procedure

This function receives a list of Data Share Recipients based on data schema recipients owner.

Syntax

```
Share.get_recipients(owner)
```

The following are it's parameters and descriptions:

- **owner:** This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive a list of Data Share Recipients:

```
s = adp.Share.get_recipients()
json.loads(s)
```

```
//Output:
[{'id': 40436,
  'name': 'TEST_PY_API_RECIPIENT_2',
  'type': 'DELTA_SHARING',
  'description': None,
  'email': 'test@gmail.com'}]
```

Get Recipient Sharing Profile Procedure

This function receives Recipient's Data Sharing profile.

Syntax

```
Share.get_recipient_sharing_profile(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **owner:** This field displays the Recipient owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive the Data Sharing profile of a Data Share Recipient:

```
s = adp.Share.get_recipient_sharing_profile('test_py_api_recipient_2')
json.loads(s)
```

```
//Output:
{'shareCredentialsVersion': 2,
 'type': 'persistent_oauth2.0',
 'endpoint': 'https://abc-xyz//xx/delta_sharing',
 'tokenEndpoint': 'https://abcdefg.xx.com/adp/admin/oauth/token',
 'clientId': 'abcdefghijklmnop..',
 'clientSecret': 'abcdefrtyuioohh..'}

```

Provider Data Share Functions

The procedures listed here covers all the functions required by the provider of a Data Share.

Create Provider Procedure

This function creates a Data Share Recipient.

Syntax

```
Share.create_provider(name, endpoint, share_links, provider_type, bearer,
client_id,
client_secret, token_endpoint, description, owner )
```

The following are its parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **endpoint:** This field displays the endpoint of the Provider Data Share service. It is mandatory.
- **share_links:** The provider share links dictionary has the following fields:
 - [{"shareName": "shareNameValue", "shareLinkName": "shareLinkNameValue"}]
This field is mandatory.
- **provider_type*:** This field displays the type of a provider data share type. You can have two values, DELTA or LIVE.
- **bearer:** This field displays the provider Data Share bearer token. This field is mandatory.

- **client_id**: This field displays the provider Data Share Client ID address. This field is mandatory.
- **client_secret**: This field displays the provider Data Share Client Secret token. This field is mandatory.
- **token_endpoint**: This field displays the provider Data Share Token endpoint address. This field is mandatory.
- **email**: This field displays the mail address of the Recipient.
- **description**: This field displays the description of the Provider. If this field is missing, the tool uses an empty string.
- **owner**: This field displays the Provider owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can create a Data Share Provider:

```
provider_json = {
    "shareCredentialsVersion": 1,
    "endpoint": "https://abcd/delta-sharing/",
    "bearerToken": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}

bearer_token = provider_json['bearerToken']
endpoint     = provider_json['endpoint']
provider_name = "provider_demo_delta_sharing_1"
description  = "provider demo delta sharing desc"
owner        = "ADMIN"
provider_type = "DELTA"
share_links  = json.dumps([{"shareName": "DELTA_SHARING",
                           "shareLinkName": "DEMO_DELTA_SHARING"}])

adp.Share.create_provider (
    bearer          = bearer_token,
    description     = description,
    endpoint        = endpoint,
    name            = provider_name,
    owner           = owner,
    provider_type   = provider_type,
    share_links     = share_links)

//Output:
'{"CreateShareProvider": true}'
```

Delete Provider Procedure

This function deletes a Data Share Provider.

Syntax

```
Share.delete_provider(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Recipient. This field is mandatory.
- **owner:** This field displays the Provider owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can delete a Data Share Provider:

```
adp.Share.delete_provider(name = 'provider_demo_delta_sharing_2')
```

```
//Output:  
'{ "status": true }'
```

Rename Provider Procedure

This function renames a Data Share Provider.

Syntax

```
Share.rename_provider(name, new_name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Provider. This field is mandatory.
- **new_name:** This field displays the name of the Data Share Provider of your choice. It is mandatory.
- **owner:** This field displays the Provider owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can rename a Data Share Provider:

```
adp.Share.rename_provider (name = 'provider_demo_delta_sharing_1', new_name =  
'provider_demo_delta_sharing_2')
```

```
//Output:  
'{ "status": true }'
```

Get Provider Procedure

This function receives the details of the Data Share Provider.

Syntax

```
Share.get_provider(name, owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Provider. This field is mandatory.
- **owner:** This field displays the Provider owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive details of a Data Share Provider:

```
name = 'provider_demo_delta_sharing_2'
s = adp.Share.get_provider(name)
json.loads(s)

//Output:
{'name': 'PROVIDER_DEMO_DELTA_SHARING_2',
 'id': 40437,
 'endpoint': 'https://abcd/',
 'shareType': 'DELTA',
 'tokenEndPoint': None,
 'shareLinks': [{ 'name': 'DEMO_DELTA_SHARING',
                  'id': 40438,
                  'shareType': 'DELTA',
                  'shareName': 'DELTA_SHARING' }]}
```

Get Providers

This function receives a list of the Data Share Providers based on the providers owner.

Syntax

```
Share.get_providers(owner)
```

The following are its parameters and descriptions:

- **name:** The name of the Provider. This field is mandatory.
- **endpoint:** This field displays the endpoint of the Provider Data Share service. It is mandatory.
- **share_links:** The provider share links dictionary has the following fields:
 - [{"shareName": "shareNameValue", "shareLinkName": "shareLinkNameValue"}]This field is mandatory.
- **owner:** This field displays the Provider owner. If this field is missing, the tool uses the current schema owner.

Example

In this example, you can receive a list of Data Share Providers:

```
s = adp.Share.get_providers()
json.loads(s)

//Output:
[{'id': 40437,
 'name': 'PROVIDER_DEMO_DELTA_SHARING_2',
 'owner': 'ADMIN',
```

```
'endpoint': 'https://abcd/delta-sharing/',
'shareType': 'DELTA'}]
```

Catalog Functions

The Oracle Data Studio Catalog or Catalog is a multi-catalog tool that provides a way to search for data and other objects in your currently connected Autonomous Database, and also in a wide range of other connected systems.

The Catalog API functionality enables to browse, search and discovering connected data assets from anywhere in the cloud and beyond.

It also enables to mount new catalogs over other systems and search for data and other items in the connected Autonomous Database.

The Catalog API functionality enables to mount new catalogs over:

- Other Autonomous Databases in your tenancy.
- Any other database that can be connected using a DB Link, for example an on-premises database that your Autonomous Database can connect to.
- Shared data, for example data shared from DataBricks using Delta Sharing.
- Existing external data Catalogs such as AWS Glue, or the OCI Data Catalog.

This below section provides functions examples to help you get started using the Oracle Data Studio Catalog API.

Before using these examples, you must create the ADP instance and then connect.

```
import adp as ords
adp = ords.login('<protocol://host:port>', 'schema_name', 'schema_password')
Catalog = adp.Catalog
```

Get Catalogs Procedure

This function receives list of catalogs.

Syntax

```
Catalog.get_catalogs(search)
```

Parameters:

- **search (String):** It is user supplied catalog search text. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `None` for it.

Return Type

It is JSON String

Return Structure

```
'[
{"label": catalog name (String),
```

```

"type": catalog type (String),

"id": catalog id (String),

"details" : catalog details (JSON Nested Object),

"enabled": catalog enabled status (true | false),

"created": catalog create date (String)

"updated": catalog update date (String)},

...

]'

```

Example

```

import json
import pandas as pd
# catalog api function call
r = Catalog.get_catalogs()
# pandas data frame representation
j = json.loads(r)
pd.json_normalize(j)

```

	label	type	id	enabled	schema	created	updated
0	API_CATALOG_SHARE_1	CATALOG	"ADMIN"."CATALOG"."API_CATALOG_SHARE_1"	True	ADMIN	2025-02-23T15:23:26Z	2025-02-23T15:23:26Z
1	API_CATALOG_SHARE_2	CATALOG	"ADMIN"."CATALOG"."API_CATALOG_SHARE_2"	True	ADMIN	2025-02-23T15:23:37Z	2025-02-23T15:23:37Z
2	API_CATALOG_SHARE_3	CATALOG	"ADMIN"."CATALOG"."API_CATALOG_SHARE_3"	True	ADMIN	2025-02-23T15:23:48Z	2025-02-23T15:23:48Z
3	CATALOG_DELTA_SHARING_1	CATALOG	"ADMIN"."CATALOG"."CATALOG_DELTA_SHARING_1"	True	ADMIN	2025-03-02T14:18:44Z	2025-03-02T14:18:44Z

Error Type

JSON String

Error Structure

```

'{"code": error code (Number), "error": error message
  (String)}'

```

Get Catalog Entities Procedure

This function receives list of catalog entities.

Syntax

```

Catalog.get_catalog_entities(catalog_name, type, search)

```

Parameters:

- **catalog_name (String):** It is a user supplied catalog name.

- **type (String):** It is a user supplied catalog entity type. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `TABLE` for it.
- **catalog entity types :** 'TABLE', 'VIEW', 'SCHEMA', etc.
- **search (String):** It is user supplied catalog search text.

Examples

```
import json
import pandas as pd

# catalog api function call
r = Catalog.get_catalog_entities('CATALOG_DELTA_SHARING_1')

# pandas data frame representation
j = json.loads(r)
pd.json_normalize(j)
```

	label	type	id	schema	created	updated
0	BOSTON-HOUSING	TABLE	"DEFAULT"."DB"."BOSTON-HOUSING"@CATALOG_DELTA...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
1	COVID_19_NYT	TABLE	"DEFAULT"."DB"."COVID_19_NYT"@CATALOG_DELTA_S...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
2	FLIGHT-ASA_2008	TABLE	"DEFAULT"."DB"."FLIGHT-ASA_2008"@CATALOG_DELT...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
3	LENDING_CLUB	TABLE	"DEFAULT"."DB"."LENDING_CLUB"@CATALOG_DELTA_S...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
4	NYCTAXI_2019	TABLE	"DEFAULT"."DB"."NYCTAXI_2019"@CATALOG_DELTA_S...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
5	NYCTAXI_2019_PART	TABLE	"DEFAULT"."DB"."NYCTAXI_2019_PART"@CATALOG_DE...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
6	OWID-COVID-DATA	TABLE	"DEFAULT"."DB"."OWID-COVID-DATA"@CATALOG_DELT...	DEFAULT	2025-03-02T14:18:47Z	2025-03-02T14:18:47Z
7	DEFAULT	SCHEMA	"DEFAULT"."SCHEMA"."DEFAULT"@CATALOG_DELTA_SH...	DEFAULT	2025-03-02T14:18:49Z	2025-03-02T14:18:49Z

Return Type

It is JSON String.

Return Structure

```
'[
{
  "label": catalog entity name (String),
  "type": catalog entity type (String),
  "schema": catalog schema (String),
  "id": catalog entity id (String).
  "details" : catalog entity details (JSON Nested Object),
  "created": catalog entity create date (String)
  "updated": catalog entity update date (String)},
...
]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Get Database Links Procedure

This function receives list of catalog entities.

Syntax

```
Catalog.get_database_links(catalog_name, owner, search)
```

Parameters:

- **catalog_name (String):** It is a user supplied catalog name.
- **owner (String):** It is user supplied catalog owner name.
- **search (String):** It is user supplied catalog search text. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `NONE` for it.

Return Type

It is JSON String.

Return Structure

```
'[  
  
{"label": data base link name (String),  
  
"type": database link type (String),  
  
"catalog": data base link catalog (String),  
  
"application": data base link application (String).  
  
"credentialName": data base link credential name (String),  
  
"credentialOwner": data base link credential owner (String),  
  
"hidden": data base link is hidden (String),  
  
"valid": data base link is valid (String),  
  
"created": database link created create date (String)  
  
"updated": database link created update date (String),  
  
]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Get Autonomous Database Procedure

This function receives Oracle Data Studio Catalog autonomous databases.

Syntax

```
Catalog.get_autonomous_database(search)
```

Parameters:

- **search (String):** It is user supplied catalog search text. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `NONE` for it.

Return Type

It is JSON String.

Return Structure

```
'[  
  
{"label": autonomous database name (String),  
  
"type": autonomous database type (String),  
  
"catalog": autonomous database catalog (String),  
  
"application": autonomous database application (String).  
  
"schema": autonomous database schema (String),  
  
"workloadType": autonomous database workload type (String),  
  
"lifecycleState": autonomous database life cycle state (String),  
  
"ocid": autonomous database ocid (String),  
  
"created": autonomous database create date (String)  
  
"updated": autonomous database update date (String),  
]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Enable Catalog Procedure

This function enables Oracle Data Studio Catalog.

Syntax

```
Catalog.enable_catalog(catalog_name)
```

Parameters:

- **catalog_name (String):** It is a user supplied catalog name.

Example

```
# catalog api function call  
Catalog.enable_catalog('CATALOG_DELTA_SHARING_1')
```

Output

```
'{ "updated": true }'
```

Return Type

It is JSON String.

Return Structure

```
'{ "updated": operation status (True | False)}'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Disable Catalog Procedure

This function disables Oracle Data Studio Catalog.

Syntax

```
Catalog.enable_catalog(catalog_name)
```

Parameters:

- **catalog_name (String):** It is a user supplied catalog name.

Examples

```
# catalog api function call  
Catalog.disable_catalog('CATALOG_DELTA_SHARING_1')
```

Output

```
'{ "updated": true }'
```

Return Type

It is JSON String.

Return Structure

```
'{ "updated": operation status (True | False)}'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Unmount Catalog Procedure

This function disables Oracle Data Studio Catalog.

Syntax

```
Catalog.unmount_catalog(catalog_name)
```

Parameters:

- **catalog_name (String):** It is a user supplied catalog name.

Example

```
# catalog api function call  
Catalog.unmount_catalog('CATALOG_DELTA_SHARING_1')
```

Output

```
'{ "unmounted": true }'
```

Return Type

It is JSON String.

Return Structure

```
'{ "unmounted": operation status (true | false)}'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Preview Catalog Table Procedure

This function previews Oracle Data Studio Catalog table.

Syntax

```
Catalog.preview_catalog_table(catalog_name, table_name, schema_name,  
                              row_limit)
```

Parameters:

- **catalog_name (String):** It specifies catalog name.
- **table_name (String):** It specifies catalog table name.
- **schema_name (String):** It specifies catalog schema name.
- **row_limit (Number):** It specifies catalog table preview row limit. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value 100 for it.

Example

```
import pandas as pd  
  
# catalog api function call  
q = Catalog.preview_catalog_table(  
    catalog_name = 'CATALOG_DELTA_SHARING_1',  
    table_name   =  
    'COVID_19_NYT',  
    schema_name = 'DEFAULT')  
  
# pandas data frame representation  
pd.DataFrame.from_records(q)
```

Output

	date	county	state	fips	cases	deaths
0	2021-01-10	Washakie	Wyoming	56043.0	804	21
1	2021-01-10	Weston	Wyoming	56045.0	485	4
2	2021-01-11	Autauga	Alabama	1001.0	4902	55
3	2021-01-11	Baldwin	Alabama	1003.0	15417	173
4	2021-01-11	Barbour	Alabama	1005.0	1663	35
...
95	2021-01-11	Yakutat plus Hoonah-Angoon	Alaska	2998.0	58	1
96	2021-01-11	Yukon-Koyukuk Census Area	Alaska	2290.0	238	4
97	2021-01-11	Apache	Arizona	4001.0	8309	259
98	2021-01-11	Cochise	Arizona	4003.0	8969	164
99	2021-01-11	Coconino	Arizona	4005.0	12996	233

100 rows × 6 columns

Return Type

It is JSON String.

Return Structure

```
'[{"columnName": "row_value",.. }, {}, ...]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Mount OCI Data Catalog Procedure

This function mounts OCI Data Catalog.

Syntax

```
Catalog.mount_oci_data_catalog(catalog_name, oci_catalog_credential,  
oci_catalog_region, oci_catalog_name, oci_storage_credential)
```

Parameters:

- **catalog_name (String):** It specifies catalog name.
- **oci_catalog_credential (String):** It specifies OCI catalog credential.
- **oci_catalog_region (String):** It specifies OCI catalog region.
- **oci_catalog_name (String):** It specifies OCI catalog name.

- **oci_storage_credential (String)**: It specifies OCI storage credential. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `None` for it.

Return Type

It is JSON String.

Return Structure

```
'[{"catalogName": catalog name (String),  
  "shareName": share name (String)}]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Mount Autonomous Database Share Catalog Procedure

This function mounts autonomous database share Catalog.

Syntax

```
Catalog.mount_autonomous_database_catalog(catalog_name, database_name,  
  create_database_link,  
  database_link_name, credential_name)
```

Parameters:

- **catalog_name (String)**: It specifies catalog name.
- **database_name (String)**: It specifies autonomous database name.
- **create_database_link (Boolean)**: It creates database link name mode. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `False` for it.
- **database_link_name (String)**: It is a database link name. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `False` for it.
- **credential_name (String)**: It is a swift credential name. This parameter is not required to be provided by the user when making a request, and if it is omitted, the system will automatically use the default value `False` for it.

Return Type

It is JSON String.

Return Structure

```
'[{"catalogName": catalog name (String),  
  "shareName": share name (String)}]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

Mount Shares Catalog Procedure

This function mounts Oracle Data Studio Shares Catalog.

Syntax

```
Catalog.mount_shares_catalog(catalog_name, share_name,  
                             share_provider)
```

Parameters:

- **catalog_name (String):** It specifies catalog name.
- **share_name (String):** It is a share name.
- **share_provider (String):** It is share provider name.

Example

```
Mount Shares Catalog# catalog api function call  
Catalog.mount_shares_catalog (  
    catalog_name = 'CATALOG_DELTA_SHARING_1',  
    share_name   = 'DELTA_SHARING',  
    share_provider = 'PROVIDER_DELTA_SHARING_1')
```

Output

```
'[{"catalogName": "CATALOG_DELTA_SHARING_1", "shareName": "DELTA_SHARING"}]'
```

Return Type

It is JSON String.

Return Structure

```
'[{"catalogName": catalog name (String),  
  "shareName": share name (String)}]'
```

Error Type

JSON String

Error Structure

```
'{"code": error code (Number), "error": error message (String)}'
```

12

The Data Load Page

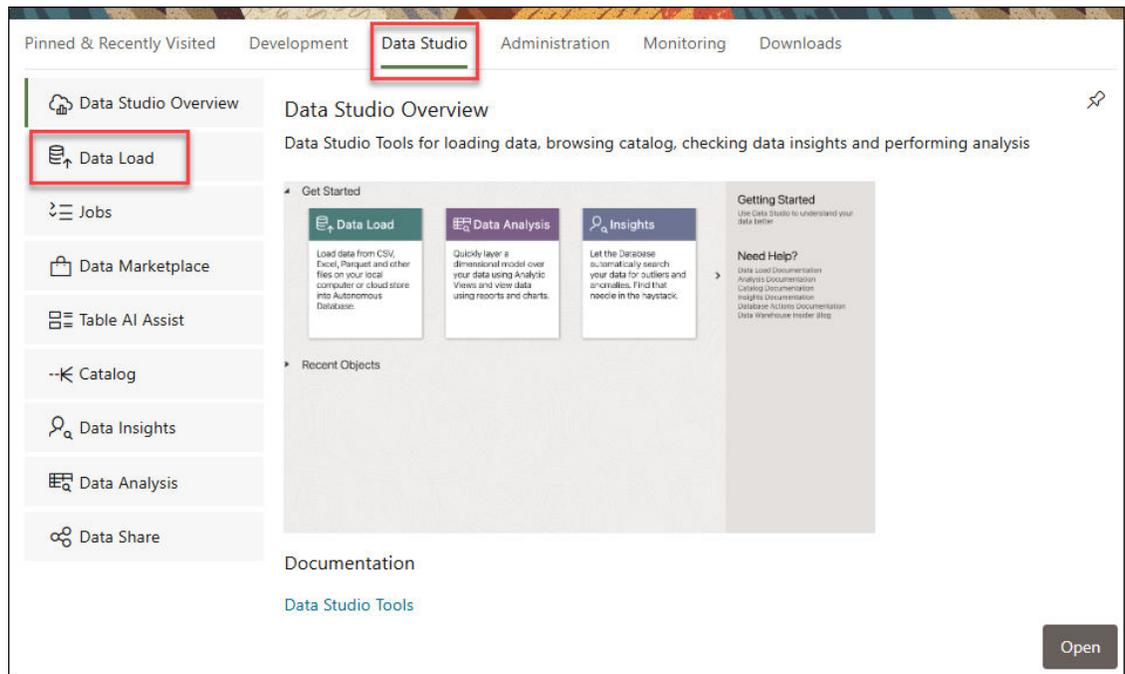
Use the Data Load dashboard page to make more data available to your Oracle Autonomous Database. You can load data from files or databases, from links to external databases or cloud storage files, or from a live feed of data from cloud storage.

From the Data Load page, you can also explore the data in your Autonomous Database and manage your cloud storage locations.

Note

If you do not see the Data Load card then your database user is missing the required DWROLE role.

To reach the Data Load page, click the **Data Studio** tab in the Database Actions page and select the **Data Load** pane.



You can alternatively click the Selector  icon and select **Data Load** from the Data Studio menu in the navigation pane.

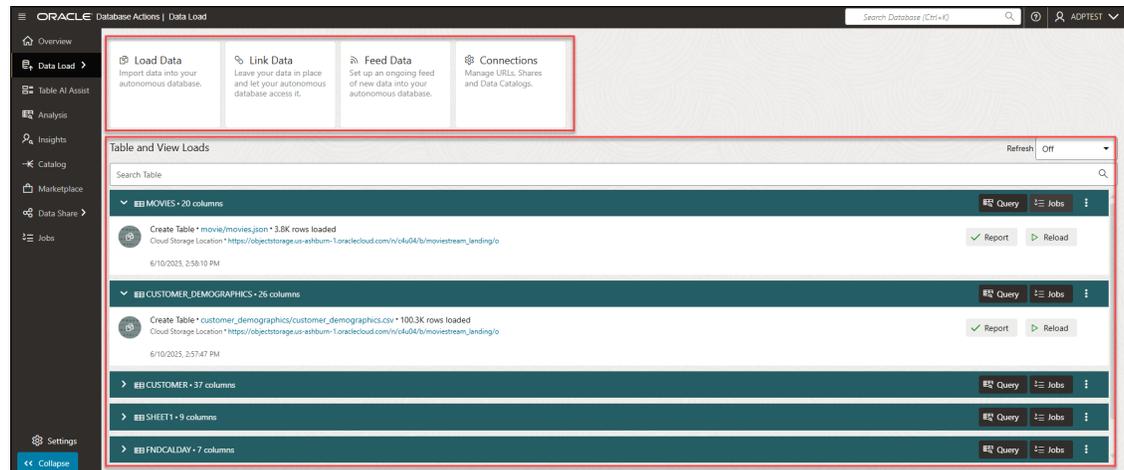
The Data Load Dashboard provides a central location for all data load and live feed tasks. The various icons on the dashboard, provide access to loading data, linking data, setting up an ongoing feed, checking data load jobs, and managing connections.

You can view the following tiles on the top section of the dashboard:

- Load Data
- Link Data
- Feed Data
- Connections

The bottom section of the dashboard displays the tables, views, and life feed you load into the tool.

The following image illustrates the **Data Load Dashboard**.



After drilling down onto a table, in the **Table and View Loads** section, you can view the details of the table load such as the name of the table, the number of rows and columns in the table, and the time you load the table.

Tables which are not active for last 7 days are collapsed.

You can also view the source location of the data you load. Click the location to view objects in the location.

Click the **Actions** icon beside the table name and select **Table** to view the following properties of the table:

- [View Details](#)
- [Gather Statistics](#)
- [Register to Cloud Link](#)
- [Publish Listings to Marketplace](#)
- [Create Analytic View](#)
- [The Table AI Assist Tool](#)
- [Export Data to Cloud](#)
- [The Data Analysis Tool](#)
- [Edit](#)
- **Drop** : Select **Drop** to delete the table.

Select **Report** to view a report of total rows processed successfully and failed for the selected table.

Select **Reload** to reload the source of the data load job. Clicking Reload takes you to the **Link Data** page where you can link your data source to the Autonomous Database.

Note

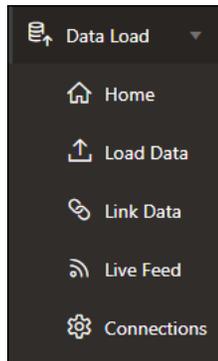
For tables you load via Cloud Store, you can view **Create Live Feed** icon. See [Create Live Feed from Data Load](#). After the creation of live feed is successful, you will view a Live Feed badge besides the table load header.

In case of errors, you can reload data with suggested fixes.

To load or create links to data or create live table feeds, on the Data Load dashboard, select a combination of an operation and a data source location. The following table lists the operations and the source locations that support those operations.

Operation	Source Location	Description
Load data	Local file Database Cloud storage Directory Share AI Sources	Load data from files on your local device, from remote databases, or from cloud storage into tables in your Oracle Autonomous Database.
Link data	Database Objects in Oracle Cloud Infrastructure (OCI) Catalog	Create external tables or views in your Oracle Autonomous Database that link to data in cloud storage or remote databases. Changes to the source data automatically appear in the target objects.
Feed data	Cloud storage	Set up a feed of data from a cloud storage bucket into a table. Changes to the source data load into the target table as scheduled or on demand.
Connections	Cloud storage Data Catalog links Share Providers	Create a cloud storage location. Subscribe to Share Provider and Register Data Catalog. You can manage cloud storage locations, Data Catalog and Shares.

The left navigation pane of the Data Load Dashboard provides links to the Data Load features and pages. When you select a link from the left navigation pane, the corresponding page appears in the target content area.



- Home: This is the current Data Load dashboard that you view.
- [Load Data](#)
- [Link Data](#)
- [Live Feed](#)
- [Connections](#)

Topics:

- [Data Studio Settings](#)
This topic explains the Data Studio settings on the Data Studio tool.
- [Managing Connections](#)
Connections that are established from the Data Studio to the cloud, catalogs and shares are listed on this page.
- [Loading Data](#)
You can load data from files on your local device, from remote databases, or from cloud storage buckets from directories and share providers. The file formats that you can load are CSV, XLS, XLSX, TSV, TXT, XML, JSON, GEOJSON, AVRO, PARQUET, GZ, GZIP, ZIP, PDF, PNG, JPG, JPEG and TIFF.
- [Linking Data](#)
You can link to data in remote databases or in cloud storage buckets.
- [Feeding Data](#)
You can run a live table feed on demand, on a schedule, or as the result of a notification.
- [DBMS_LIVE_FEED Package Reference](#)
This chapter provides information about the packages you use with the Data Load Tool in Data Studio. The `DBMS_LIVE_FEED` topic also covers the procedures included in the `DBMS_LIVE_FEED` package.

Data Studio Settings

This topic explains the Data Studio settings on the Data Studio tool.

This settings on the **Data Studio** navigation menu and the Data studio overview page enables you to use the associated Cloud Service, configure AI profile, configure SMTP and create Vector Indexes.

This helps in defining the data during the load. This setting sets a variety of general preferences like setting default credentials to define the data you load, configure SMTP for data sharing, configure AI profile and configure Marketplace cloud link access.

Integration of Slack with Data Studio bridges the gap between real time collaboration between various team members in staying informed and making correct decisions.

To set the Data Studio Settings, on the **General** tab of the wizard, specify the email address to configure your profile for Marketplace and Sharing.

On the **Cloud Services** tab of the wizard, under **Cloud Object Store Access**, specify the following field values:

- **OCI Credential:** Selecting an OCI credential enables you to discover the OCI buckets. Click **+** to create OCI Credentials.
- **OCI Compartment:** This drop-down field lists the buckets present in your compartment for loading and linking from Oracle cloud services. If you select any other cloud service provider apart from Oracle, this field displays **None** by default.
- **AWS Credential:** Selecting an AWS credential in Data Studio Settings enables you to discover the S3 buckets. Click **+** to create credentials for Amazon S3 cloud storage.
- **Azure Credential:** Selecting an Azure credential in preferences enables you to discover the Azure BLOB storage. Click **+** to create credentials for Microsoft Azure BLOB storage and Azure Data Lake Storage.
Click **Check Object Store Access** to check if you have access to the object store. You will view a Policy Check wizard with the following bucket lists:
 - OCI list buckets
 - AWS list buckets
 - Azure list buckets

The Object Storage Listing Policy checker evaluates if you have access to the buckets. If yes, the checker box is highlighted in green with a check sign, else the checker box displays the policy highlighted in red with a red cross.

Under **Cloud Service Provider for AI:** Select any of the following options from the drop-down field:

- Oracle Cloud Interface (OCI)
- Amazon Web Services (AWS)

Click **Check Cloud Service Access**.

You will view the following options in the Cloud Services Policy Checker dialog box. The AI cloud policies listed below must be checked against the listed policies and approved by the Cloud Service vendors for support. The Data Studio tool will test these services and inform you if the data you load in the tool is able to access the services.

- **Language-** Checks if the Data Studio tool has accessed the Cloud's language processing abilities. See [Use OCI Language Service Capabilities in Data Studio](#) for more information on this service.
- **Document Understanding-** Checks if the data you load into the Data Studio tool has accessed the Cloud's Document Understanding AI ability. See [Use OCI Document Understanding to extract Tables from Images](#) for more information on this service.
- **Translation-** Checks if the Data Studio tool accesses the OCI Translation feature. See [Perform Text Translation](#) for more information on this service.

On the **AI Profile** tab of the Data Studio Settings wizard, you can configure your AI preferences.

1. Select **AI Profile** from the drop-down use this AI profile to support and configure how natural language prompts are translated into SQL statements. The drop-down lists the AI Profiles you create from the **Create AI Profile** icon. Click **+** to create AI profile.
 - To utilize the AI services in Data Studio, you must have either an **OCI AI** or **OpenAI** or **Cohere** or **Azure OpenAI service** account.
 - You must also have access to `DBMS_CLOUD_AI` package to set your AI profile.
 - If you have Select AI set up using either **OpenAI** or **Cohere** or **Azure OpenAI** service, you can enhance your data by loading small reference data tables that are generated by large language models. Try out the suggested prompts or come up with your own to generate data to load into the autonomous database. You can also use this AI profile to generate Marketplace dataset description and tags. This AI profile is used to generate table description in Catalog App.

See [Use DBMS_CLOUD_AI to Configure AI Profiles](#) to create and configure your AI profile.

2. Click **Check AI Access**.

The **AI Policy checker** checks for the following service:

- **Generative AI**- The Data Studio tool checks if it has access to the Generative AI service.

3. Under **Manage AI Profiles**, you can view a tabular data that displays information about the AI Profile. It displays the following field values based on the AI Profile:

- Profile Name
- Provider
- Credential
- Model
- Actions: You can edit any of the fields you configured to create the AI Profile. You can also delete the AI profile.

For more details on the column fields displayed under **Manage AI Profiles**, refer to [Create AI Profile](#).

On the **Vector Index** tab of the Data Studio Settings wizard, you can create Vector Index.

Note

You cannot view the **Vector Index** tab with **Oracle AI Database 19c**.

You can view a list of Vector Indexes under **Manage Vector Indexes**. If you do not have an available Vector Index, click **Create Vector Index**.

Selecting **Create Vector Index**, opens **Create Vector Index** wizard.

1. On the **Required Settings** tab, you can view the fields that configure the vector index you create and specify the source from which its data is loaded:
 - **Vector Index Name**: Specify the name of the vector index object that you create using this wizard. You can use this name in SQL and in the console to manage or query the index.
 - **Location**: Select a cloud location of the source data for indexing from the drop-down field. You can also type the location name. It is an Object Storage bucket or similar cloud store that the database reads from when building or refreshing the vector index.

- **AI Profile:** Select the AI profile you create from the AI Profile tab in the Data Studio Settings wizard. This field uses the Vector Index linked to the AI profile field in the [Create AI Profile](#) wizard.
- **Object Storage Credential Name:** Select a stored credential from the drop-down that allows the database to securely access the specified Object Storage location (for example, an OCI user credential or resource principal-based credential).

You can click **Create** to create the Vector Index and skip the Optional Field values.

Click **Next** to specify the following **Optional** field values.

These optional fields control how documents are chunked, indexed, and retrieved during vector search. You can optionally fill these fields to control how the vector index is built, updated, and queried so that it returns useful results for your workload.

2. On the **Optional Settings**, specify the following field values:
 - **Chunk Size:** Specify the number of characters (or tokens, depending on implementation) in each piece of text that is embedded and stored as one vector. Larger chunks capture more context but may be less precise for passage-level matches.
 - **Chunk Overlap:** Specify the amount of overlap between consecutive chunks so that content near boundaries appears in multiple chunks, reducing the chance of missing relevant text that falls at a split point.
 - **Refresh Rate:** Specify the Interval (in minutes) at which the vector index is automatically refreshed from the source data so that new or updated content is re-embedded and searchable.
 - **Vector Distance Metric:** Specifies the similarity metric used to compare vectors when searching (for example, COSINE), which determines how “closeness” between query and stored vectors is computed.
 - **Match Limit:** Specifies the maximum number of results the vector search returns for each query.
 - **Similarity Threshold:** Specifies the minimum similarity score a result must have to be returned from a vector search. Results with scores below this threshold are filtered out as not relevant enough.

Click **Create** to create Vector Index.

After creating the Vector Index, you can **Edit** a previously created AI Profile from **Manage AI Profiles** and add the Vector Index.

On the SMTP tab of the Data Studio Preferences wizard, you can configure email service like a local email client on your system (e.g., Thunderbird).

Note

You must configure SMTP only once and then the system uses the saved configuration from that point forward.

Specify the following field values in the SMTP tab of the wizard:

1. **Server host:** Enter the endpoint used to send the email. For example, *internal-mailrouter.oracle.com*.
2. **Server port:** Enter the SMTP port used to accept an email. Email Delivery supports TLS on port 25. Sender:

3. Enter the email address of the sender. For example, `oaappgens_us@oracle.com`.
4. **Server Encryption:** This field indicates if TLS, the standard means of performing encryption in transit for emails, is being used. Providers must encrypt emails while it's in transit to the Oracle Cloud Infrastructure Email Delivery service. Encrypted emails are protected from being read during transit. If there is no encryption, enter, *None*.
5. Select credential to use for SMTP connection from the drop-down. If there is no available credential in the drop-down, you can create credential by clicking **Create Credential** icon. Refer to [Create Oracle Cloud Infrastructure Native Credentials](#) for more details.

Click **Test** to test the SMTP configuration. You will view a screen that asks you to run the ACL script. You can run the script if you have the ADMIN rights. This is just a first time setting. After you receive a successful message of SMTP Test, you can save the SMTP configuration.

On the **Marketplace** tab, you can set the configurations you require for the Data Marketplace tool.

Access to data in Data Marketplace is restricted based on user authorizations. As non admin users you have permission to access the Data Marketplace tool based on the Data Studio Settings configuration. You can view the following options:

- **Restricted, Requires Authorization:**
Selecting this option enables you to publish restricted cloud links and, consequently, restricted marketplace listings. You will have the ability to grant and revoke access to these listings to specific OCIDs.

- **Publish:**
This option specifies who and from where you are allowed to access the registered table.

You can choose from any of the following available options:

- **TENANCY:** You can grant remote data access to any resource, tenancy, compartment, or database in the tenancy of the Autonomous Database instance that is registering the data set. This scope is more restrictive than the **REGION** scope.
- **COMPARTMENT:** You can grant remote data access to any resource, compartment, or database in the compartment of the Autonomous Database instance that is registering the data set.
- **REGION:** You can grant remote data access via cloud links to other tenancies in the Autonomous Database instance that is registering the data set.

Note

Marketplace listing currently can only be published up to the compartment level. The tool does not allow you to publish a marketplace listing to entire region for security reasons.

- **None:** You do not have access to any resource.

- **Read:**
This option allows **READ** access for the specific user to access the registered table.

On the **Slack** tab you can configure your Slack profile to receive notifications on certain events while using the Jobs feature.

Specify the following field values:

- **Slack Credential Name:** Enter the Credentials from the drop-down. The drop-down lists the credentials in your compartment. Click **+** to create your own credentials.

- Selecting **+** opens a Create Credential dialog box. Specify the following field values to create a credential:
 - * **Credential Name:** Enter a name of your choice. For example, `SLACK_CRED`.
 - * **Cloud Service:** Select **Slack** from the drop-down.
 - * **Username:** Enter your OCI user name. You must use Oracle Cloud Infrastructure User Name from your profile in the OCI console. For example:
`oracleidentitycloudservice/foo@example.com`. Or
`default/foo@example.com`
 - * **Password:** `*****`. For the slack provider, the username value can be any valid string and the password is the Slack bot token.
Click **Create Credential** to create a new credential.

Enter this value in the **Slack Credential Name** field.

- **Slack Channel ID:** This specifies the Channel ID in a `String` value. The Channel ID is a unique ID for a channel and is different from the channel name. In Slack, when you view channel details you can find the Channel ID on the `About` tab.
- **Slack Message:** Enter a message to test the notifications.
Click **Test** to test the notification. You receive a confirmation message after you successfully authenticate into Slack from Data Studio Settings.. Click **Save** to set the Slack Credential.

You can refer to the [Create Job](#) to set up notifications to receive Slack alerts about the status of a job run.

On the **Share provider Identity** tab you can provide the details of the provider before you share the data. The share provider identification will be available to recipients with whom you grant the share while using the Data Share tool. This provides information to the recipient on how to identify you.

Specify the following field values:

- **Name:** Enter the name of the provider. For example, `ams`.
- **Email:** Enter the email address of the provider. For example, `ams@gmail.com`.
- **Description:** Enter a description of the provider. For example, `AMS sharing stuff`.

Click **Save** to save the provider identification information. You will receive a successful message of configuring the provider identification information.

Select **Apply Changes** to save and apply the configuration you made in the Marketplace tab of the Data Settings dialog.

You have successfully set and configured your preferences in the Data Studio Settings wizard.

- [Slack Integration with Data Studio](#)
You can integrate Data Studio with Slack on Data Studio Settings using Slack Channel ID and Credentials. With this configuration you can send or schedule reports of the Jobs feature when certain events occur in Slack. The team members of the Slack channel can view the updates, discuss the data insights, and make well informed decisions.
- [Create AI Profile](#)
Autonomous AI Database uses AI profiles to facilitate and configure access to an LLM and to setup for generating, running, and explaining SQL based on natural language prompts using **Data Studio**.

Slack Integration with Data Studio

You can integrate Data Studio with Slack on Data Studio Settings using Slack Channel ID and Credentials. With this configuration you can send or schedule reports of the Jobs feature when certain events occur in Slack. The team members of the Slack channel can view the updates, discuss the data insights, and make well informed decisions.

You can:

- Schedule job reports sent to Slack.
- Set up real time triggers whenever the job is complete and also display it's running time.
- You can customize what notifications you want to receive from the reporting tab within the [job creation feature](#).

You must first select and authenticate the Slack profile you will use to integrate with the Jobs feature of the Data Studio suite of tools.

By automatically sharing reports and making data available in your team's Slack channel, you can collaborate more effectively and enhance workflow efficiency.

You can configure your or your team's Slack profile by setting it up in the Slack tab of Data Studio Settings.

Refer to the previous chapter of [Data Studio Settings](#) for information on configuring the Slack profile .

Create AI Profile

Autonomous AI Database uses AI profiles to facilitate and configure access to an LLM and to setup for generating, running, and explaining SQL based on natural language prompts using **Data Studio**.

Before creating an AI Profile using Data Studio, you must know the following concepts:

- **Retrieval Augmented Generation (RAG) Settings:** RAG Settings configure Retrieval-Augmented Generation pipelines in Oracle Autonomous Database Serverless, enabling AI profiles to load and query unstructured data from knowledge bases alongside structured tables. These settings define how documents are processed into vectors for semantic search, improving `SELECT AI` accuracy for natural language queries.

Note

You cannot access RAG features like creating and using Vector Indexes in **Oracle AI Database 19c**. These features are supported in **Oracle AI Database 26ai**.

- **Vector Index:** Vector indexes must be specified to store embeddings generated from RAG pipelines or table data, allowing efficient similarity searches during AI inference. In AI profile creation via `DBMS_CLOUD_AI.CREATE_PROFILE`, the Vector Indexes link the profile to Oracle's vector search capabilities, ensuring the AI model retrieves relevant context from vectorized data. You can create Vector Indexes from [Data Studio Settings](#).
- **Table Metadata:** Table metadata, provided as an `object_list` in the profile's attributes (e.g., `[{"owner": "SH", "name": "customers"}]`), scopes the AI's access to specific database tables and enforces security policies. This prevents the AI from querying unintended data, aligns with IAM controls, and optimizes natural language to SQL generation for defined schemas.

For more details, see [Select AI with Retrieval Augmented Generation \(RAG\)](#).

The AI Profiles support two modes:

- **RAG Mode:** It utilizes Vector Index for semantic search over unstructured data.
- **Natural Language to SQL (NL2SQL) Mode:** It does not require Vector Index to enable table-focused options like object list mode.

Selecting **+** opens Create AI Profile wizard:

1. On the **Service** tab of the Create AI Profile wizard, specify the following field values:

Create AI Profile

1 ————— 2 ————— 3
Service **RAG Settings** **Table Metadata**

Use these settings to set up provider endpoints and adjust LLM behavior.

Profile name *

AI Provider * ⓘ

AI Model * ⓘ

Credential * ⓘ

> **Advanced Settings**

Show code

Back Next Create AI Profile Cancel

- **Profile Name:** Enter the name of the AI profile. For example, *OCI AI*.
- **AI Provider:** Select any of the following providers from the drop-down field:
 - *OCI*
 - *Open AI*
 - *Azure OpenAI Service*
 - *Cohere*
 - *Google*
 - *Anthropic*
 - *Hugging Face*
- **AI Model:** Select the available AI models from the drop-down that your AI profile will point to so that the database answers natural language questions over your data and generate or explain SQL, including retrieval-augmented results using embeddings and vector indexes.

- **Credential:** These are the stored credentials that the database uses to securely connect to the external AI provider your profile is configured to use.

Under **Advanced Settings**, you will view **Generation Settings** and **Service Settings**:

- **Generation Settings:** Generation Settings help you configure the behavior and limits of the AI model accessed through the profile in Oracle Cloud Infrastructure's AI services, allowing customization for different use cases like chat, SQL generation, or natural language queries from Autonomous Database.
- **Service Settings:** Service Settings refer to the configuration fields that dynamically appear based on the selected AI Provider, allowing users to input provider-specific credentials and parameters like endpoints, regions, or deployment names.

Note

For AI Providers such as *OpenAI*, *Cohere*, *Google*, *Anthropic*, or *Hugging Face*, the *Service Settings* section is **not** displayed.

The **Service Settings** field values varies based on the AI Provider you select.

When you select **OCI** from **AI Provider**, you will view the following fields:

- **OCI API Format:** This specifies what request and response schema the Generative AI endpoint expects, for example *COHERE*, *Generic*, or an *OCI-native format* for Oracle models.
- **OCI Region:** This is the OCI region identifier where the endpoint is running. It ensures calls from your database are routed to the correct regional AI models. Each Autonomous AI Database OCI region supports different LLMs—some regions like *us-chicago-1*, *uk-london-1*, and *ap-osaka-1* offer many, while others have fewer. We only show models available in your selected region. If the OCI AI model does not appear, you can switch regions to refresh the model list. For more information, refer to [Regions with Generative AI](#) and [Pretrained Foundational Models in Generative AI](#).
- **OCI Compartment ID:** This is the OCID of the OCI compartment that owns the endpoints you want to use. It scopes access and billing to the correct logical container in your tenancy, aligning with your IAM policies.
- **OCI Endpoint ID:** This is the OCID of a specific AI endpoint on a dedicated AI cluster, for example *ocid1.generativeaiendpoint.oc1.us-chicago-1*. Specifying this field tells Select AI to use that dedicated endpoint instead of a shared on-demand model.

When you select **Azure OpenAI Service** from the **AI Provider** field, you will view the following fields:

- **Azure Resource Name:** Specify the unique name of your Azure OpenAI Service resource instance, which forms part of the endpoint URL (e.g., *https://your-resource-name.openai.azure.com/*) for accessing the service.
- **Azure Deployment Name:** Specify the custom name you assign when deploying a specific language model within that Azure OpenAI resource.
- **Azure Embedding Deployment Name:** Specify the custom name of a separate deployment for an embedding model (such as *text-embedding-ada-002* or *text-embedding-3-large*) within the same resource, used exclusively for creating vector representations of text.

See [Azure OpenAI Embedding skill - Azure AI Search | Microsoft Learn](#) for more information on the above fields.

Under **Generation Settings**, you can view the fields:

- **Stop Tokens:** These are token sequences that instruct the AI model where to stop generating further text output. When the model generates any of these tokens, it stops. This helps in controlling the response length and preventing unwanted trailing text.
- **Max Tokens:** This defines the maximum number of tokens the AI model can generate in response to a prompt. It limits the length of the generated output to avoid overly long or expensive responses.
- **Case Sensitive Values:** This setting indicates whether the input values or parameters should be treated as case sensitive by the AI processing. If true, exact casing must be matched for input or parameter values.
- **Conversation:** This typically indicates whether the AI profile supports conversational context, allowing the model to retain context over multiple interactions or prompt exchanges—useful for chat-style AI applications.
- **Temperature:** This is a slider parameter controlling randomness in the AI's responses. A lower temperature (e.g., 0.2) makes output more focused and deterministic, while a higher temperature (e.g., 0.8) yields more creative and diverse text generation.

Click **Next** to proceed to the **RAG Settings** tab of Create AI Profile.

2. RAG Settings:

You can view the following *optional* fields in RAG Settings.:

- **Vector Index:** Select the Vector Index you create from the list of available vector indexes. It specifies which Oracle Database vector index to use for storing and searching the embeddings of your documents, so the AI profile can quickly find the most relevant content during Retrieval-Augmented Generation. See [Create Vector Index](#) for more information.

Your AI Profile requires a Vector Index to function.

- **Embedding Model:** It specifies which embedding model is used to convert your text into vector representations that are stored in that index, ensuring the semantic search behavior matches the model you configure for RAG.

The above fields are *optional*. You can skip them and click **Create AI Profile** to create AI Profile. You can alternatively proceed to the **Table Metadata** tab.

3. Under **Table Metadata** tab, you can view the following *optional* fields:

Create AI Profile

1 ————— 2 ————— 3

Service
RAG Settings
Table Metadata

Use these settings to provide AI with essential context about your data, helping to improve the accuracy of its results.

Object List Mode ?

Annotations ?

Comments ?

Constraints ?

Show code

Back
Next
Create AI Profile
Cancel

? Note

If you skipped selecting a Vector Index in the previous step, you will view a setting called **Object List Mode**.

- **Object List Mode:** You can select any of the available field options:
 - **All:** This option sends metadata for all tables in the specified schemas to the LLM, so it can generate SQL against any of them.
 - **Automated:** This option lets Select AI automatically choose a subset of relevant tables by running vector search on schema metadata, so only the most likely tables for a given prompt are used.
 - **Selected Tables:** This option limits metadata to only the specific tables you list in the AI profile, so SQL generation is restricted to those tables.
You can also view **Object List** where you can select list of objects LLM can use. Selecting **Object List** will also show another additional setting **Enforce Object List** switch.
 - **None:** This option does not send any table metadata from an object list, so the AI either cannot generate SQL over tables or rely on other configuration that does not use an explicit object list.

? Note

You cannot view the **Object List Mode** field with **Oracle AI Database 19c**.

- **Annotations:** Use this toggle to enable Annotations that indicates whether the AI can use extra descriptive metadata attached to objects (like column descriptions or tags) as additional context when generating results.

- **Comments:** Use this toggle to controls if table and column comments from the data dictionary are exposed to the AI as context, which can help it better understand the meaning of your data.
- **Constraints:** Use this toggle to controls whether information about primary keys, foreign keys, and other constraints is provided to the AI, helping it reason about relationships between tables and improve query accuracy.

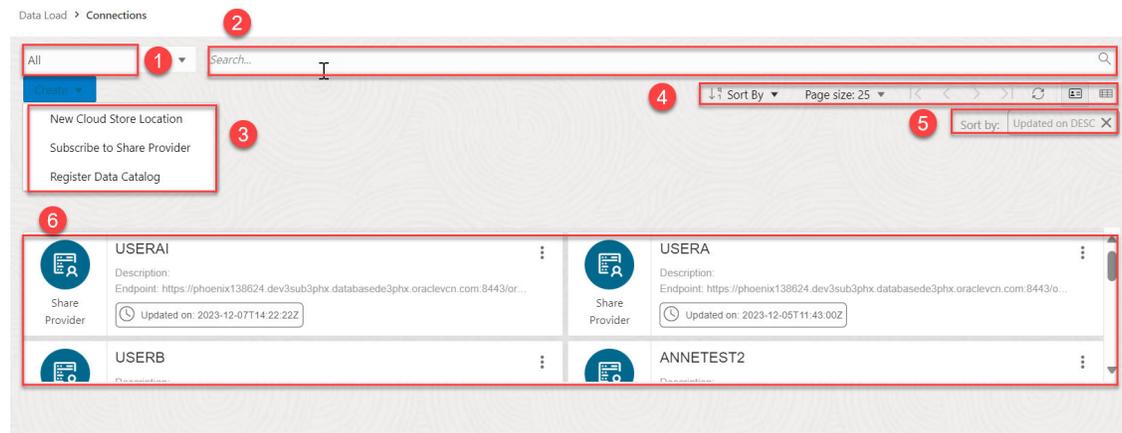
Click **Create AI Profile**.

You have successfully created and configured your AI profile in the Data Studio Settings wizard.

Managing Connections

Connections that are established from the Data Studio to the cloud, catalogs and shares are listed on this page.

The Connections page contains the Search for entities such as Cloud Storage Links, Data Catalog Links, and Share Providers, and a list of entity cards. You can enter the entity you are looking for in the field or click one of the entity card from the list. You can register the cloud store you want to use from this page. You can also register data catalogs and subscribe to a share provider.



The Connections page consists of the following:

1. **Entity Selector:** You can Use the drop-down lists to select the entity from which the connection is created. You can select from Cloud Storage Links, Data Catalog Links, Share Provider or all.
2. **Search field:** Searches for entities in the field by name. The search functionality is not case-sensitive, retrieves all matching entries, and does not require the use of wild card characters.
3. **Create drop-down:** The options available in the Create drop-down are:
 - **New Cloud Store Location:** Before you can load data from a cloud store, you must establish a connection to the cloud store you want to use. You can select cloud store location from the cloud store locations field. A cloud storage link is a connection to a bucket in a cloud store. You can view the existing cloud storage links and add new ones. Refer to the [Create Credentials](#) to add a cloud store location.

- **Subscribe to a Share Provider:** Upload the JSON profile file and create a share provider description. To subscribe, you need to use the information contained in the uploaded JSON profile you received from the share provider. From the Consume Share feature of the Data Share tool, you upload the JSON profile and follow the subscribe wizard. Refer to the [Consume Versioned Share](#) for more details.
 - **Register Data Catalogs:** You can create a connection by registering a data source as a data asset in your data catalog. You can view, delete and rename the data catalogs. Refer to the [Register Data Catalog](#) chapter.
4. The toolbar consists of the following buttons:
- **Sort By**

To select sorting values, click the **Sort By** button to open the list of options. Then click the **Ascending** or **Descending** icon next to one or more of the sorting values. For example, if you select the **Ascending** icon next to **Entity name** and the **Descending** icon next to **Entity type**, the entities will be sorted in alphabetical order by entity name and then in reverse alphabetical order by entity type.

Click **Reset** in the list to clear the choices in the list.
 - **Page size**

By default, up to 25 entities are displayed on the page. If you want more entities on a page, select a number from this list.
 - **Previous and Next**

If the search results are displayed on multiple pages, click these buttons to navigate through the pages.
 - **Refresh**

Click to refresh the data load jobs shown on the page, based on the current search field.
5. The sorting values you choose are listed next to the **Sort by** label beneath the toolbar. Click the **X** icon on a sorting value to remove it.
6. Display area: The area below the Create drop-down field displays the entity carts returned by a search.

View Entity Details

- To view details about the existing entities, click **Actions**.
- For a Share Provider, refer to the [View Share Provider Entity detail](#).
- For a Cloud Storage Link, click **Actions** to perform the following operations:
 - Select **View Details** to view details about the table.
 - Select **Objects** to view objects available in the selected storage link. You can click a file on the navigator pane to view it on the display area.
 - Selecting **Link Tables** opens the Link Data page on the Data Load tool with the selected cloud storage link on the Cloud Location URL field. You can link data present in the cloud storage to the Autonomous AI Database. See [Linking to Objects in Cloud Storage](#).
 - Selecting **Load Tables** opens the Load Data page on the Data Load tool with the selected cloud storage link on the Cloud Location URL field. You can load data present in the cloud storage to the Autonomous AI Database. See [Loading Data from Cloud Storage](#).

- Selecting **Create Live Table Feed** opens the Create Live Table feed wizard with the selected cloud storage link on the Cloud Location URL field. See [Feeding Data](#) to view more details.
- Select **Edit** to update any details on the cloud storage location. See [Create Credentials](#) to view details on creation of cloud storage location.
- Select **Rename** to rename the cloud store location to a different name.
- Select **Delete** to delete the cloud store location.
- [Register Data Catalog](#)
You can register data catalogs you want to use with registering the data catalog.
- [Create Credentials](#)
The procedure for creating a credential varies depending on the cloud storage provider. If your source files reside in a cloud store provided by one of the following, see the example for that provider.
- [Create Oracle Cloud Infrastructure Native Credentials](#)
To establish cloud storage connection from Data Studio to Oracle Cloud Infrastructure (OCI) Object storage service, you must configure the cloud storage location with your OCI authentication details. You can create Oracle Cloud Infrastructure (OCI) Native Credentials by using the CREATE_CREDENTIAL procedure of DBMS_CLOUD package.
- [Create Amazon Web Services \(AWS\) Credentials](#)
To access AWS Glue Catalog you must have AWS credentials that you can create via the Create Credential wizard. This wizard stores cloud service credentials in the Autonomous AI Database.
- [Manage Resource Principal with DBMS_CLOUD](#)
The Oracle Cloud Infrastructure (OCI) Resource Principal allows applications and services running within the OCI to access an Autonomous Database without the need for traditional database credentials (e.g., username and password).

Create Credentials

The procedure for creating a credential varies depending on the cloud storage provider. If your source files reside in a cloud store provided by one of the following, see the example for that provider.

- **Oracle Cloud Infrastructure (OCI)**, see [Create an OCI Cloud Store Location](#).
- **Amazon S3**, or you are calling an AWS API, see [Create an Amazon S3 Cloud Store Location](#).
- **Microsoft Azure Blob Storage** or you are calling an Azure API, see [Create a Microsoft Azure Cloud Store Location](#).
- **Google Cloud Storage**, see [Create a Google Cloud Storage Location](#).
- **Other (Swift compatible)** cloud storage, see [Create an Other \(Swift Compatible\) Cloud Store Location](#).
- **OCI cloud storage** by using native OCI credentials, see the subsequent section.

Note

This method is suggested if you need to use the OCI REST APIs. You need to use the Native OCI credentials if you want to use the OCI REST APIs. To create an OCI cloud storage location using Oracle Cloud Infrastructure Signing Keys, you must first [Create Oracle Cloud Infrastructure Native Credentials](#).

- **OCI Cloud Shell**, see [Create Credentials using OCI Cloud Shell](#).

Create an OCI Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.
2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

```
My_Cloud_Store
```

3. (Optional) In the **Description** field, enter a description for the link. For example:

```
My cloud storage link.
```

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard. In the Credentials section, select **Cloud Username and Password**.
5. From the **Cloud Store** drop-down list, select **Oracle**.
6. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

```
my_credential
```

7. For an OCI cloud store, in the **Oracle Cloud Infrastructure User Name** field, enter your OCI user name. You must use Oracle Cloud Infrastructure User Name from your profile in the OCI console. For example:

```
oracleidentitycloudservice/foo@example.com
```

or

```
default/foo@example.com
```

8. For an OCI cloud store, in the **Auth Token** field, enter your auth token. For example:

```
LPB>Ktk(lMlSD+a]r
```

9. In the **Bucket URI** field, enter the URI and bucket for your OCI instance bucket.
 - a. To get the URI and bucket, go to the bucket in the Object Storage compartment in your Oracle Cloud Instance.
 - b. In the Objects group, click the Actions (three vertical dots) icon for a file in the bucket, then click **View Object Details**.

- c. Copy all of the URL Path (URI) except for the file name. Be sure to include the trailing slash. For example, for the file `https://objectstorage.us-phoenix-1.oraclecloud.com/n/myoci/b/my_bucket/o/MyFile.csv`, select the following:

```
https://objectstorage.us-phoenix-1.oraclecloud.com/n/myoci/b/  
my_bucket/o/
```

- d. Paste the string into the **URI + Bucket** field.
10. Click **Next**.
The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

 **Note**

The display area is blank when we create a new cloud storage location.

11. Click **Create** create the cloud storage location.

Create an Amazon S3 Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.
2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

```
My_Cloud_Store
```

3. (Optional) In the **Description** field, enter a description for the link. For example:

```
My cloud storage link.
```

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard.
In the Credentials section, select **Cloud Username and Password**.
5. From the **Cloud Store** drop-down list, select **Oracle**.
6. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

```
my_credential
```

7. From the **Cloud Store** drop-down list, select **Amazon S3**.
8. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

```
my_credential
```

9. In the **AWS Access Key ID** field, enter your AWS access key ID. For example: `myAccessKeyID`
10. In the **AWS Secret Access Key** field, enter your AWS secret access key. For information on AWS access keys, see [Managing access keys for IAM users](#).

11. In the **Bucket URI** field, enter the URI and bucket for your Amazon S3 bucket. For example:

```
https://s3.us-west-2.amazonaws.com/my_bucket
```

12. Click **Next**.
The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

 **Note**

The display area is blank when we create a new cloud storage location.

13. Click **Create** create the cloud storage location.

Create an Microsoft Azure Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.
2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

```
My_Cloud_Store
```

3. (Optional) In the **Description** field, enter a description for the link. For example:

```
My cloud storage link.
```

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard.
In the Credentials section, select **Cloud Username and Password**.
5. From the Cloud Store drop-down list, select **Microsoft Azure**.
6. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

```
my_credential
```

7. In the **Azure Storage Account Name** field, enter the name of your Azure storage account. For example:

```
myaccount
```

8. In the **Azure Storage Account Access Key** field, enter your Azure access key. For information on Azure storage accounts, see [Create a storage account](#).
9. In the **Bucket URI** field, enter the URI and bucket for your Microsoft Azure bucket. For example:

```
https://myaccount.blob.core.windows.net/mycontainer
```

10. Click **Next**.
The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

Note

The display area is blank when we create a new cloud storage location.

11. Click **Create** to create the cloud storage location.

Create an Microsoft Azure Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.
2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

`My_Cloud_Store`

3. (Optional) In the **Description** field, enter a description for the link. For example:

`My cloud storage link.`

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard. In the Credentials section, select **Cloud Username and Password**.
5. From the Cloud Store drop-down list, select **Microsoft Azure**.
6. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

`my_credential`

7. In the **Azure Storage Account Name** field, enter the name of your Azure storage account. For example:

`myaccount`

8. In the **Azure Storage Account Access Key** field, enter your Azure access key. For information on Azure storage accounts, see [Create a storage account](#).
9. In the **Bucket URI** field, enter the URI and bucket for your Microsoft Azure bucket. For example:

`https://myaccount.blob.core.windows.net/mycontainer`

10. Click **Next**. The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

Note

The display area is blank when we create a new cloud storage location.

11. Click **Create** to create the cloud storage location.

Create a Google Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.

2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

```
My_Cloud_Store
```

3. (Optional) In the **Description** field, enter a description for the link. For example:

```
My cloud storage link.
```

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard.

In the Credentials section, select **Cloud Username and Password**.

5. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

```
my_credential
```

6. From the **Cloud Store** drop-down list, select **Google**.

7. In the **HMAC Access Key** field, enter your HMAC access ID. For example:

```
GOOGTS1C3LPB3KTKSDB2BFD
```

8. In the **HMAC Access Secret** field, enter your HMAC secret. For information on HMAC keys, see [HMAC Keys](#).

9. In the Storage Settings tab of the Add Cloud Store dialog box, enter a name for the cloud storage link. For example:

```
My_Cloud_Store
```

10. (Optional) In the **Description** field, enter a description for the link. For example:

```
My cloud storage link.
```

11. In the **Bucket URI** field, enter the bucket and URI for your Google bucket. For example:

```
https://my_bucket.storage.googleapis.com
```

12. Click **Next**.

The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

Note

The display area is blank when we create a new cloud storage location.

13. Click **Create** to create the cloud storage location.

Create an Other (Swift Compatible) Cloud Store Location

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.
2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

My_Cloud_Store

3. (Optional) In the **Description** field, enter a description for the link. For example:

My cloud storage link.

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard. In the Credentials section, select **Cloud Username and Password**.

Note

If you have the user OCID, tenancy OCID, private key and fingerprint, select **Oracle Cloud Infrastructure Signing Keys** and refer to the *Create an OCI Cloud storage location using Oracle Cloud Infrastructure Signing Keys* section of this topic.

5. From the Cloud Store drop-down list, select **Other (Swift Compatible)**.
6. Enter a name in the **Credential Name** field. The name must conform to Oracle object naming conventions. For example:

my_credential

7. In the **Access User Name** field, enter your access user name. For example:

OTHER_KEY123...

8. In the **Access Key** field, enter your access key.
9. In the **Bucket URI** field, enter the URI and bucket for your cloud store bucket. For example:

https://someswiftcompatibleprovider.com/my_bucket

10. Click **Next**. The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area. The display area is blank when we create a new cloud storage location.
11. Click **Create** to create the cloud storage location.

Create an OCI Cloud storage location using Oracle Cloud Infrastructure Signing Keys

To create an OCI Cloud storage location using Oracle Cloud Infrastructure Signing Keys:

1. On the **Connections** page, click **Create** and select **Create New Cloud Store Location**. This opens the Add Cloud Store Location wizard.

2. In the Storage Settings tab of the Add Cloud Store Location box, enter a name for the cloud storage link. For example:

My_Cloud_Store

3. (Optional) In the **Description** field, enter a description for the link. For example:

My cloud storage link.

4. Click **Select Credential** and **Create Credential** to create a credential. This opens a Create Credential wizard.
If you have the user OCID, tenancy OCID, private key and fingerprint, select **Oracle Cloud Infrastructure Signing Keys**.

Note

If you only have a username and password select **Cloud Username and Password** in this step and refer to the *Create an Other (Swift Compatible) Cloud Store Location* section of this topic.

5. Specify the following information about your OCI account:

Credential Name: Specify a name to identify the credentials. See [Create an OCI Credential Object](#) to enter the Credential name to specify the credential name.

6. **Fingerprint:** The fingerprint of the RSA key pair. See [Create an OCI Credential Object](#) to enter the Credential name to enter the fingerprint.
7. **Private Key:** The unencrypted private key in the RSA key pair. This should not be encrypted by using any passphrase. See [Create an OCI Credential Object](#) to enter the Credential name to enter the private key.
8. **Oracle Cloud Infrastructure Tenancy:** The OCID of the tenant. See [Where to Get the Tenancy's OCID and User's OCID](#) for details on obtaining the Tenancy's OCID.
9. **Oracle Cloud Infrastructure User Name:** The OCID of the user. See [Where to Get the Tenancy's OCID and User's OCID](#) for details on obtaining the User's OCID.
10. Select **Create Credential**.
11. In the **Bucket URI** field, enter the URI and bucket for your cloud store bucket. For example:

`https://objectstorage.<region>.oraclecloud.com/n/<namespace>/b/<bucket>/`

12. Click **Next**.
13. The dialog box progresses to the Cloud Data tab. This tab lists the objects available on this cloud storage location in the display area.

Note

The display area is blank when we create a new cloud storage location.

14. Click **Create**.

You will receive a notification that the cloud storage location is created successfully.

Create Credentials using OCI Cloud Shell

You can create OCI native credentials using a Cloud Shell script.

To create credentials using OCI Cloud shell, you must run `adb-create-cred.sh` to generate credential scripts. The `adb-create-cred.sh` script searches for existing credentials, if found you are asked if you want to reuse them or if you want new credentials created. Depending on your decision, the generated credential scripts include new or existing credentials. Download the OCI Native Credential script or copy it to run it directly in your database using any SQL or JSON tool or utility.

Note

For a list of arguments supported by the Create Credential script, enter `adb-create-cred.sh --help`.

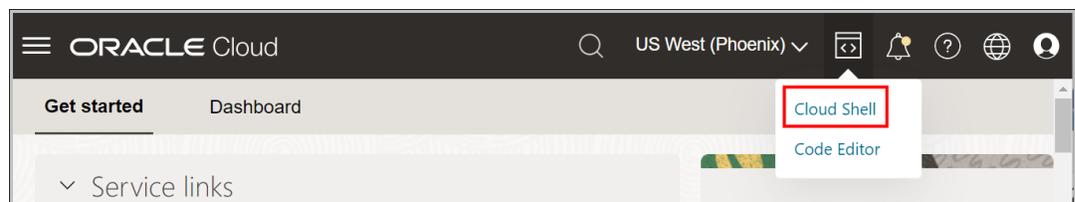
In this example, OCI Native Credential scripts are generated in the script area for your tenancy without running them in your database.

1. On the **Connections** page, click **Create** and select **New Cloud Store Location**. You need to know the Name and Cloud Store Location of the Credential to create a new one.
2. Select **Create Credential**.
3. On the Create Credential wizard, select **Create using OCI Cloud Shell**. You will view a script in the script editor field.

Note

- If you are in your home region and the Autonomous AI Database is in your home region, run the script without the region argument. The script assumes the Autonomous AI Database is in your home region and will not prompt for an alternate region.
- If you are in your home region and your Autonomous AI Database is in an alternate region, then you must pass in the `--region` argument when running the script.

4. Click **Copy Script**.
5. Sign into your OCI Tenancy, select the **Developer Tools** icon and click **Cloud Shell** from the drop down list.



6. Paste the `adb-create-cred.sh` script you copied in the previous step that includes arguments the tool uses to locate the database, connect to the database and the credential name you provide in the **Credential Name** field.

```

Cloud Shell

[redacted]@cloudshell:~ (us-ashburn-1)$ ./adb-create-cred.sh \
> --region=us-ashburn-1 \
> --compartment=[redacted] \
> --database=[redacted] \
> --username=ADMIN \
> --credential=MYOCICRED

```

- The Cloud Shell script informs you of existing OCI API Keys and Fingerprint and if you want to reuse them. Enter `y` to reuse the existing keys and fingerprint.

```

You specified the Region [us-ashburn-1]

You specified the Compartment [ [redacted] ]

You named the OCI Native Credential [MYOCICRED]

You specified the Database [ [redacted] ]

#####
# This initial step ensures your OCI API Keys and Fingerprint are setup properly. #
#####

You have an existing API Key and Fingerprint! Do you want to try to reuse them? (y/n):

```

If you choose to create new credentials, and decide after overwriting the credentials that you want to use the overwritten credentials, you can go back to the directory where the backup is and reuse them. The backup file name or folder has a suffix with this syntax: `_bkp_YYYYMMDD_abc`. For example, this backup file was created on June 06, 2024: `_bkp_20240603_woT`.

- The Cloud Shell script window displays if you want to run the credential scripts on your Autonomous AI Database. Enter `y` to run the credential script in the database with the region, compartment and database arguments.

```

Your API Key was NOT removed, it was found in your OCI Profile. Let's use it!

Your Native Credential SQL file [ $HOME/oci_native_credential.sql ] was created.

Your Native Credential JSON file [ $HOME/oci_native_credential.json ] was created.

#####

Proceed to run the Credential Scripts on your Autonomous Database? (y/n): y

```

- You can view a notification that informs you that you do not have a wallet file. The script creates a wallet file. You must specify the ADMIN password you use to connect to the database. In case the script detects the wallet file, the script prompts you to either reuse it or set up a new one.

```

You do not have a Wallet File. Let's set it up for you.

Fixing path in [~/.oracle/wallet/sqlnet.ora] to that of your wallet File.

#####
# We setup your credentials, generated SQL scripts to add the credentials, we #
# also setup your Autonomous Database connection, and are now connecting you to #
# your Autonomous Database so you can execute the credential scripts. #
#####

You specified the Database Username [ADMIN]

Enter Autonomous Database password:

```

Note

If you have an Autonomous AI Database private endpoint in a Virtual Cloud Network, the `adb-create-cred.sh` script generates the SQL and JSON scripts. However, it does not perform the steps required to access the Autonomous AI Database private end point. Instead, it prompts you to execute `cat ~/oci_native_credential.sql` to copy the SQL into whatever SQL tool you have access to via a Bastion or Jump Host.

- The Cloud Shell script displays a notification of successful login and creation of credentials in the specific database.

The script prompts you if you want to run it on **another** database. Enter `n` to not run credentials in another database.

The script exits and displays the path to run the `oci_native_credential.sql` script to create a credential.

```

Login successful.

OCI Native Credential [MYOCICRED] creation successful.

#####
# Please be aware that it may take up 15 minutes to propagate your credentials to #
# OCI Services. During this period of time your new credentials will not work. #
#####

Proceed to run the Credential Scripts on another Autonomous Database? (y/n): n

Exiting...

To view PL/SQL script for creating credentials,

execute: cat $HOME/oci_native_credential.sql

~@cloudshell:~ (us-ashburn-1)$

```

Note

If the connection to the Autonomous AI Database is unsuccessful, you can run `cat ~/oci_native_credential.sql`, and copy the SQL to run it directly in your database using any SQL tool.

Create Oracle Cloud Infrastructure Native Credentials

To establish cloud storage connection from Data Studio to Oracle Cloud Infrastructure (OCI) Object storage service, you must configure the cloud storage location with your OCI authentication details. You can create Oracle Cloud Infrastructure (OCI) Native Credentials by using the `CREATE_CREDENTIAL` procedure of `DBMS_CLOUD` package.

Create an Oracle Cloud Infrastructure (OCI) Credential Object

To access Object Storage, you must have credentials that you can create via the `CREATE_CREDENTIAL` procedure of `DBMS_CLOUD` package. `DBMS_CLOUD` supports creation of credential objects that contains OCI native authentication. The `DBMS_CLOUD` procedure stores cloud service credentials in Autonomous Database.

The `DBMS_CLOUD.CREATE_CREDENTIAL` procedure is overloaded with the Oracle Cloud Infrastructure-related parameters, including: `user_ocid`, `tenancy_ocid`, `private_key`, and `fingerprint`. This is for using Oracle Cloud Infrastructure Signing Keys authentication.

Let us create an OCI native authentication credential when creating an object store credential object. In the OCI native authentication, the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure includes the following parameters:

Table 12-1 DBMS_CLOUD.CREATE_CREDENTIAL parameter descriptions

Parameter	Description
<code>credential_name</code>	The <code>credential_name</code> parameter must conform to Oracle object naming conventions.
<code>user_ocid</code>	Specifies the user's OCID. See Where to Get the Tenancy's OCID and User's OCID for details on obtaining the User's OCID.
<code>tenancy_ocid</code>	Specifies the tenancy's OCID. See Where to Get the Tenancy's OCID and User's OCID for details on obtaining the Tenancy's OCID.
<code>private_key</code>	Specifies the generated private key. Private keys generated with a passphrase are not supported. You need to generate the private key without a passphrase. See How to Generate an API Signing Key for details on generating a key pair in PEM format.
<code>fingerprint</code>	Specifies a fingerprint. After a generated public key is uploaded to the user's account the fingerprint is displayed in the console. Use the displayed fingerprint for this argument. See How to Get the Key's Fingerprint and How to Generate an API Signing Key for more details.

Here is the syntax of the DBMS_CLOUD.CREATE_CREDENTIAL procedure:

```
DBMS_CLOUD.CREATE_CREDENTIAL (
    credential_name IN VARCHAR2,
    user_ocid       IN VARCHAR2,
    tenancy_ocid   IN VARCHAR2,
    private_key    IN VARCHAR2,
    fingerprint    IN VARCHAR2);
```

Once you obtain all the necessary inputs and generate your private key, here is a sample of your CREATE_CREDENTIAL procedure:

```
BEGIN DBMS_CLOUD.CREATE_CREDENTIAL ( credential_name =>
    'OCI_NATIVE_CRED', user_ocid =>
    'ocid1.user.oc1..aaaaaaatfn77fe3fxux3o51lego7glqjejrzsqsrs64f4jsjrhbsk5qzndq',
    tenancy_ocid =>
    'ocid1.tenancy.oc1..aaaaaaapwfkqz3upqklvmelbm3j77nn3y7uqmlsod75rea5zmtmb1574v
e6a', private_key =>
    'MIIEogIBAAKCAQEAsbnPOYEKxM5h0DF+qXmie6ddo95BhlSMSIxRRS01JEMPeSta0C7WEg7g8SOSz
hIroCkgOqDzkcyXnk4Bl0dn5Wm/BYpdAtTxk0sln2DH/GCH7l9P8xC9cvFtacXkQPMAXIBDv/
zwG1kZQ7Hvl7Vet2UwwuhCsesFgZzRAHkv4cqqE3uF5p/
qHfzZHoevdq4EAV6dZK4Iv9upACgQH5zf9IvGt2PgQnuEFrOm0ctzW0v9JVRjKnaAYgAbqa23j8tKa
pgPuREkfsZv2UMgF7Z7oJYMJEuzGseNULsXn6N8qcvr4fhuKtOD4t6vbIonMPIm7Z/
a6tPaISUFv5ASYzYEUwIDAQABAoIBACaHnIv5ZoGNxkOgF7ijeQmatoELdeWse2ZXll+JaINeTwKU1
fIB1cTAmSFv9yrbYb4ubKCJuYZJeC6I92rT6gEiNpr670Pn5n43cwblszcTryWOYQVxAcLkejbPA7j
Zd6CW5xm/vEgRv5qgADVcZDCzriJ0t1Fghicc+EJ4BFvOetnzEuSidnFo07K3tHGbpG+DPN5qrO/
8NmrBebqezGkOuOVkOA64mp467DQUhpAvsy23RjBQ9iTuRktDB4g9cOdOVFouTZTnevN6JmDxufu9L
ov2yvVMkUC2YKd+RrTAE8cvRrn1A7XKkH+323hNC59726jT57JvZ+ricRixSECgYEA508e/
alxHUIAU9J/uq98nJY/
6+GpI9OCZDkEdBexNpKeDq2dfAo9pejFKYjH8ERj9quA7vhHEwFL33wk2D24XdZl6vq0tZADNSzOtT
rtSgHykvzcn7nXv2fBWAPIN59s9/oEKIOdkMis9fps1mFPFIn8ro4ydUWuR7B2nM2FWkCgYEAxKs/
zOIbrzVlHEVgSH2NJvJqs24S8w+99uLQK2Y06R59L0Sa90QHNCdjB1MaKlanAahP30l0am0SB450ke
iUD6BtuNHH8EIXGL4vX/SYE/
AF6tw3DqcOYbLPpN4CxIITF0PLCRoHKxARMZLcJBTMGpxdmTNGyQAPWXNSrYEFsCgYBp0sHr7TxJ1
Wt07gvvvd91yCugYBJAyMBr18YY0soJnJRHL67A/
hlk8FYGjLW0oMlVBtduQrTQBGVQjedEsepbrAcC+z7+b3yfMb6MStE2BmLPdF32XtCH1bOTJSqFe8
FmEWUv3ozxguTUam/
fq9vAndFaNre2i08sRfi7wfmQKBgBrzcNHN5odTIV8l9rTYZ8BHdIoyOmxVqM2tdWONJREROYyBtU7
PRsFxBEubqskLhsVmYFO0CDORZlgbwIOJPqkJjh+2t9SH7Zx7a5iV7QZJS5WeFLMUEv+YbYAjnXK+d
OnPQtKhOblQwCEY3Hsblj7Xz7o=', fingerprint =>
    '4f:0c:d6:b7:f2:43:3c:08:df:62:e3:b2:27:2e:3c:7a');END;
PL/SQL procedure successfully completed.
```

You can now retrieve the new credentials with the following query:

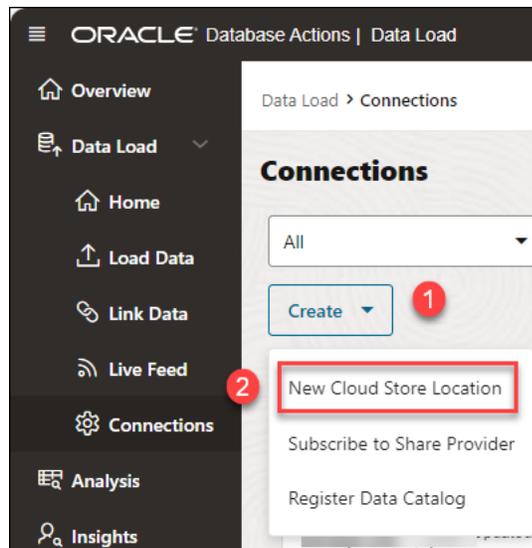
```
SELECT owner, credential_name FROM dba_credentials WHERE credential_name LIKE
'%NATIVE%'; OWNER CREDENTIAL_NAME-----
ADMIN OCI_NATIVE_CRED
```

Create Amazon Web Services (AWS) Credentials

To access AWS Glue Catalog you must have AWS credentials that you can create via the Create Credential wizard. This wizard stores cloud service credentials in the Autonomous AI Database.

Use the `CREATE CREDENTIAL` wizard depending on where your source files reside. To create AWS credentials:

1. On the **Connections** page, click **Create** and select **New Cloud Store Location**.



2. Select **Create Credential** from **Add Cloud Store Location** wizard.

The screenshot shows a 'Create Credential' dialog box with the following elements:

- Radio buttons: Cloud Username and Password, Oracle Cloud Infrastructure Signing Keys
- Cloud Store: Amazon S3 (selected in a dropdown menu)
- Credential Name: * (empty text field)
- AWS access key ID: * (empty text field)
- AWS secret access key: * (empty text field)
- Buttons: ? (help), Create Credential (highlighted with a red box), Cancel

3. Select **Cloud Username and Password**.
4. Select **Amazon S3** from the Cloud Store drop-down.
5. Enter a name in the Credential Name field. The name must conform to Oracle object naming conventions. For example, MY_AWS_CRED.
6. In the **AWS Access Key ID** field, enter your AWS access key ID. For example: myAccessKeyID, see [Managing access keys for IAM users](#).
7. In the **AWS Secret Access Key** field, enter your AWS secret access key. For information on AWS access keys, see [Managing access keys for IAM users](#).
8. Click **Create Credential** to create AWS credentials.

Manage Resource Principal with DBMS_CLOUD

The Oracle Cloud Infrastructure (OCI) Resource Principal allows applications and services running within the OCI to access an Autonomous Database without the need for traditional database credentials (e.g., username and password).

You do not need to create a credential object when you use a resource principal. The Autonomous Database creates and secures the resource principal credentials you use to access the specified Oracle Cloud Infrastructure resources. A resource principal consists of a temporary session token and secure credentials that enable the database to authenticate itself to other Oracle Cloud Infrastructure services.

Prerequisites to Use Resource Principal with Autonomous Database:

- There are several steps required to set up a resource principal on an Autonomous Database: You must create and define Oracle Cloud Infrastructure Identity and Access Management (IAM) policies.

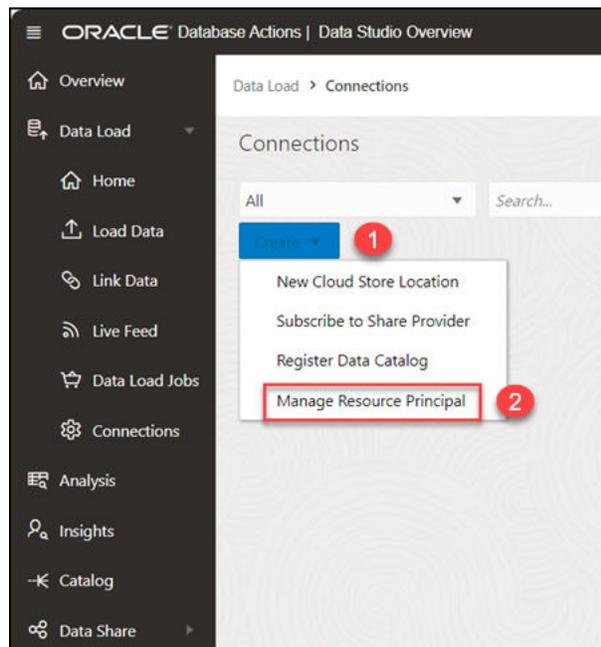
- Enable the resource principal for the ADMIN user, and optionally enable the resource principal for a database user.

When you authenticate using a resource principal, you do not need to create and manage credentials to access OCI resources. The Autonomous Database makes the resource principal available to you and secures the resource principal for you.

Manage Resource Principal using Data Studio

You can also enable Resource Principal using Data Studio.

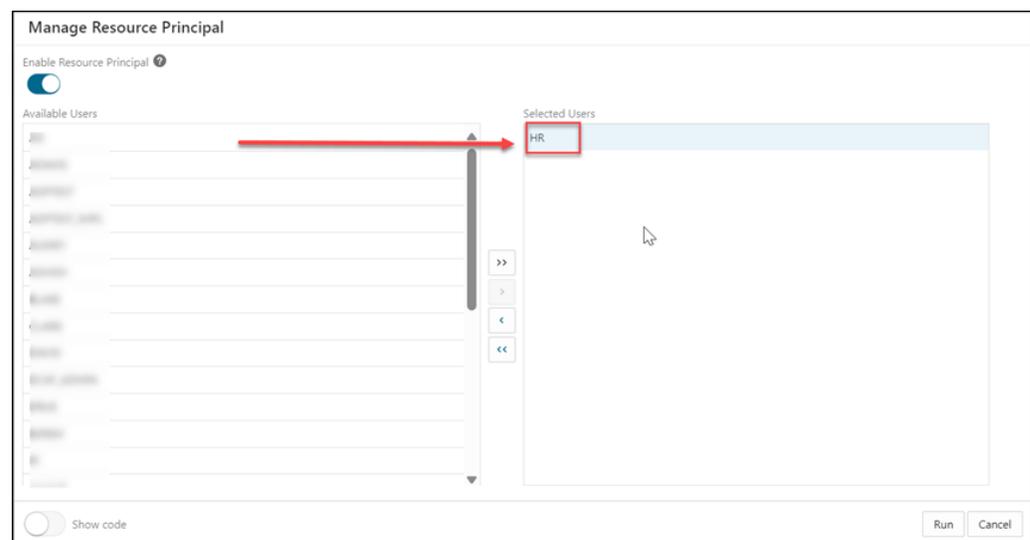
1. From the Database Actions Launchpad, select the **Data Load** tile.
2. Click **Connections** This opens the Connections page.
3. On the **Connections** page, click **Create** and select **Manage Resource Principal**.



This opens the Manage Resource Principal dialog.



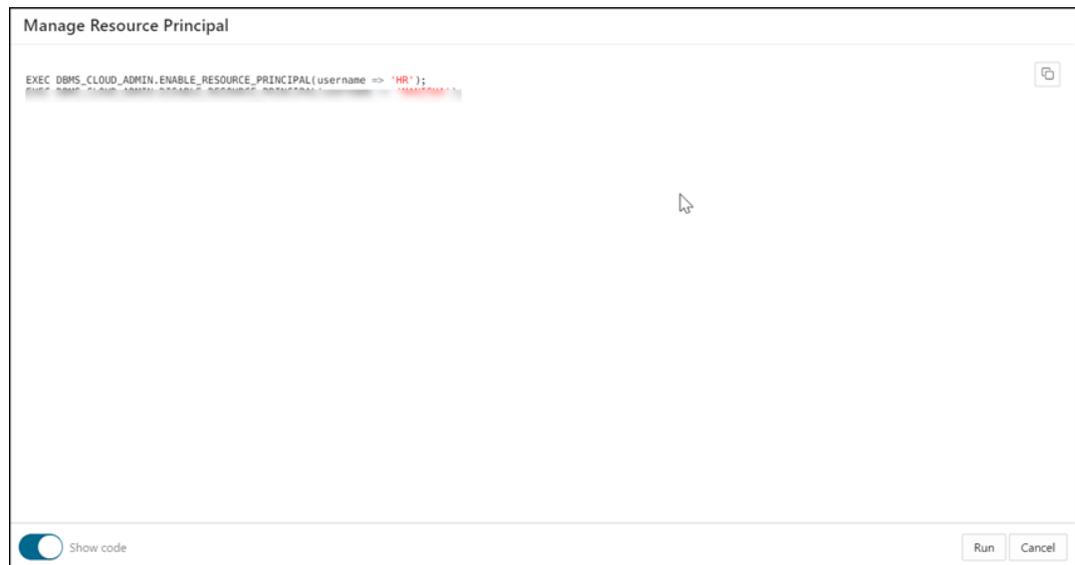
4. Select the user or users you want to add from the **Available Users** list to the **Selected Users** list to enable their resource principal. Choose any of the available options:
 - >: This option enables you to move the user to **Selected Users**.
 - <: To remove the selected user from **Selected Users**, select this option.
 - >>: This option allows you to move all the tables to the **Selected Users** list.
 - <<: To remove all the selected users from **Selected Users**, select this option.



Note

You can grant access to the resource principal credential to a database user only if the ADMIN user has enabled the resource principal credential.

5. Select the **Show Code** option to view the PL/SQL code equivalent of the Manage Resource Principal dialog box. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Enable Resource Principal** in the Manage Resource Principal dialog box.



6. Click **Run** to complete the process of enabling the resource principal of the selected users. A confirmation notification is displayed that confirms that the resource principal of selected users is enabled.
7. Select a user from Selected Users to Available Users to **disable** the resource principal for the selected user. This removes the credential `OCI$RESOURCE_PRINCIPAL`.

Register Data Catalog

You can register data catalogs you want to use with registering the data catalog.

Register OCI Data Catalog

To register a data catalog you need to specify the details of the credentials you want to register your data source. A credential object manages a data catalog instance. In OCI native authentication, the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure includes these parameters: `credential_name`, `user_ocid`, `tenancy_ocid`, `private_key`, and `fingerprint`.

See REF `DBMS_DCAT` Package to refer to the procedures to add custom parameters such as region and data catalog ID to the Data Catalog. The Data Catalog ID is a unique Oracle cloud Identifier for data catalog instance and region is the data catalog region.

To register Data catalog:

1. From Connections page, click **Create** and select **Register Data Catalog**. This opens Register Data Catalog wizard.
2. In the Catalog Settings tab, specify the following details:
 - **Catalog Name:** `Test`. Enter a name of your choice.
 - **Description:** Specify a description. This is an optional field.
 - Under Data Catalog details, fill in the following field values:

- **Credential for Data Catalog Connection:** Select a value from the drop-down. The drop-down lists the values of the credentials you create. If you do not have credentials, you can create one. Refer to [Create Credentials](#) for more information.
 - **Region:** us-ashburn-1. Enter the region name you use while you create the credentials using the DBMS_DCAT package.
 - **Data Catalog ID:**
ocidl.datacatalog.oc1.iad.amaaaaaa7ratczyayxh7uzl124cp3uwzsugfj7qlubak77toiehidpsqsygq. Enter the data catalog ID from the DBMS_DCAT package.
 - Select the **Register Data Catalog Connection** button to register the data catalog within the Autonomous AI Database. You can view this option when you select **OCI Data Catalog** from Catalog Type.
 - Select the **Use separate credential** for object Storage to select the database credentials from the drop-down. You can view this option when you select **OCI Data Catalog** from Catalog Type.
3. Click **Next** to progress to the Register Assets tab. This tab creates a connection with the source Data Catalog objects you select from the list of objects.
 4. After successful registering of the data catalog objects, click **Create**.
 5. You will receive a notification that the data catalog is created successfully. After successful creation of the data catalog, you can view the data catalog entity in the list of entities in the **Connections** page.

Register AWS Glue Catalog

You can integrate Oracle Data Catalog with Amazon's Glue data catalog.

To access the AWS Glue Catalog, register the catalog on the **Connections** page.

This enables you to synchronize Data Catalog metadata with AWS Glue and query data stored in S3 from an Autonomous AI Database without manually deriving the schema for the external data sources and creating external tables.

Refer to [Query External Data with Glue Data Catalog](#) Query External Data with Glue Data Catalog for more details.

You also need an AWS Credential to be associated with the AWS Glue Catalog. The tool initiates the connection by specifying the following field values:

- Cloud Store
- Credential Name
- AWS access key ID
- AWS Secret Access Key

To register the Data Catalog:

1. From the Connections page, click **Create** and select **Register Data Catalog**. This opens the Register Data Catalog wizard.
2. In the Catalog Settings tab, specify the following details:
 - **Catalog Name:** MY_GLUE_CATALOG
 - **Description:** Specify a description. This is an optional field.
 - Under Data Catalog Details, fill in the following field values:
 - **Catalog Type:** Select AWS Catalog from the drop-down.

- **Credential for Data Catalog Connection:** Select a credential value from the drop down. The drop-down lists the values of the credentials you create. If you do not have credentials, you can create one. Refer to [Create Amazon Web Services \(AWS\) Credentials](#) for more information.
- **Region:** eu-west-2. Enter the region name you use while you create the AWS credentials.

The screenshot shows the 'Register Data Catalog' interface. At the top, there are two progress indicators: '1' for 'Catalog Settings' and '2' for 'Register Assets'. The 'Catalog Settings' section contains the following fields:

- Catalog Name ***: MY_GLUE_CATALOG
- Description**: (empty)
- Data Catalog Details** (with a help icon):
 - Catalog Type**: AWS Glue
 - Credential for Data Catalog Connection ***: ADPAWSGLUE_CRED (AKIAIYNYFIU3B2LWCHGG3) with a 'Create Credential' button.
 - Region ***: eu-west-2
 - Register Data Catalog Connection**: A disabled toggle switch.

At the bottom right, there are four buttons: 'Back', 'Next' (highlighted with a red box), 'Create', and 'Cancel'.

The **Register Data Catalog Connection** field will be greyed out since you selected AWS Catalog. You can select this field only if you want to register data catalog connection within the Autonomous AI Database (i.e., when you select OCI Data Catalog).

3. Click **Next** to progress to the Register Assets tab. The Register Assets tab creates a connection with the source data catalog objects you select from the list of objects.
4. After successfully registering the data catalog objects, click **Create**.

The screenshot shows the 'Register Data Catalog' interface at Step 2: 'Register Assets'. The progress indicators show '1' for 'Catalog Settings' and '2' for 'Register Assets'. The 'Register Assets' section contains the following:

- Select Source Data Catalog Assets**: A tree view showing the following assets:
 - Catalog (us-west-1)
 - spapadom-iceberg-db
 - spapadom-iceberg-tpcds
 - spapadom-iceberg-tpch

At the bottom right, there are four buttons: 'Back', 'Next', 'Create' (highlighted with a red box), and 'Cancel'.

5. You will receive a notification that the data catalog has been created successfully. After the successful creation of the data catalog, you can view the data catalog entity in the list of entities on the **Connections** page.

Loading Data

You can load data from files on your local device, from remote databases, or from cloud storage buckets from directories and share providers. The file formats that you can load are CSV, XLS, XLSX, TSV, TXT, XML, JSON, GEOJSON, AVRO, PARQUET, GZ, GZIP, ZIP, PDF, PNG, JPG, JPEG and TIFF.

The Data Load tool supports the loading of source files with boolean format. The data type boolean has the truth values `TRUE` and `FALSE`. If there is no `NOT NULL` constraint, the boolean data type also supports the truth value `UNKNOWN` as the null value. You can use the boolean data type wherever the datatype appears in Oracle SQL syntax.

It also supports the `VECTOR` data type, which allows you to store vector embeddings directly within Oracle Database tables.

The following topics describe the interfaces for these actions.

- [Loading Data From Local Files](#)
- [Loading Data from Other Databases](#)
- [Loading Data from Cloud Storage](#)
- [Loading from File System](#)
- [Loading Data from AI Source](#)
- [Loading from Share Providers](#)
- [Use OCI Language Service Capabilities in Data Studio](#)
You can utilize OCI Language Service Capabilities such as Sentiment Analysis, Key Phrase Extraction and Language Detection to analyze data without machine learning (ML) or artificial intelligence (AI) expertise.
- [Use OCI Document Understanding to extract Tables from Images](#)
You can utilize Oracle Cloud Infrastructure(OCI) Document Understanding Capabilities such as Table Extraction to detect tables from the data you load.
- [Use GeoJSON in Data Load](#)
A GeoJSON object accommodates information about the specific geometry (e.g. Point, Line, Polygon, etc.) along with optional metadata (e.g. ID, etc.).
- [Loading Data From Local Files](#)
To load data from local files into your Oracle Autonomous AI Database, on the Data Load page, select **LOAD DATA** and **LOCAL FILE**.
- [Loading Data from Other Databases](#)
To load data from tables in another database into your Oracle Autonomous AI Database, on the Data Load page, select **LOAD DATA** and **DATABASE**. Select a database link from the drop-down list. Drag one or more tables from the list of database tables and drop them in the Data Load Cart.
- [Loading Data from Cloud Storage](#)
You can load data from a cloud store to a table in your Autonomous AI Database.
- [Load Apache Iceberg Tables](#)
Data Studio supports loading and linking Iceberg tables from the object store.

- [Loading Data from File System](#)
You can load files from file system directories to your Autonomous Database.
- [Loading Data from AI Source](#)
You can use Data Studio tools to load data from AI source.
- [Loading Data from Share](#)
You can select tables from a Share. You need to subscribe and access provided data share.
- [Create Live Feed from Data Load](#)
The Data load tool loads the data from folders in cloud object stores and enables it to schedule repeated data loads in real time. This is the creation of Live Feed from a data load job.

Use OCI Language Service Capabilities in Data Studio

You can utilize OCI Language Service Capabilities such as Sentiment Analysis, Key Phrase Extraction and Language Detection to analyze data without machine learning (ML) or artificial intelligence (AI) expertise.

For example, you can use it for feedback on a product. A phone manufacturer has launched a new phone model and they want to know what the customer sentiment is on their product. If a large percentage of sentiment is negative it could signal a potential fault with the camera which wasn't detected in Quality Control (QC).

In this section, we will go through the following topics:

- [Overview of Sentiment Analysis and Keyphrase Extraction](#)
- [Parameters to Analyze Data](#)
- [Perform Sentiment Analysis](#)
- [Perform Key Phase Extraction](#)
- [Overview of Sentiment Analysis, Key Phrase Extraction and Language Detection](#)
Sentiment Analysis, Key Phrase Extraction, Language Detection and Text Translation are currently supported in loading data from local files and loading data from cloud storage.
- [Parameters to Analyze Data](#)
When you invoke an **Add Expression** from the Settings tab, you must configure the model using parameters.
- [Perform Sentiment Analysis](#)
To determine the Sentiments of input data:
- [Perform Key Phrase Extraction](#)
To extract Key Phrase information from input data:
- [Perform Language Detection](#)
The Data Studio supports the detection of multiple languages.
- [Perform Text Translation](#)
The Data Studio supports the translation of multiple languages supported by OCI Language Service.

Overview of Sentiment Analysis, Key Phrase Extraction and Language Detection

Sentiment Analysis, Key Phrase Extraction, Language Detection and Text Translation are currently supported in loading data from local files and loading data from cloud storage.

Sentiment Analysis

Sentiment Analysis analyses the text to define your sentiment on a topic or product. The Language service sentiment analysis uses natural language processing (NLP). The Data Studio tool uses the Oracle Cloud Infrastructure (OCI) Language service to analyze and understand the input data. The Data Studio tool dynamically adds new columns to the data load that contains the output of the OCI Language service. You can detect the sentiments of any column of the source data. For example, when searching through a column containing reviews for an application, assume that you want a general opinion about the application. The Data Studio tool performs sentiment analysis on the input data and creates a new expression column defined in the target table that consists of the sentiment.

For more details, refer to [Sentiment Analysis in OCI](#).

Key Phrase Extraction

Key phrase extraction identifies the main concepts in a text. Keyword extraction is the automated process of extracting the words with the most relevance, and expressions from the input text. It helps summarize the content and recognizes the main topics. A word cloud can be generated with key phrases to help visualize key concepts in text comments or feedback. For example, a Movie review could generate a word cloud based on key phrases identified in their comments and might see that people are commenting most frequently about the direction, acting, and cinematography staff.

For more details, refer to [Key Phrase Extraction in OCI](#).

Language Detection

You can utilize OCI Language Service Capabilities such as Language Detection to detect the language of the input text. It returns which natural language the text is in. You can use it to overcome language barriers thus improving communication with people from other countries. You can determine the language of the input text and translate content to different languages. For more details refer to [Language Extraction in OCI](#).

Text Translation

The Text Translation translates input text into any one of the OCI Language service-supported target languages. The Data Studio tool uses the Oracle Cloud Infrastructure (OCI) Language service to analyze and understand the input data. The Data Studio tool dynamically adds new column to the data load that contains the translated input source column. For example, when you need to know what customers are saying about your product in the local market language – French. The Data Studio tool performs language translation on the input data and creates a new expression column defined in the target table that consists of the translated source language in the column.

Before you start:

- **Load Data from Local Files or Cloud storage:** Load the data you wish to analyze into your Oracle Autonomous Database from Local files or Cloud storage. Make sure the data is loaded to the data load cart without any errors. You can view and fix mapping errors from the **Errors** quick filter in the **Load Data from Cloud Store Location**. After you load data into the Data load tool,

- Click the **Settings** icon on the Data Load job cart to review the settings.
- On the Settings pane of the **Load Data from Cloud Store Location** wizard, if there is a mapping error, the mapping grid cell will be highlighted with red to indicate an invalid value that must be fixed.

Load Data from Cloud Store Location emp.csv (665B)

Settings Table Settings Template

Preview Option Merge into Table ? Name EMP_MERGE_CLOUD ?

Table Partition Column None ?

SQL

Properties Hide Properties

Encoding AL32UTF8 - Unicode 9.0 Universal Character Set... Text enclosure " Field delimiter Comma

Column header row ? 1 ? Start processing data at row ? 2 ? Convert invalid values to null ?

Newlines included in data values ?

Mapping ? **2 mappings error(s)** ?

Search... ? Quick Filter: Columns Expressions **Errors** ? Bulk Edit

Include	Merge Key	Source	Target column	Format
<input checked="" type="checkbox"/>	<input type="checkbox"/>	EMPNO	-Select-	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ENAME	-Select-	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	JOB	JOB (VARCHAR2(64))	
<input type="checkbox"/>	<input type="checkbox"/>	MGR	-Select-	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	HIREDATE	HIREDATE (DATE)	DD-MON-RR
<input type="checkbox"/>	<input type="checkbox"/>	SAL	-Select-	
<input type="checkbox"/>	<input type="checkbox"/>	COMMISSION	-Select-	

? Close

- Click the **Errors** quick filter, it shows only the columns with errors.

Load Data from Cloud Store Location emp.csv (665B)

Settings Table Settings Template

Preview Option Merge into Table ? Name EMP_MERGE_CLOUD ?

Table Partition Column None ?

SQL

Properties Hide Properties

Encoding AL32UTF8 - Unicode 9.0 Universal Character Set... Text enclosure " Field delimiter Comma

Column header row ? 1 ? Start processing data at row ? 2 ? Convert invalid values to null ?

Newlines included in data values ?

Mapping ? **2 mappings error(s)** ?

Search... ? Quick Filter: Columns Expressions **Errors** ? Bulk Edit

Include	Merge Key	Source	Target column	Format
<input checked="" type="checkbox"/>	<input type="checkbox"/>	EMPNO	-Select-	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ENAME	-Select-	

? Close

- The tools perform sentiment analysis, Key Phrase extraction and Language Detection while you specify settings for the data load job.

Parameters to Analyze Data

When you invoke an **Add Expression** from the Settings tab, you must configure the model using parameters.

Table 12-2 Parameters for Sentiment Analysis, Key Phrase Extraction, and Language Detection

Parameter	Description
Expression Type	Select any one of the following operations you wish to perform on the input text: Sentiment Analysis, Key Phrase Extraction, and Language Detection.
Input Column	Select the column which you wish to analyze. The input column drop-down only contains columns that Sentiment Analysis, Key Phrase Extraction, and Language Detection support. For Sentiment Analysis, only VARCHAR2, NVARCHAR2, CLOB, or NCLOB target columns will be displayed in the input drop-down.
Target Column	<ul style="list-style-type: none"> • Enter the name of the newly created expression column defined in the target table. • For Sentiment Analysis, this column displays the sentiment of the input column. The different types of sentiments the tool identifies are: <ul style="list-style-type: none"> – Positive – Neutral – Mixed – Negative If the tool cannot determine the sentiment of the input column, it returns NULL in the expression column. • For Key Phrase Extraction, this column displays the key phrases of the input column you select. • For Language Detection, this column displays the language of the input column you select.

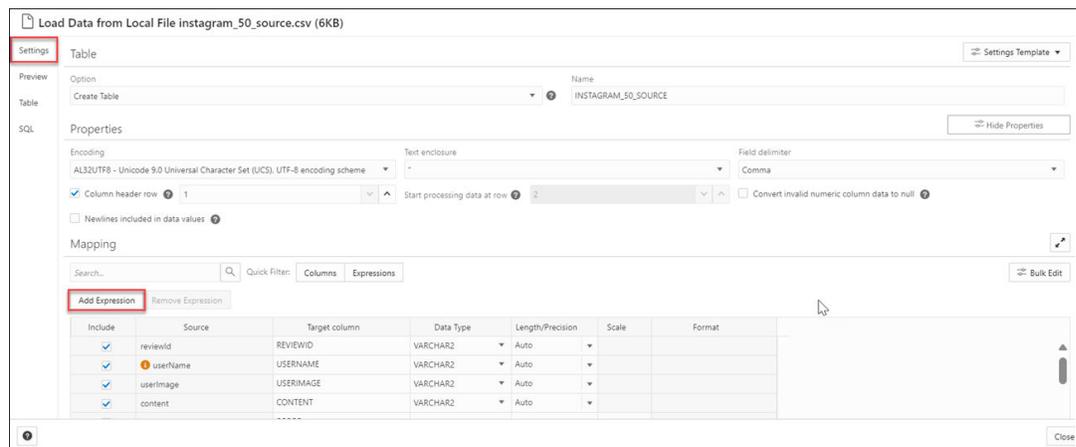
Perform Sentiment Analysis

To determine the Sentiments of input data:

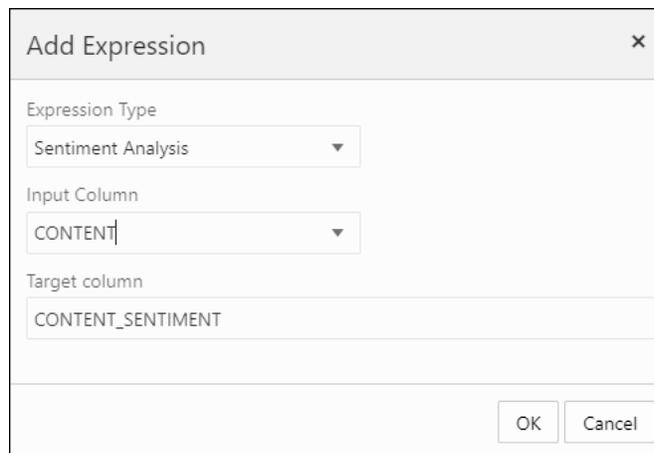
1. After you load data into the Data Load cart from local files or cloud storage, you can view the file in the cart. Click the **settings** icon.



2. Clicking the Setting icon opens a **Load Data from Local File** wizard. In this example, we have loaded data from a local file.
3. On the Settings tab of the wizard, click **Add Expression** under the Mapping section.



This opens the **Add Expression** dialog box.



4. On the **Add Expression** dialog, specify the following fields:
 - **Expression Type:** From the Expression Type drop-down, select **Sentiment Analysis**.
 - **Input Column:** Select the column from the drop-down that you wish to analyze. For example, *CONTENT*.

- **Target column:** Enter the name of the newly created expression column. For example, `CONTENT_SENTIMENT`. Refer to the [Parameters to Analyze Data](#) for more details.
5. Click **OK**. You will see a new row added to the mapping grid. This row determines the output expression column generated by the OCI Language service.

Mapping

Search... Quick Filter: Columns Expressions

Add Expression Remove Expression

Include	Source	Target column	Data Type	Length/Precision	Scale	Format
<input checked="" type="checkbox"/>	at	AT	DATE			YYYY-MM-DD HH24:MI:SS
<input checked="" type="checkbox"/>	replyContent	REPLYCONTENT	VARCHAR2	Auto		
<input type="checkbox"/>	FILENAME	FILENAME	VARCHAR2	AUTO		
<input type="checkbox"/>	SYSTIMESTAMP	UPDATED_TIMESTAMP	TIMESTAMP WITH TIME ZO6			YYYY-MM-DD"THH24:MI:SS.FFTZ
<input checked="" type="checkbox"/>	Sentiment for CONTENT	CONTENT_SENTIMENT	VARCHAR2	40		

6. Click **Close**.
7. Click **Start** in the Data Load menu cart to load data from local files. You will view a confirmation message asking if you wish to start the load from local files.
8. Click **Run** to confirm.

When the data load job completes, the **Table and View Loads** of the Data Load page display the details of the source table that is loaded into the tool. It displays the number of rows and columns and when the upload was complete.

- [Output Data Generated from OCI Sentiment Analysis](#)
When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

Output Data Generated from OCI Sentiment Analysis

When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

To locate the generated expression column, from the Database Actions Launchpad, navigate to **Data Load**. Select the table you load under the **Table and View Loads** section.

Click the three vertical dots beside the load name, and click **Table** then select **View Details**.

This opens the Preview tab of the data load which displays the updated source file. For example, here is an output dataset from sentiment analysis of the Instagram application. Here, `CONTENT` is the target column, and `CONTENT_SENTIMENT` is the sentiment analysis of the input column. This column displays one of the following values such as positive, neutral, mixed, or negative. It displays **Null** when the tool is unable to determine the sentiment.

INSTAGRAM_50_SOURCE

		CONTENT	CONTENT_SENTIMENT
Preview	1	Nice app pppp	Positive
Lineage	2	This is very good app	Positive
Impact	3	Osm	(null)
Statistics	4	Mast app aaye 👍	(null)
Job Report	5	Good	Positive
Data Definition	6	Osm	(null)
	7	Good as a social media	Positive
	8	Good	Positive
	9	Best	Positive
	10	👍👍👍👍👍👍	(null)
	11	Amazing	Positive
	12	I really enjoy using this app	Positive

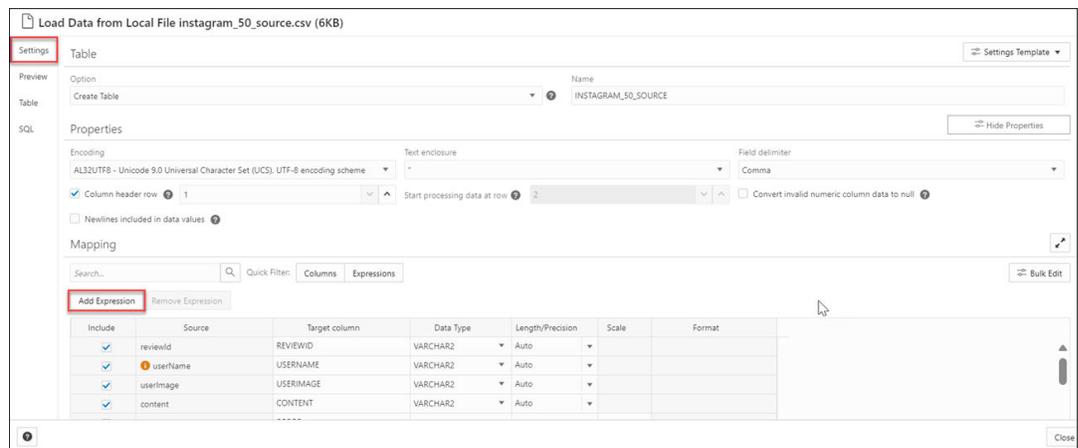
Perform Key Phrase Extraction

To extract Key Phrase information from input data:

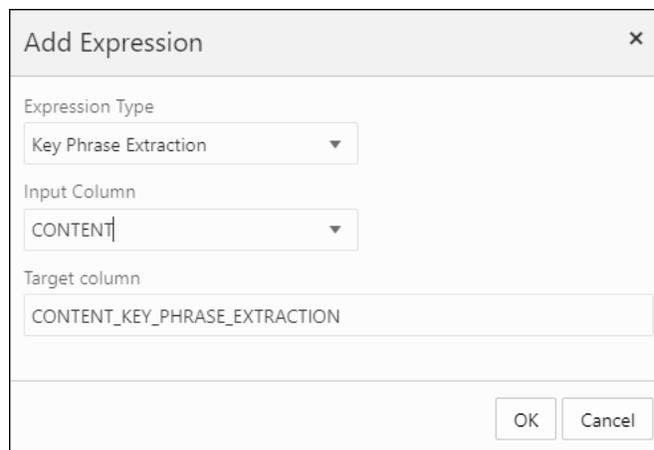
1. After you load data into the Data Load cart from local files or cloud storage, you can view the file in the cart. Click the **settings** icon.



2. Clicking the Setting icon opens a **Load Data from Local File** wizard. In this example, we have loaded data from a local file.
3. On the Settings tab of the wizard, click **Add Expression** under the Mapping section.



This opens the Add Expression dialog box.



4. On the Add Expression dialog, specify the following fields:
 - **Expression Type:** From the Expression Type drop-down, select Key Phrase Extraction.
 - **Input Column:** Select the column from the drop-down that you wish to analyze. For example, *CONTENT*.
 - **Target column:** Enter the name of the newly created expression column. For example, *CONTENT_KEY_PHRASE_EXTRACTION*.

Refer to the [Parameters to Analyze Data](#) for more details.

5. Click **OK**. You will see a new row added to the mapping grid. This row determines the output expression column generated by the OCI Language service.

Mapping

Search... Quick Filter: Columns Expressions

Include	Source	Target column	Data Type	Length/Precision	Scale	Format
<input checked="" type="checkbox"/>	at	AT	DATE			YYYY-MM-DD HH24:MI:SS
<input checked="" type="checkbox"/>	replyContent	REPLYCONTENT	VARCHAR2	Auto		
<input type="checkbox"/>	FILENAME	FILENAME	VARCHAR2	AUTO		
<input type="checkbox"/>	SYSTIMESTAMP	UPDATED_TIMESTAMP	TIMESTAMP WITH TIME ZONE			YYYY-MM-DD"TT"HH24:MI:SS.FFTZ
<input checked="" type="checkbox"/>	Key Phrase Extraction for CON	CONTENT_KEY_PHRASE_EXTRACTI	VARCHAR2	4000		

6. Click **Close**.
7. Click **Start** in the Data Load menu cart to load data from local files. You will view a confirmation message asking if you wish to start the load from local files.
8. Click **Run** to confirm.

When the data load job completes, the **Table and View Loads** of the Data Load page display the details of the source table that is loaded into the tool. It displays the number of rows and columns and when the upload was complete.

- [Output Data Generated from OCI Key Phrase Extraction](#)
When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

Output Data Generated from OCI Key Phrase Extraction

When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

To locate the generated expression column, from the Database Actions Launchpad, navigate to **Data Load**. Select the table you load under the **Table and View Loads** section.

Click the three vertical dots beside the load name, and click **Table** then select **View Details**.

For example, here's an output dataset from sentiment analysis of the Instagram application. Here, **CONTENT** is the target column, and **CONTENT_KEY_PHRASE_EXTRACTION** column displays the key phrases extracted from the input column.

INSTAGRAM_50_SOURCE

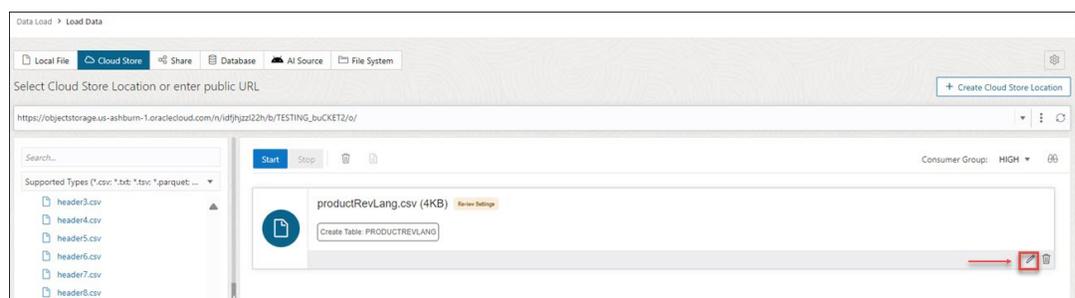
	CONTENT	TENT_KEY_PHRASE_EXTRACTION
Preview		
Lineage	1 Nice app pppp	"nice app pppp"
Impact	2 This is very good app	"good app"
Statistics	3 Osm	"osm"
Job Report	4 Mast app aaye 🍌	"mast"
Data Definition	5 Good	(null)
	6 Osm	"osm"
	7 Good as a social media	"social media"
	8 Good	(null)
	9 Best	(null)
	10 🍌 🍌 🍌 🍌 🍌 🍌 🍌	(null)
	11 Amazing	(null)
	12 I really enjoy using this app	"app"
	13 Good	(null)
	14 Always my first choice. 🍌 🍌	"first choice. 🍌 "
	15 Good	(null)
	16 I love talking to my friends	"friends"
	17 Best social media app in the	"best social media app"

Perform Language Detection

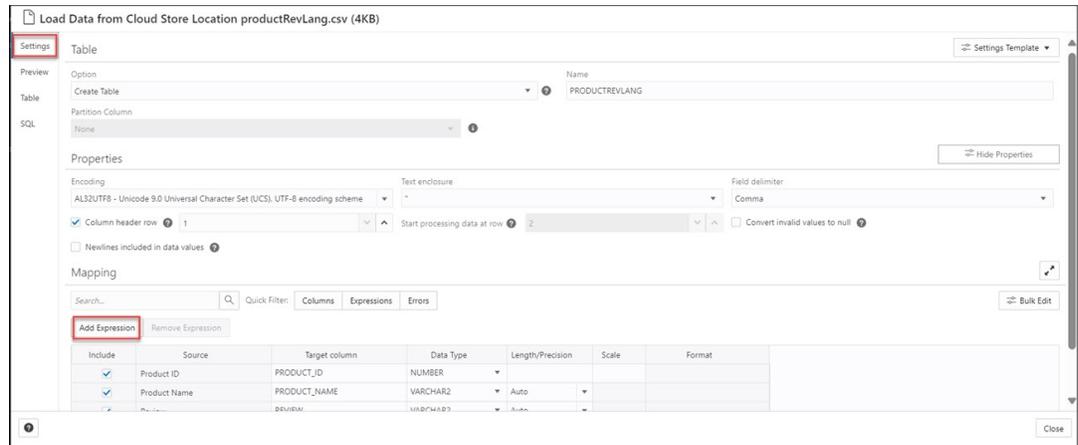
The Data Studio supports the detection of multiple languages.

To determine the language of the input text:

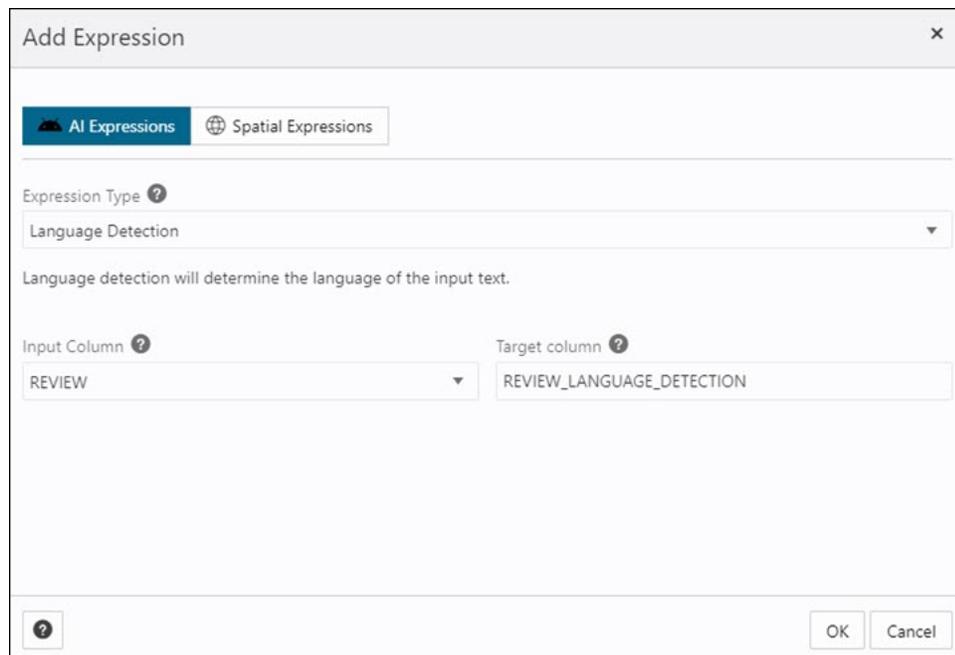
1. After you load data into the Data Load cart from cloud storage, you can view the file in the cart. Click the **Settings** icon.



2. Clicking the Setting icon opens a Load Data from Cloud Store Location wizard.
3. On the Settings tab of the wizard, click **Add Expression** under the Mapping section.



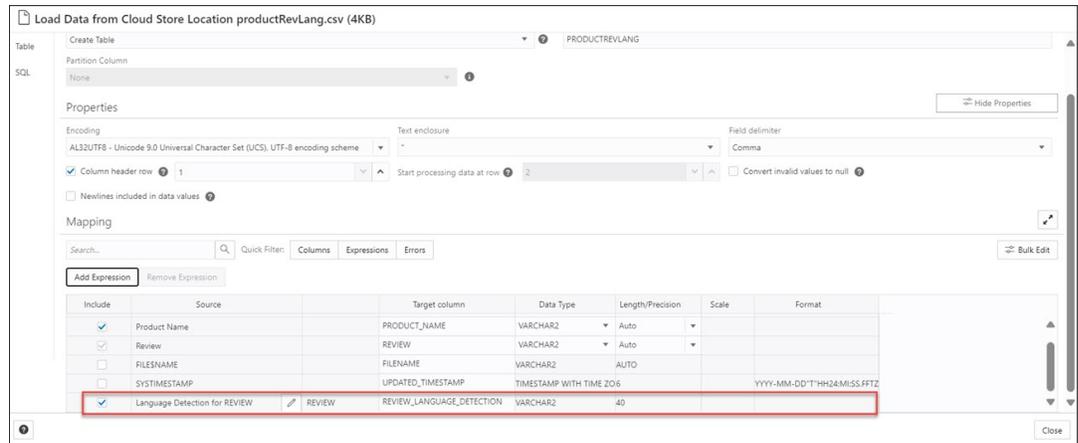
This opens the **Add Expression** dialog box. Select **AI Expressions**.



- On the Add Expression dialog, specify the following fields:
 - Expression Type:** From the Expression Type drop-down, select Language Detection.
 - Input Column:** Select the column from the drop-down that you wish to analyze. In this case, we need to detect the language of the Review column. For example, *REVIEW*.
 - Target column:** Enter the name of the newly created expression column. For example, *REVIEW_LANGUAGE_DETECTION*.

Refer to the [Parameters to Analyze Data](#) for more details.

- Click **OK**. You will see a new row added to the mapping grid. This row determines the output expression column generated by the OCI Language service.



6. Click **Close**.
7. Click **Start** in the Data Load menu cart to load data from local files. You will view a confirmation message asking if you wish to start the load from local files.
8. Click **Run** to confirm.

When the data load job completes, the **Table and View Loads** of the Data Load page display the details of the source table that is loaded into the tool. It displays the number of rows and columns and when the upload was complete.

- [Output Data Generated from OCI Language Detection](#)
When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

Output Data Generated from OCI Language Detection

When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

To locate the generated expression column, from the Database Actions Launchpad, navigate to **Data Load**. Select the table you load under the **Table and View Loads** section.

Click the three vertical dots beside the load name, and click **Table** then select **View Details**.

For example, here's an output dataset from language detection of the Review column. Here, **REVIEW** is the target column, and **REVIEW_LANGUAGE_DETECTION** column displays the language detected from the input column.

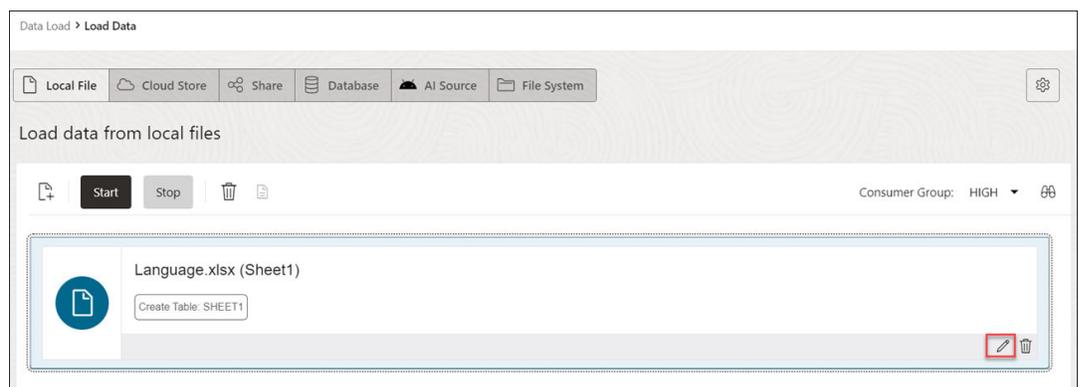
	PRODUCT_ID	PRODUCT_NAME	REVIEW	REVIEW_LANGUAGE_DETECTION
Preview	1	Cookie Crisp Breakfast Cere	Tasty & fresh.	English
Lineage	2	Cookie Crisp Breakfast Cere	القشاش، حرق، تشيب دا وولف	Arabic
Impact	3	Cookie Crisp Breakfast Cere	Mislio sam da bi kutija bila t	English
Statistics	4	Cookie Crisp Breakfast Cere	Meh... forsendelse var en re	Danish
	5	Cookie Crisp Breakfast Cere	Ce que j'aime chez Cookie C	French
Job Report	6	Cookie Crisp Breakfast Cere	In meinen örtlichen Lebensr	German
Data Definition	7	Cookie Crisp Breakfast Cere	έλα για τα μπισκότα τραγα	Greek
	8	Cookie Crisp Breakfast Cere	別の州に住んでいる弊のた!	Japanese
	9	Cookie Crisp Breakfast Cere	Bu misir gevveğinde hanika !	Turkish
	10	Cookie Crisp Breakfast Cere	Это отличная коробка для	Russian
	11	Cookie Crisp Breakfast Cere	Miloval jsem to ! Bylo to vel	Czech
	12	Cookie Crisp Breakfast Cere	Delicioso me llegó intacto. I	Spanish
	13	Cookie Crisp Breakfast Cere	Cookies that are social acce	English
	14	Cookie Crisp Breakfast Cere	After all the years this cerea	English
	15	Cookie Crisp Breakfast Cere	Tasty as ever. Wish the nutri	English
	16	Cookie Crisp Breakfast Cere	The cereal arrived just as it :	English
	17	Cookie Crisp Breakfast Cere	I like these cookies can't go	English
	18	Cookie Crisp Breakfast Cere	They don't taste horrid, but	English
	19	Cookie Crisp Breakfast Cere	Bk have it your way	English
	20	Cookie Crisp Breakfast Cere	Vegan friendly ♥	English

Perform Text Translation

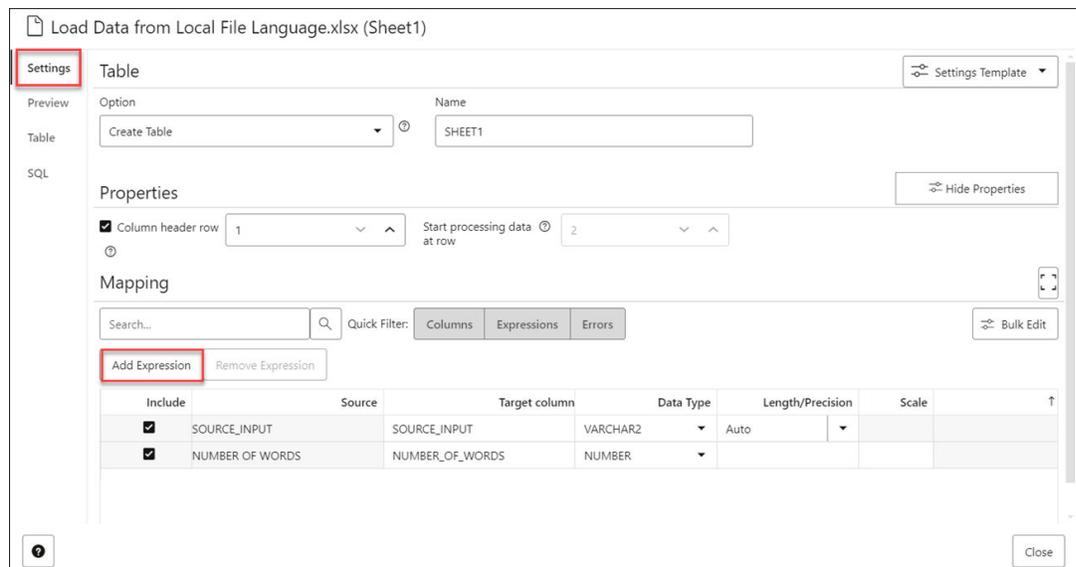
The Data Studio supports the translation of multiple languages supported by OCI Language Service.

To determine the translated input text:

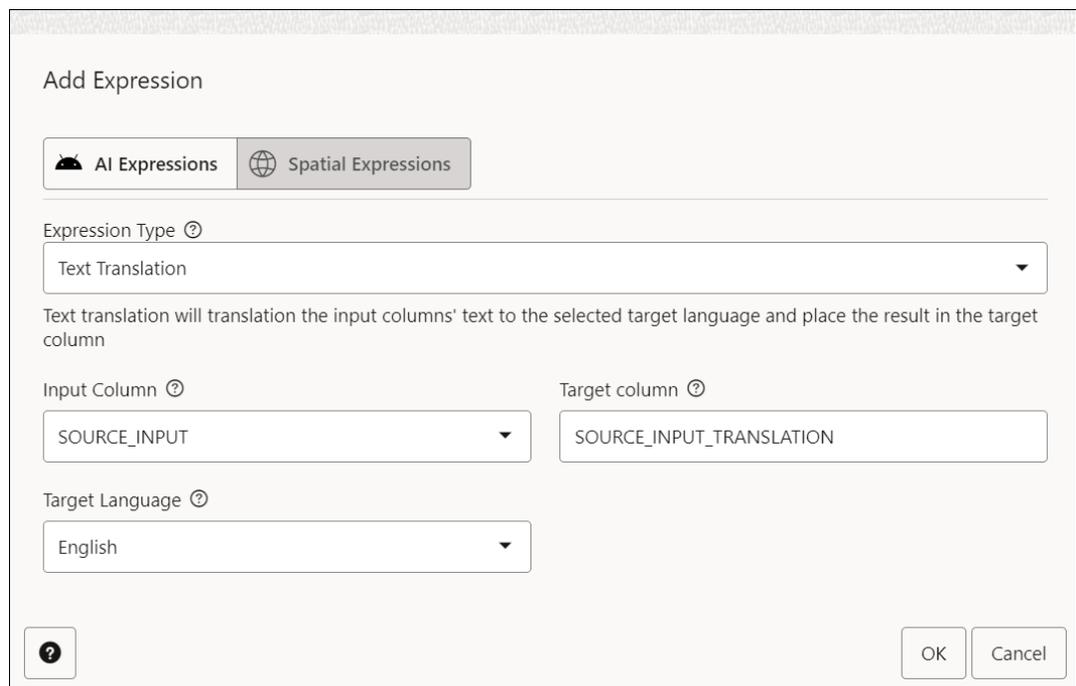
1. After you load data into the Data Load cart from local files, you can view the file in the cart. Click the **settings** icon.



2. Clicking the **Setting** icon opens a Load Data from Local File wizard.
3. On the Settings tab of the wizard, click **Add Expression** under the Mapping section.



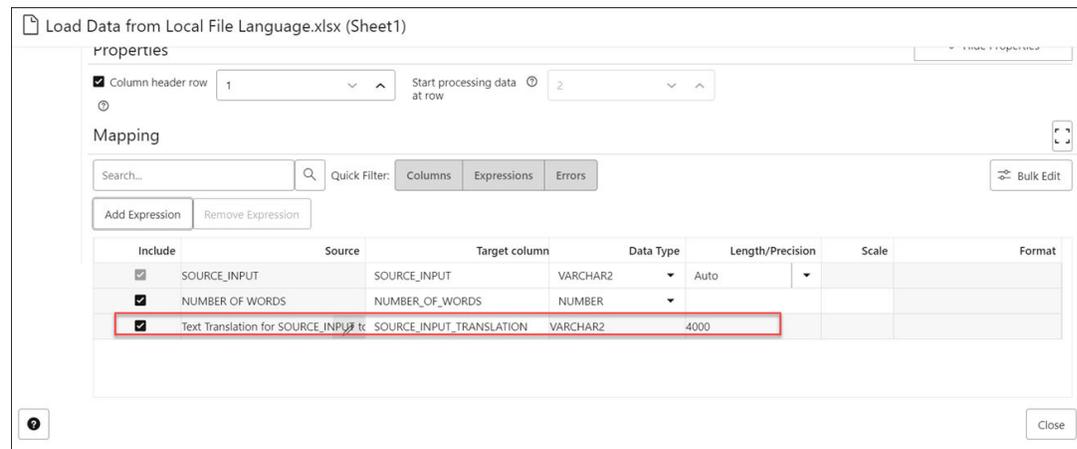
This opens the **Add Expression** dialog box. Select **AI Expressions**.



4. On the **Add Expression** dialog, specify the following fields:
 - **Expression Type:** From the Expression Type drop-down, select **Text Translation**.
 - **Input Column:** Select the column from the drop-down that you wish to analyze. In this case, we need to translate the text of the *SOURCE_INPUT* column.
 - **Target column:** Enter the name of the newly created expression column. For example, *SOURCE_INPUT_TRANSLATION*.
 - **Target Language:** Select the language of the text the tool translates from the drop-down. For example, *English*.

Refer to the [Parameters to Analyze Data](#) for more details.

5. Click **OK**. You will see a new row added to the mapping grid .This row determines the output expression column generated by the OCI Language service.



6. Click **Close**.
7. Click **Start** in the Data Load menu cart to load data from local files. You will view a confirmation message asking if you wish to start the load from local files.
8. Click **Run** to confirm.

When the data load job completes, the **Table and View Loads** of the Data Load page display the details of the source table that is loaded into the tool. It displays the number of rows and columns and when the upload was complete.

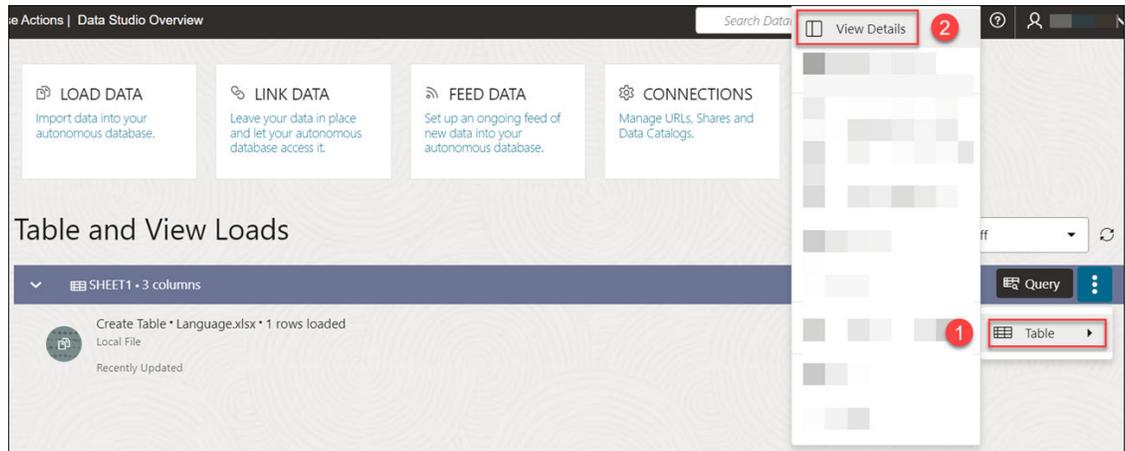
- [Output Data Generated from OCI Text Translation](#)
When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

Output Data Generated from OCI Text Translation

When you analyze columns using OCI Language Service model, the Data Studio generates a new expression column and saves the result in the updated table.

To locate the generated expression column, from the Database Actions Launchpad, navigate to **Data Load**. Select the table you load under the **Table and View Loads** section.

Click the three vertical dots beside the load name, and click **Table** then select **View Details**.



Use OCI Document Understanding to extract Tables from Images

You can utilize Oracle Cloud Infrastructure(OCI) Document Understanding Capabilities such as Table Extraction to detect tables from the data you load.

The supported file formats to use this feature are PDF, JPEG, JPG, PNG and Tag Image File Format (TIFF) formats.

For example, if a PDF document consists of all the employee details including their hiring date and salary, document understanding will identify the table and extract the table structure. It will extract the content of the table while maintaining the row and column relationship of cells.

In this section you will go through the following topics:

- [Overview of Table Extraction](#)
- [Perform Table Extraction from Image](#)
- [Overview of Table Extraction](#)

The Load Data tool supports Table Extraction in loading data from local files and loading data from cloud storage.

- [Perform Table Extraction from Image](#)

Let us perform table extraction with a sample PDF.

Overview of Table Extraction

The Load Data tool supports Table Extraction in loading data from local files and loading data from cloud storage.

Table Extraction

OCI Document Understanding service allows customers to uncover insights in unstructured documents powered by deep learning models. It enables you to extract text, tables and identify document types among other great capabilities. Table Extraction identifies all the tables in a document and extracts the content in tabular format maintaining the row/column relationship.

Note

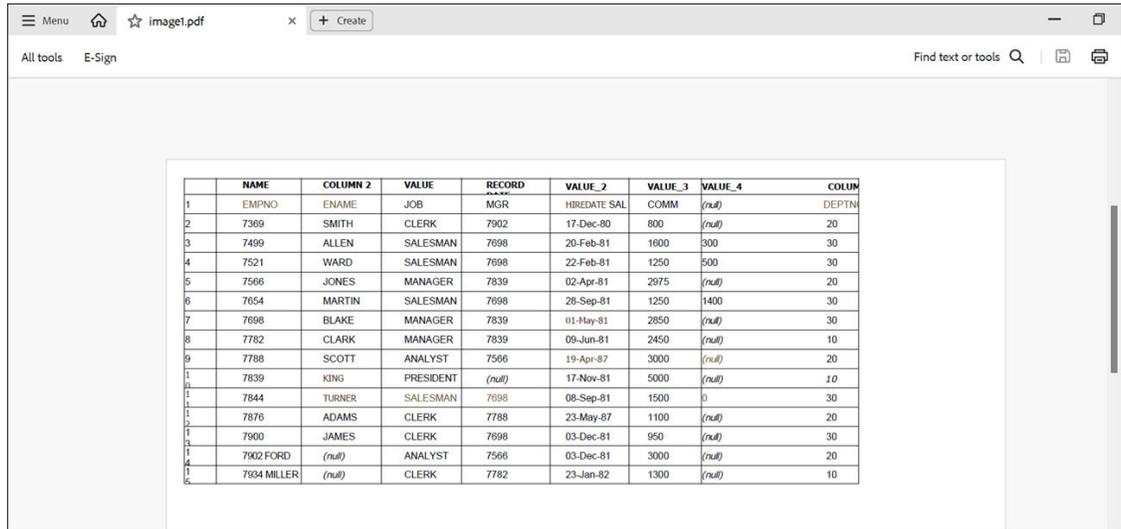
This service is available for source data in English Language only.

For more details, see [Table Extraction](#).

Perform Table Extraction from Image

Let us perform table extraction with a sample PDF.

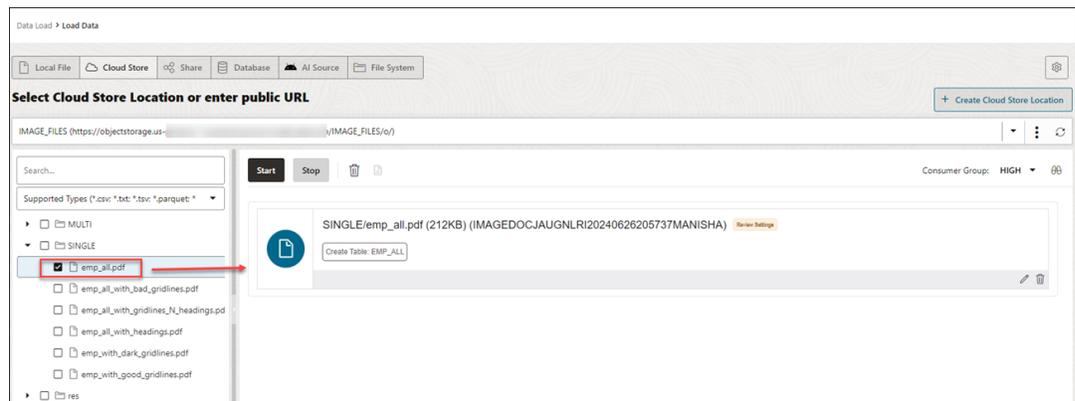
You can see a table in the PDF below.



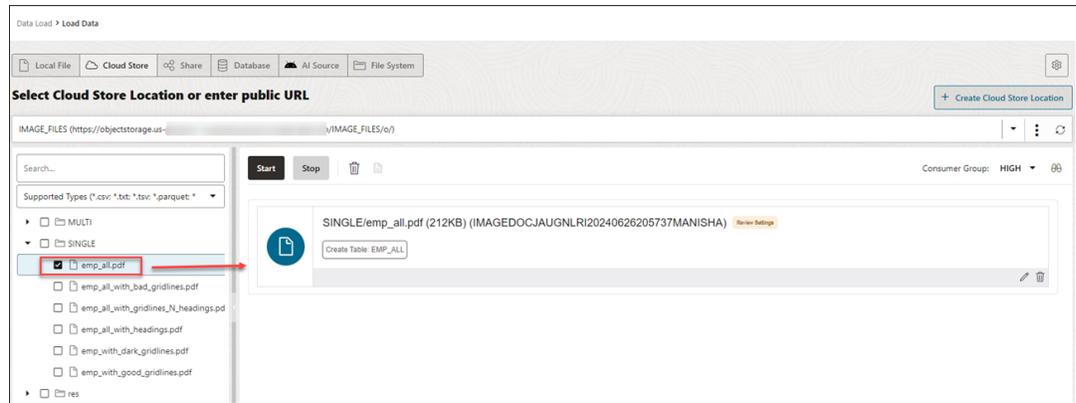
	NAME	COLUMN 2	VALUE	RECORD	VALUE_2	VALUE_3	VALUE_4	COLUM
1	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTN
2	7369	SMITH	CLERK	7902	17-Dec-80	800	(null)	20
3	7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
4	7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
5	7566	JONES	MANAGER	7839	02-Apr-81	2975	(null)	20
6	7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7	7698	BLAKE	MANAGER	7839	01-May-81	2850	(null)	30
8	7782	CLARK	MANAGER	7839	09-Jun-81	2450	(null)	10
9	7788	SCOTT	ANALYST	7566	19-Apr-87	3000	(null)	20
1	7839	KING	PRESIDENT	(null)	17-Nov-81	5000	(null)	10
1	7844	TURNER	SALESMAN	7698	08-Sep-81	1500	0	30
1	7876	ADAMS	CLERK	7788	23-May-87	1100	(null)	20
1	7900	JAMES	CLERK	7698	03-Dec-81	950	(null)	30
1	7902 FORD	(null)	ANALYST	7566	03-Dec-81	3000	(null)	20
1	7934 MILLER	(null)	CLERK	7782	23-Jan-82	1300	(null)	10

To extract a table from the PDF located in the cloud storage location:

1. Load the PDF file into the Data Load cart from the cloud storage. You can drag and drop the file from the navigator to the cart.



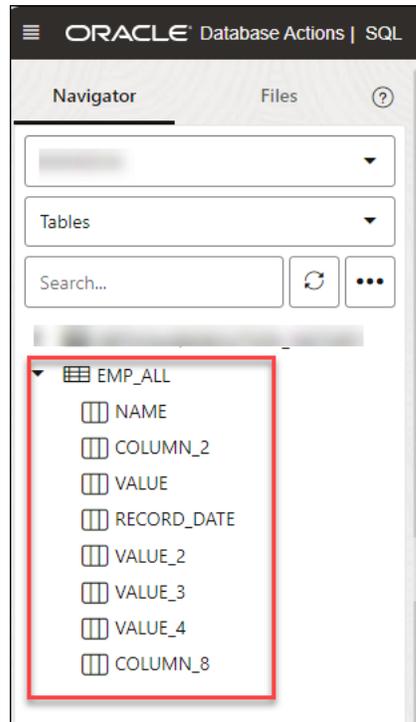
2. Click **Start** to load the data. You will receive a notification that confirms if you wish to start loading the data. Click **Run** to continue.



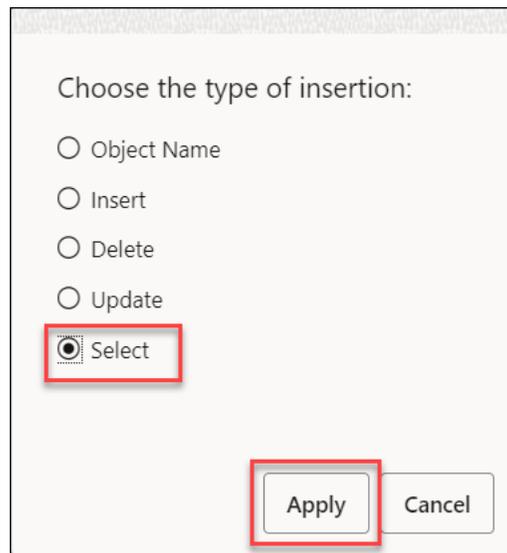
3. After the loading is complete, you can view the loaded file under Table and View Loads in the Data Load dashboard.



4. Extract the table from the PDF you loaded in the autonomous database:
 - Click **Database Actions** in the breadcrumb.
 - Select **Development**, then click **SQL**.
 - Select **Tables**, on the left Navigator, from the object drop-down. From the list of available tables, drag and drop the `EMP_ALL` table to the worksheet.



A dialog displays offering the types of available insertions.



Click **Select**, then **Apply**.

- Click **Run** (the arrow icon at the top of the worksheet). The tool displays the extracted table under the Query Results tab.

The screenshot shows the Oracle Database Actions | SQL interface. On the left is a Navigator pane with a tree view showing a schema named EMP_ALL containing tables: NAME, COLUMN_2, VALUE, RECORD_DATE, VALUE_2, VALUE_3, VALUE_4, and COLUMN_8. The main area displays a SQL query in a worksheet:

```

1 SELECT
2   NAME,
3   COLUMN_2,
4   VALUE,
5   RECORD_DATE,
6   VALUE_2,
7   VALUE_3,
8   VALUE_4,
9   COLUMN_8
10 FROM
11   MANZISHA.EMP_ALL;
12

```

Below the query is the Query Result pane, which shows the execution time as 0.008 seconds and a table of results:

	NAME	COLUMN_2	VALUE	RECORD_DATE	VALUE_2	VALUE_3	VALUE_4	COLUMN_8
1	EMPNO	ENAME	JOB	MGR	HIREDATE SAL	COMM	(null)	DEPTNO
2	7369	SMITH	CLERK	7902	17-Dec-80	800	(null)	20
3	7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
4	7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
5	7566	JONES	MANAGER	7839	02-Apr-81	2975	(null)	20
6	7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30

Use GeoJSON in Data Load

A GeoJSON object accommodates information about the specific geometry (e.g. Point, Line, Polygon, etc.) along with optional metadata (e.g. ID, etc.).

The extension for a GeoJSON file is *.geojson. You can load GeoJSON data into the Autonomous Database using the Data Load in Data Studio. If the table contains GeoJSON data, then the data is loaded in a column that projects GeoJSON data from the document set of SQL data type `SDO_GEOMETRY`.

Load a table with GeoJSON Data

When you load a table in Data Studio with GeoJSON data and review its settings, you will see that it creates table `Brazil_Geo`, which has a column `geometry` of GeoJSON data.

The screenshot shows the 'Load Data from Cloud Store Location' settings window for a file named 'brazil_geo.geojson (14MB)'. The 'Table' section is set to 'Create Table' with the table name 'BRAZIL_GEO'. The 'SQL' section is set to 'None'. The 'Mapping' section shows a table with columns for 'Include', 'Source', 'Target column', 'Data Type', and 'Length/Precision':

Include	Source	Target column	Data Type	Length/Precision
<input checked="" type="checkbox"/>	geometry	GEOMETRY	SDO_GEOMETRY	
<input checked="" type="checkbox"/>	name	NAME	VARCHAR2	Auto

After you load `BRAZIL_GEO` you will view that the tool loads GeoJSON object into a new column `geometry` with the data type `SDO_GEOMETRY`.

Edit Table				
Schema		ADPTEST	Name	
			BRAZIL_GEO	
Columns	General		Constraints	
Primary Key	+ - ↑ ↓			
Unique Keys		Name	Data Type	PK Identity Column
Indexes	1	GEOMETRY	MDSYS.SDO_GEOMETRY	<input type="checkbox"/> <input type="checkbox"/>
	2	NAME	VARCHAR2 (64)	<input type="checkbox"/> <input type="checkbox"/>

You can also view the same in the **Data Definition** tab when you **View Details** of the Table load after it is run.

BRAZIL_GEO		
Preview	1	
	2	CREATE TABLE "ADPTEST"."BRAZIL_GEO"
	3	("GEOMETRY" "MDSYS"."SDO_GEOMETRY" ,
	4	"NAME" VARCHAR2(64)
Lineage	5) SEGMENT CREATION IMMEDIATE
	6	PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
Impact	7	NOCOMPRESS LOGGING
	8	STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
	9	PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
Statistics	10	BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
	11	TABLESPACE "DATA" ;
Job Report		
Data Definition		

- [Add Spatial Expression for GeoJSON columns](#)
Oracle Spatial is designed to make spatial data management easier and more natural to users of location-enabled applications and geographic information system (GIS) applications.
- [Parameters for Spatial Data Analysis](#)
When you invoke an **Add Expression** and select **Spatial Expressions** from the **Add Expression** wizard, you must configure the model using parameters.
- [Perform Spatial Analysis](#)
You can load Spatial data using the Data Load tool from local files or cloud storage.
- [Output Data Generated from Spatial Analysis](#)
When you analyze columns using Spatial Expressions, the Data Studio generates a new expression column and saves the result in the updated table.

Add Spatial Expression for GeoJSON columns

Oracle Spatial is designed to make spatial data management easier and more natural to users of location-enabled applications and geographic information system (GIS) applications.

Once spatial data is stored in an Oracle database, it can be easily manipulated, retrieved, and related to all other data stored in the database.

The spatial data model in Oracle Spatial is a hierarchical structure consisting of elements, geometries, and layers. Layers are composed of geometries, which in turn are made up of elements.

Tolerance is used to associate a level of precision with spatial data. **Tolerance** reflects the distance that two points can be apart and still be considered the same (for example, to accommodate rounding errors). The tolerance value must be a positive number greater than zero. See [Tolerance in Spatial Data](#) for more information.

Parameters for Spatial Data Analysis

When you invoke an **Add Expression** and select **Spatial Expressions** from the **Add Expression** wizard, you must configure the model using parameters.

Table 12-3 Parameters for Spatial Data Analysis

Parameter	Description
Expression Type	<p>Select any one of the following spatial attributes such as Length or Area:</p> <ul style="list-style-type: none"> Length: Selecting this attribute returns and stores the perimeter of a two-dimensional polygon in the mapping table with a target column of type <code>GEOMETRY_LENGTH</code>. Area: Selecting this attribute returns and stores the area of a two-dimensional polygon in the mapping table with a target column of type <code>GEOMETRY_AREA</code>.
Input Column	<p>Select the column which you wish to analyze for Spatial Data:</p> <p>You will by default view a <code>GEOMETRY</code> column in Source mapped to the <code>SDO_GEOMETRY</code> column value. This also means that the table is spatially enabled.</p>
Target Column	<p>This is the newly created expression column defined in the target table which stores the result of the Spatial Expression and depends on the expression type value you select:</p> <ul style="list-style-type: none"> <code>GEOMETRY_LENGTH</code>: If you select Length as your Expression type, you can view the target column of type <code>GEOMETRY_LENGTH</code> which displays the geometry's length value. <code>GEOMETRY_AREA</code>: If you select Area as your Expression type, you can view the target column of type <code>GEOMETRY_AREA</code> which displays the geometry's area value.

Perform Spatial Analysis

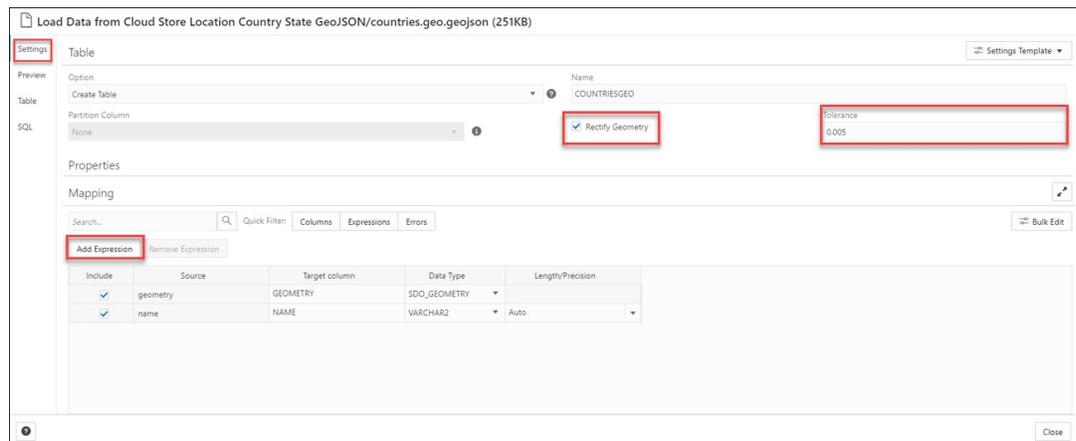
You can load Spatial data using the Data Load tool from local files or cloud storage.

To perform Spatial Analysis on GeoJSON data:

1. After you load data into the Data Load cart from local files or cloud storage, you can view the file in the cart. Click the **Settings** icon.



2. On the Settings tab of the wizard, select **Rectify Geometry** to fix invalid geometries. This icon corrects the most common errors such as duplicate points, polygon orientation errors, polygon construction errors etc. If it detects any uncorrectable error, it fails with an `ORA-13199: the given geometry cannot be rectified` exception.



Note

You cannot fix geometries having wrong orientation with the **Rectify Geometry** icon.

3. Specify the **Tolerance** value. In the above example, we specify 0.005 as tolerance, i.e., 5mm.
4. Click **Add Expression** under the Mapping section and select **Spatial Expressions**.

On the **Add Expression** dialog, specify the following fields:

- **Expression Type:** From the Expression Type drop-down, select **Area** to display the area of a two-dimensional polygon. You can also select **Length** to display the perimeter of a two-dimensional polygon.
- **Input Column:** You will view a `GEOMETRY` column by default. This source column stores the area of the geometry. In case you select **Length** as an expression type, this source column stores the length of the geometry.
- **Target column:** Specifies the name of the newly created special expression column. For example, `GEOMETRY_AREA` for **Area** type expression parameter and `GEOMETRY_LENGTH` for **Length** type expression parameter.

Under the Advanced Options section, specify the following field values:

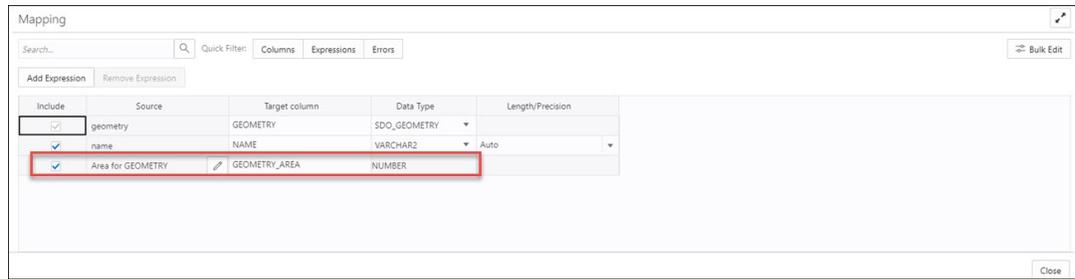
- **Tolerance:** Enter the tolerance value. For example, `0.005`.
- **Unit:** Select a unit of measure of the tolerance from the drop-down. For example, `Square meter`.

Note

If the file you load does not contain spatial data, you will view the following error on the **Add Expression** wizard "No input columns have the `SDO_GEOMETRY` data type for the currently selected expression type of Area".

Refer to the [Parameters for Spatial Data Analysis](#) for more details.

5. Click **OK**. You will see a new row added to the mapping grid. This row determines the output spatial expression column generated by the OCI Language service. In this example, you can view a new `GEOMETRY_AREA` will be created in the target table after we load the table.



For more information on Oracle Spatial, refer to [Oracle Spatial Developer's Guide](#).

When the data load job completes, the **Table and View Loads** of the Data Load page display the details of the source table that is loaded into the tool. It displays the number of rows and columns and when the upload was complete.

Output Data Generated from Spatial Analysis

When you analyze columns using Spatial Expressions, the Data Studio generates a new expression column and saves the result in the updated table.

To locate the generated expression column, from the Database Actions Launchpad, navigate to **Data Load**. Select the table you load under the **Table and View Loads** section.

Click the three vertical dots beside the load name, and click **Table** then select **View Details**.

This opens the Preview tab of the data load which displays the updated source file. For example, here is an output dataset from spatial analysis of the COUNTRIESGEO.GeOJSON file. Here, GEOMETRY_AREA is the spatial analysis of the input column. This column displays the area of the two-dimensional polygons.

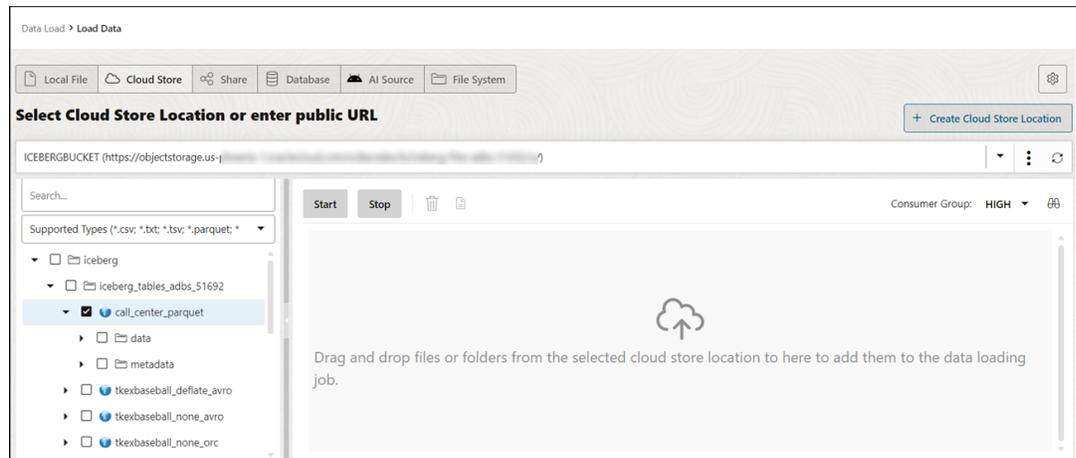
	GEOMETRY	NAME	GEOMETRY_AREA
1	["srid":4326,"polygon":{"bou	Afghanistan	652270299811.644
2	["srid":4326,"geometrycoller	Angola	1245461291967.91
3	["srid":4326,"polygon":{"bou	Albania	29694798558.8322
4	["srid":4326,"polygon":{"bou	United Arab Emirates	79881588242.9788
5	["srid":4326,"geometrycoller	Argentina	2784468563629.25
6	["srid":4326,"polygon":{"bou	Armenia	28656554690.2027
7	["srid":4326,"geometrycoller	Antarctica	12335956722891.9
8	["srid":4326,"polygon":{"bou	French Southern and Antarc	11602533443.8801
9	["srid":4326,"geometrycoller	Australia	7687613132736.02
10	["srid":4326,"polygon":{"bou	Austria	85064966191.1483
11	["srid":4326,"geometrycoller	Azerbaijan	91113317905.705
12	["srid":4326,"polygon":{"bou	Burundi	26239375271.9818
13	["srid":4326,"polygon":{"bou	Belgium	30126112398.1728
14	["srid":4326,"polygon":{"bou	Benin	116998681985.827
15	["srid":4326,"polygon":{"bou	Burkina Faso	271593711633.882
16	["srid":4326,"polygon":{"bou	Bangladesh	133782082749.821
17	["srid":4326,"polygon":{"bou	Bulgaria	110216843102.657
18	["srid":4326,"geometrycoller	The Bahamas	15584800825.1057
19	["srid":4326,"polygon":{"bou	Bosnia and Herzegovina	50605096409.9281
20	["srid":4326,"polyvaon":{"bou	Belarus	208969892313.672

Load Apache Iceberg Tables

Data Studio supports loading and linking Iceberg tables from the object store.

The tool can load multiple versions of the same file.

1. Select the Cloud Store Location that has iceberg tables defined. In this example, we will select the ICEBERGBUCKET stored in OCI.

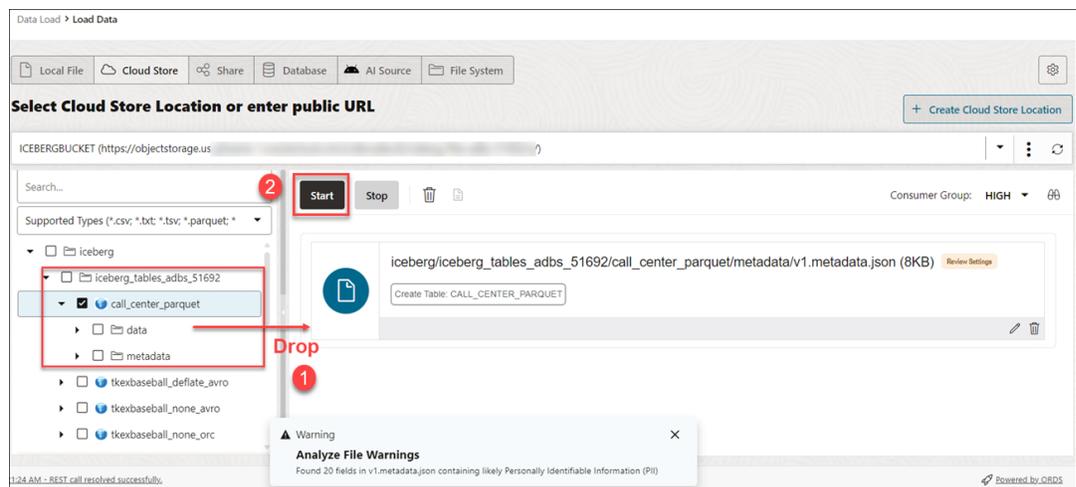


You can detect the Iceberg files with the Iceberg symbol



beside its name.

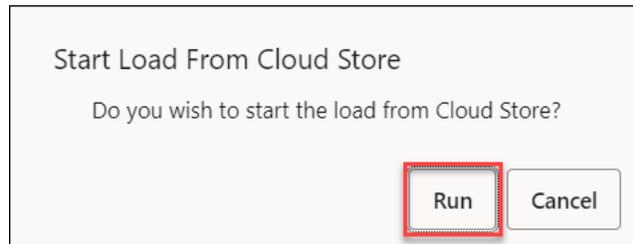
2. Drag the folder to the file navigator on the left and drop it into the cart on the right. The folder also consists of the metadata JSON file that keeps track of the table snapshots, partitioning scheme, and schema information. When you add the folder to the cart, a warning about the number of files consisting of a Personal Identifiable Information (PII) prompt is displayed.



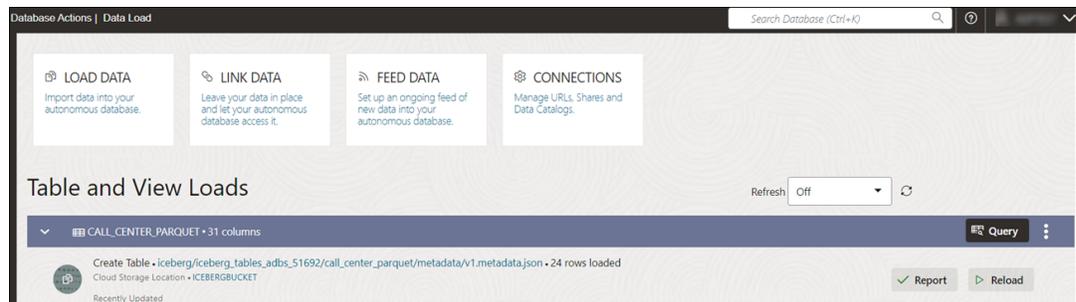
Note

: If you load multiple files instead of a folder, the tool asks if you want to load all the objects from the multiple source files into a single target table. Click **OK** to continue or **Escape** to cancel.

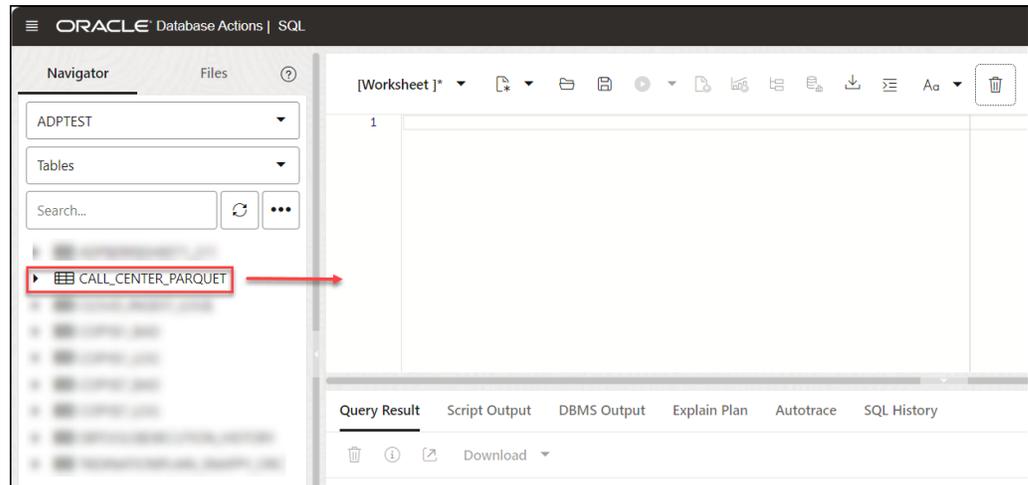
- Click **Start** in the data link cart menu bar. You will view a notification that says if you wish to start the load from the cloud store. Select **Run** to continue.



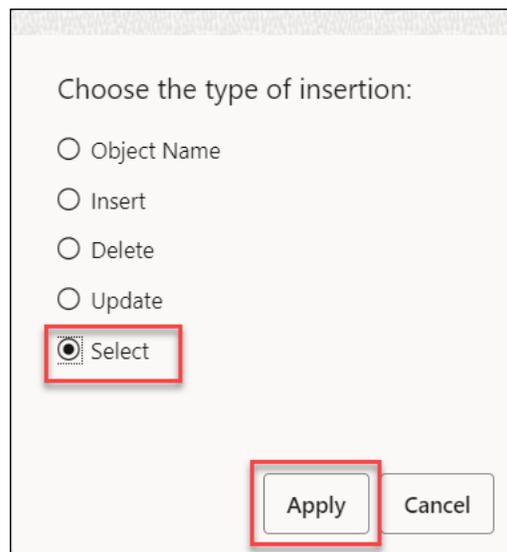
- You will view the loading progress under **Table and View Loads**. After loading, you will view the **Report** and **Reload** icons on the table load.



- Query Iceberg tables you loaded in the Autonomous Database:
 - Click **Database Actions** in the breadcrumb.
 - Select **Development**, then click **SQL**.
 - Select **Tables**, on the left Navigator, from the object drop-down. From the list of available views, drag and drop the **CALL_CENTER_PARQUET** table to the worksheet.



- A dialog displays offering the types of available insertions.



Click **Select**, then **Apply**.

- Click **Run** (the arrow icon at the top of the worksheet). The tool displays the iceberg data under Query Results.

The screenshot shows the Oracle Data Studio interface. At the top, there is a toolbar with a green play button icon highlighted by a red box. Below the toolbar, a SQL query is displayed in a text area. The query is a SELECT statement with the following columns: CC_CALL_CENTER_SK, CC_CALL_CENTER_ID, CC_REC_START_DATE, CC_REC_END_DATE, CC_CLOSED_DATE_SI, CC_OPEN_DATE_SK, CC_NAME, and CC_CLASS. The results are shown in a table below the query. The table has 4 rows of data.

CC_CALL_CENTER_SK	CC_CALL_CENTER_ID	CC_REC_START_DATE	CC_REC_END_DATE	CC_CLOSED_DATE_SI	CC_OPEN_DATE_SK	CC_NAME	CC_CLASS
1	AAAAAAAAABAAAAAAA	1998-01-01	(null)	(null)	2450952	NY Metro	large
2	AAAAAAAACAAAAAAAA	1998-01-01	2000-12-31	(null)	2450806	Mid Atlantic	medium
3	AAAAAAAACAAAAAAAA	2001-01-01	(null)	(null)	2450806	Mid Atlantic	medium
4	AAAAAAAEEAAAAAAA	1998-01-01	2000-01-01	(null)	2451063	North Midwest	medium

Loading Data from File System

You can load files from file system directories to your Autonomous Database.

You can set filters on the data for a table to load only the specified data. For example, to limit the files to only those that are CSV files, enter *.CSV in the file extension filter.

Configure and run a data load job from the Load Cloud Object page. To open that page:

1. On the Data Studio tab and select **Data Load**. You will view the Data Load dashboard.
2. Click **LOAD DATA** and select the **File System** option.

On the top of the page you need to select the directory from where you need to load the files. On the left side of the page is a navigator pane, where you choose the files in the directory containing the data. On the right of the page is the data load cart, where you stage the files and folders for the data load job. You can set options for the data load job before running it. The Autonomous Database comes with predefined CPU/IO shares assigned to different consumer groups. You can set the consumer group to either low, medium or high while executing a data load job depending on your workload. To load files from a directory into your database, do the following:

1. Prepare the Data Load Job: Refer to the [Prepare the Data Load Job](#) section for more details.
2. Add Files or Folders for the Data Load Job: Refer to the [Add Files or Folders for the Data Load Job](#) section for more details.
3. Enter Details for the Data Load Job: Refer to the [Enter Details for the Data Load Job](#) for more details.
4. Run the Data Load Job: Refer to the [Run the Data Load Job](#) section for more details.
5. View Details About the Data Load Job After It Is Run: Refer to the [View Details About the Data Load Job After It Is Run](#) section for more details.
6. View the Table Resulting from the Data Load Job: Refer to the [View the Table Resulting from the Data Load Job](#) section for more details.

- [Create Directories in Database Actions](#)

In the Autonomous Database, there is a preconfigured `data_pump_dir` where you can store files. You can also create directories, drop directories, and attach network file systems.

Create Directories in Database Actions

In the Autonomous Database, there is a preconfigured `data_pump_dir` where you can store files. You can also create directories, drop directories, and attach network file systems.

For example, you can use the `CREATE DIRECTORY` command to create additional directories. Use the database `DROP DIRECTORY` command to drop directories and use `DBMS_CLOUD.LIST_FILES` to list the contents of a directory.

Create a Directory

To add a directory, you must have the `CREATE ANY DIRECTORY` system privilege. The `ADMIN` user is granted the `CREATE ANY DIRECTORY` system privilege. The `ADMIN` user can grant `CREATE ANY DIRECTORY` system privileges to other users.

See `CREATE DIRECTORY` for more information.

Note

- `CREATE DIRECTORY` creates the database directory object in the Autonomous Data Warehouse database and also creates the file system directory. For example the directory path could be:

```
/u03/dbfs/7C149E35BB1000A45FD/data/stage
```

- You can create a directory in the root file system to see all the files with the following commands:

```
CREATE OR REPLACE DIRECTORY ROOT_DIR AS '';
```

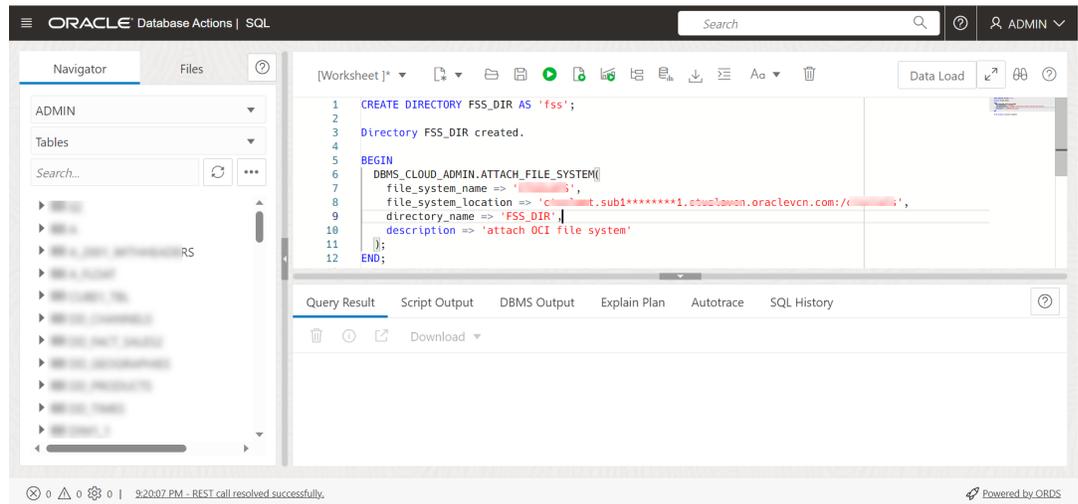
After you create the `ROOT_DIR` directory, use the following command to list all files:

```
SELECT * FROM DBMS_CLOUD.list_files('ROOT_DIR');
```

- To run `DBMS_CLOUD.LIST_FILES` with a user other than `ADMIN` you need to grant read privileges on the directory to that user.

Let us demonstrate how to create a directory and access it from Data Studio:

- **Create a directory in Database Actions:**
Login to your Database Actions instance and select the SQL card under **Development**. You can view the SQL worksheet. Now create a directory and attach a file system name of your choice to the directory you create. In the below given example, `FSS_DIR` is the name of the directory.



Run the above command. The above command gives the following output:

```
PL/SQL procedure successfully completed.
```

- **Attach the file system**

Attach your file system with the name of your choice to the FSS_DIR directory via the DBMS_CLOUD_ADMIN.ATTACH_FILE_SYSTEM procedure.

```
BEGIN
  DBMS_CLOUD_ADMIN.ATTACH_FILE_SYSTEM(
    file_system_name => '*****',
    file_system_location => '*****.sub1*****1.*****.oraclevcn.com:/
*****',
    directory_name => 'FSS_DIR',
    description => 'attach OCI file system'
  );
END;
/
```

Run the above command to view the following output:

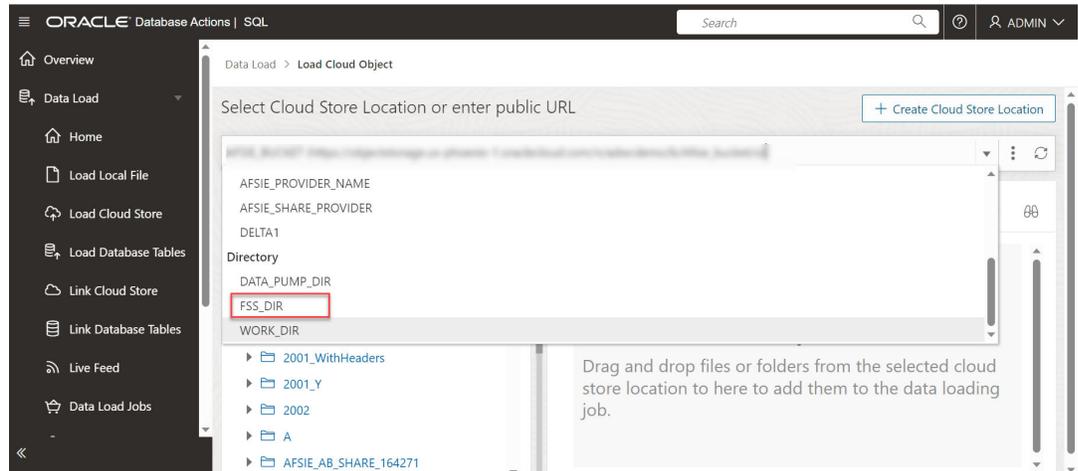
```
PL/SQL procedure successfully completed.
```

- To view the attached file system, run the following command:

```
SELECT file_system_name, file_system_location, directory_path FROM
dba_cloud_file_systems;
```

You will view the file system name, file system location and directory path.

- You can view the new directory along with the files attached to it by navigating to Load Cloud Object under Data Load menu of the Data Studio tool. Click the **Select Cloud Store**

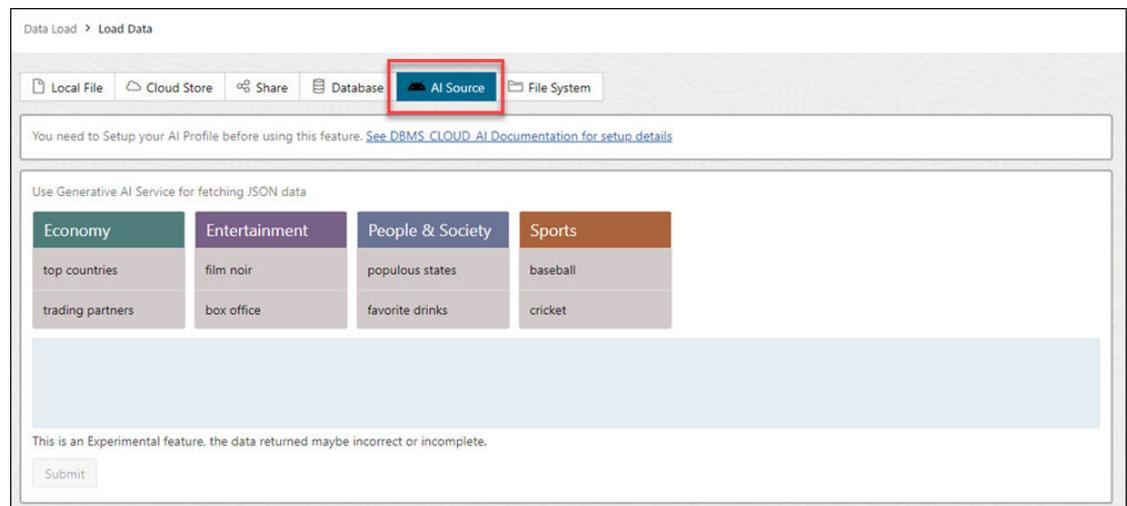
location drop-down.

You can view the newly created directory `FSS_DIR`. You can load data from the file system directories to the autonomous database using the Data Load tool. See [Loading Data from File System](#).

Loading Data from AI Source

You can use Data Studio tools to load data from AI source.

On the Data Load dashboard, click **Load Data** card, and select **AI Source**.



You need to perform a pre-requisite of setting up your AI profile before using this feature. See [Configure DBMS_CLOUD_AI Package](#) for details on the setup of this feature.

To set an AI profile, you must have:

- An OpenAI account, or a Cohere account, or Azure OpenAI service
- An access to `DBMS_CLOUD_AI` package.

Follow the steps in [Use DBMS_CLOUD_AI to Configure AI Profiles](#) section to create and configure your AI profile and utilize this feature.

Loading Data from Share

You can select tables from a Share. You need to subscribe and access provided data share.

To load tables from a share, click **Load Data** on the Data Load Dashboard. Click **Share** on the Load Data page. Click **+ Subscribe to Share Provider** to subscribe to a Share Provider.

Subscribe to the Data Share Provider

To subscribe, you need to use the information contained in the uploaded JSON profile you received from the share provider.

1. On the Subscribe to Data Share Provider, enter the **Provider Name**.

Subscribe to Share Provider

1 Provider Settings 2 Add Shares

Provider Name *
Review_provider

Description

Share Source *
 Select from Available Providers Add New Provider JSON

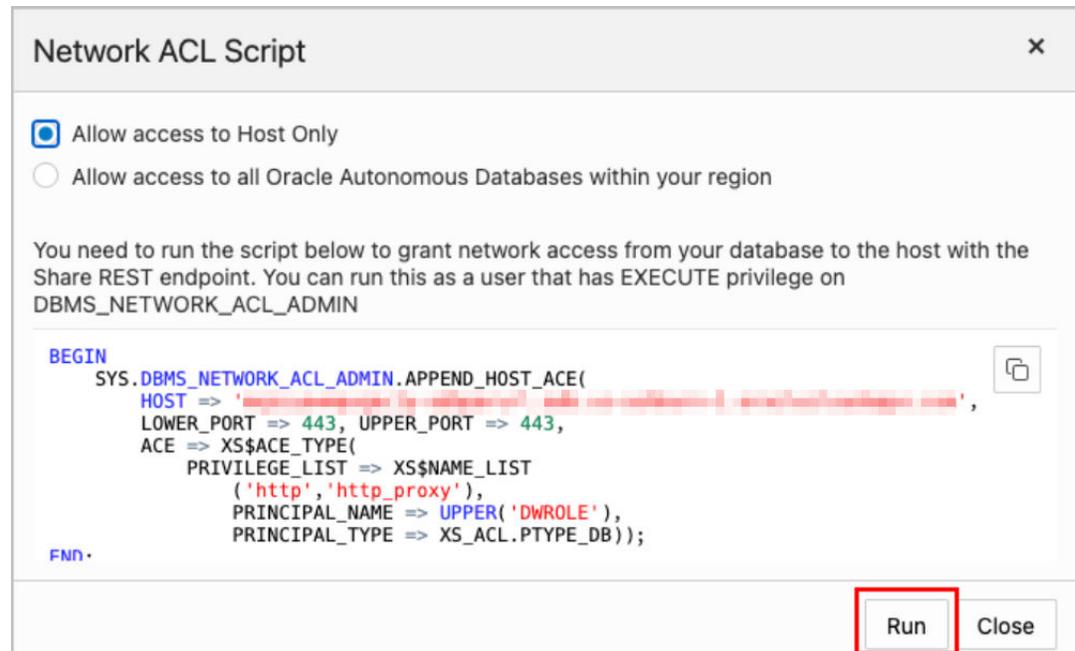
Share Provider JSON *
 From File JSON

delta_share_profile-2.json file selected for loading Share
Select a file or drop one here.

Back Next Subscribe Cancel

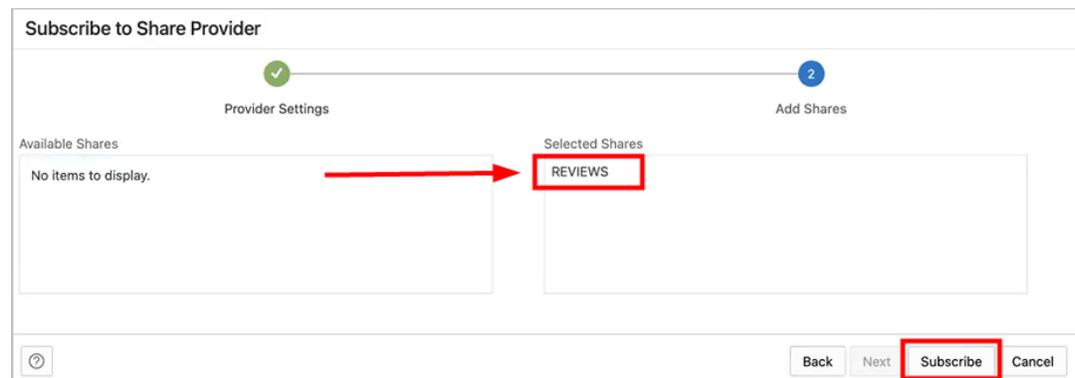
2. Select Add New Provider JSON and click on the Delta Share Profile JSON to drag and drop the JSON profile.
3. Click **Next** to progress to the Add Shares tab.
4. Select the level of network access you want to allow from your database to the host with the Share REST endpoint, and click **Run**. In this example, **Allow access to Host Only** is

selected.

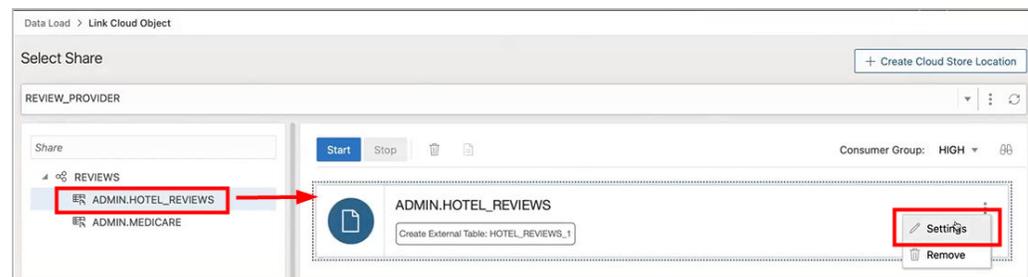


5. To register the shares available to you, move shares from **Available Shares** to **Selected Shares** and click **Subscribe**.

The screenshot below shows the **REVIEWS** share moved from **Available Shares** to **Selected Shares** before clicking **Subscribe**.

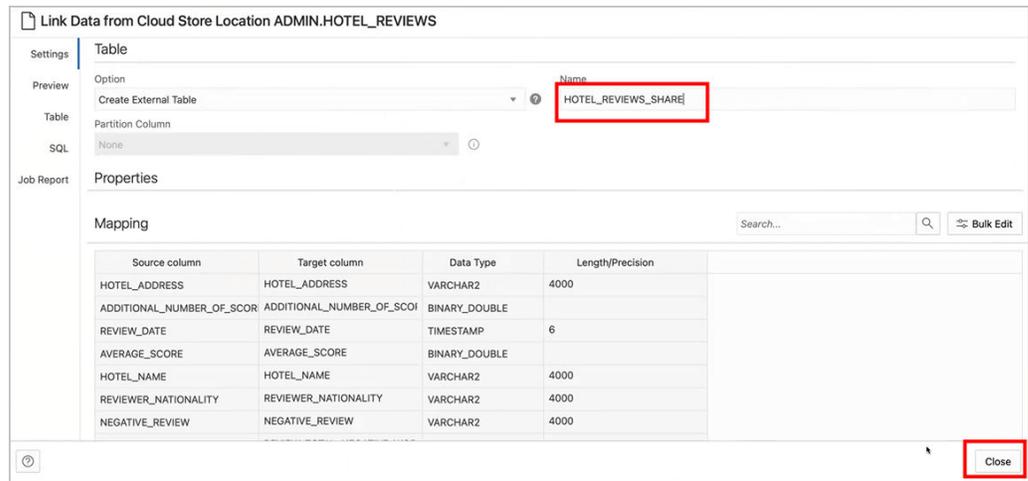


6. Create external tables derived from tables selected from the data share.
 - a. Drag and drop tables from the selected share. You can optionally click **settings** to view the table details. In this example, the only table selected is **HOTEL_REVIEWS**.

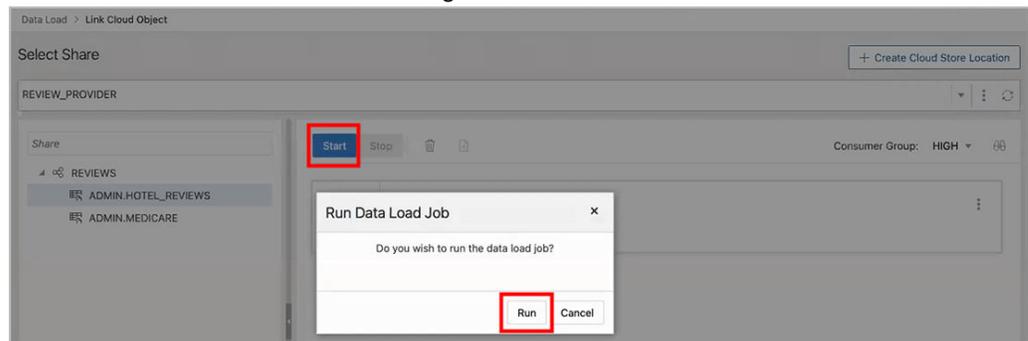


- b. You can optionally change the name of your table and click **Close**.

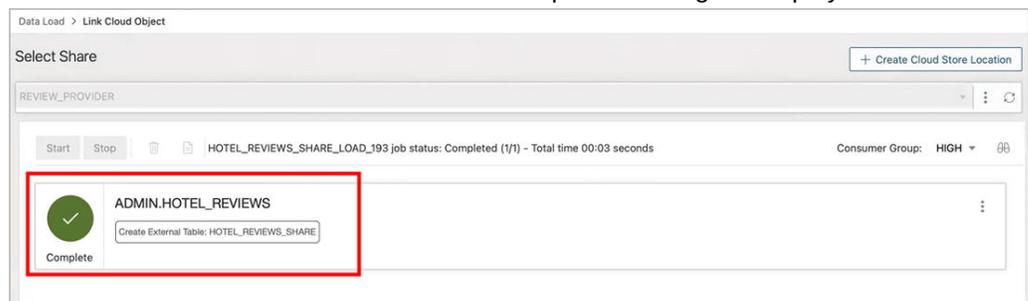
In this example, the name is changed from **HOTEL_REVIEWS** to **HOTEL_REVIEWS_SHARE**.



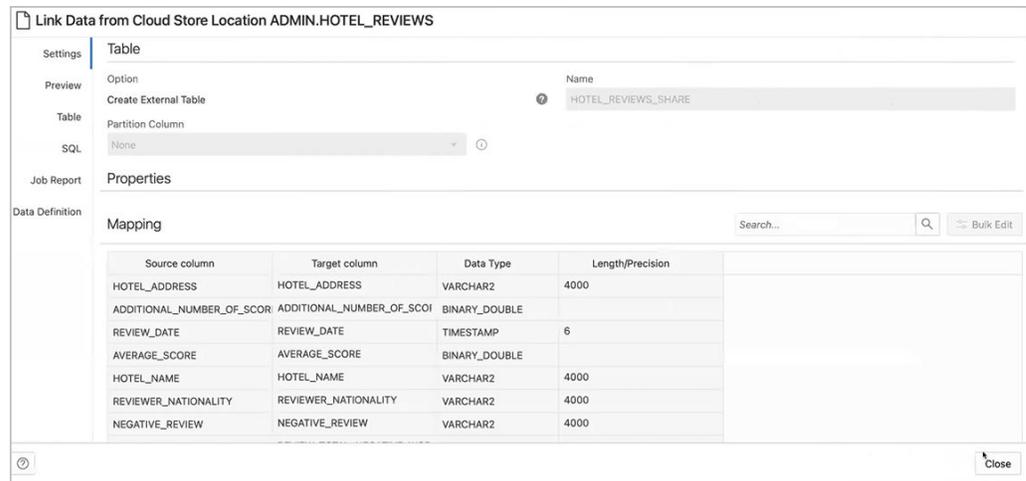
- c. Create the external table by clicking **Start**, on the Select Share page, and then clicking **Run** on the Run Data Load Job dialog.



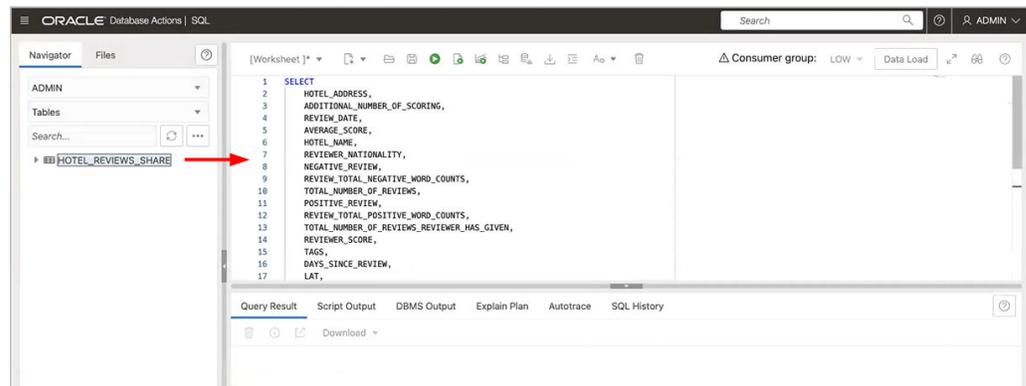
- d. When the external tables are created the Complete message is displayed.



- e. Click on the external table name to view the table details.



7. View the created tables from Database Actions.
 - a. Click on **Database Actions**, in the breadcrumb, to go back to the Database Actions launchpad.
 - b. Click on the **SQL** tile.
 - c. Select the external table, drag and drop it into the worksheet. The SQL Select statement for the table appears. This SQL statement can be run to consume the shared data.



8. Drag and drop tables from selected share

Create Live Feed from Data Load

The Data load tool loads the data from folders in cloud object stores and enables it to schedule repeated data loads in real time. This is the creation of Live Feed from a data load job.

Once the data load is complete, you can create a Live Feed from a cart item that loaded an object store folder using the Create Table or Drop Table and Create New Table options.

To create a live feed from Data Load:

1. Click **Selector** to display the navigation menu. Under Data Studio, select **Data Load**.
2. Select the **Load Data** tile to load data from various sources such as local files, databases, cloud store, directories, and Shares.
3. Click **Cloud Store** to load objects from URLs or Cloud store links.

4. Select the cloud store location from the drop-down. If are not able to view the cloud store location, select **Create Cloud Store Location** to create a new cloud store location. Follow the steps described in the [Create Oracle Cloud Infrastructure Native Credentials](#) if you do not have a cloud location available.
5. After you select the cloud store location, you can view the list of folders and files present in the cloud storage. Add files from the cloud store to the data load cart, where you can edit the details of the load job.

Note

The Data load tool does not support the creation of a live feed from a loaded cart item that consists of a single file in CSV, XLS, XLSX, TSV, TXT, XML, JSON, and AVRO format or of a folder that contains a file in XLSX format.

6. To add the folders, drag a folder from the file navigator on the left and drop them into the cart on the right. When you add the folder to the cart, a prompt is displayed that asks if you want to load all the objects from the multiple source files into a single target table. Click **Yes** to continue or **No** to cancel. You can add multiple folders to the cart, the data represented by each card will be loaded into a separate table, but all the items in the cart will be processed as part of the same data load job.
7. Select **Settings** (pencil icon) from the data load cart to enter the details about the data load job.
8. In the **Settings** tab of the Load Data from the Cloud Store Location, you can select **Create Table** or **Drop Table and Create New Table** from the **Option** drop-down.

Note

The Live feed tool works with the Data load job only if you create a table and insert the data into a new table or drop the existing table and insert the data into a new table.

9. Enter the other details for the data load job. For more information on entering the details, refer to the [Enter Details for the Data Load Job](#) topic.
10. Once you have added data sources to the data load cart and entered details about the data load job, select **Start** to run the job.
11. After the data load job is run, the data load cart displays a green check mark



Complete

which indicates that the data load job is complete.

12. Click **Create Live Table Feed** on the data load cart to create a Live table feed from the data load job. You will view a successful message that says the creation of Live Table feed is successful and if you wish to edit the live table feed. Click **Yes** to continue and **No** to cancel. Selecting **Yes** opens an Edit Live Table Feed wizard.
13. On the Edit Live Table Feed wizard, you can view the **Cloud Store Location** of the source folder and the **Object Filter** applied to the data.

Select any file whose data you want to preview from the Preview drop-down in the Data Source tab: The field shows the total number of files present in the cloud store folder you loaded. A preview of the data is displayed.

Edit Live Table Feed

1 Data Source Table Settings Preview Live Feed Settings

Cloud Store

Cloud Store Location *
AA

Object Filter 2

Preview (4 Files Selected)

Preview (4 Files Selected) 1

File Preview

```
7369,SMITH,CLERK,7902,17-DEC-80,800,,20
7499,ALLEN,SALESMAN,7698,20-FEB-81,1600,300,30
7521,WARD,SALESMAN,7698,22-FEB-81,1250,500,30
7566,JONES,MANAGER,7839,02-APR-81,2975,,20
7554,MARTIN,SALESMAN,7698,29-SEP-81,1200,1400,30
7698,BLAKE,MANAGER,7839,01-MAY-81,2850,,30
7782,CLARK,MANAGER,7839,09-JUN-81,2450,,18
7788,SCOTT,ANALYST,7566,19-APR-87,3000,,20
7839,KING,PRESIDENT,,17-NOV-81,5000,,10
7844,TURNER,SALESMAN,7698,08-SEP-81,1500,0,30
7876,ADAMS,CLERK,7788,23-MAY-87,1100,,20
7900,JAMES,CLERK,7698,03-DEC-81,950,,30
```

2 Back Next Save Cancel

14. Click **Next** to progress to the Table Settings tab.

Edit Live Table Feed

2 Table Settings Preview Live Feed Settings

Properties

Encoding: UTF8ASCII - ASCII 7-bit American

Text enclosure: -

Field delimiter: Comma

Column header row 1

Newlines included in data values

Start processing data at row 2

Convert invalid values to null

Mapping

Include	Source column	Target column	Data Type	Length/Precision	Scale	Format
<input checked="" type="checkbox"/>	7369	C7369	NUMBER			
<input checked="" type="checkbox"/>	SMITH	SMITH	VARCHAR2	32767		
<input checked="" type="checkbox"/>	CLERK	CLERK	VARCHAR2	32767		
<input checked="" type="checkbox"/>	7902	C7902	NUMBER			
<input checked="" type="checkbox"/>	17-DEC-80	C17_DEC_80	DATE			DD-MON-RR
<input checked="" type="checkbox"/>	800	C800	NUMBER			
<input checked="" type="checkbox"/>	COLNAME_COL7_MISSING	COLNAME_COL7_MISSING	NUMBER			

Back Next Save Cancel

You can view the Properties and Mapping details of the data load job on the Table Settings tab.

Note

You cannot select or edit any of the details of this tab.

15. Click **Next** to progress to the Preview tab of the wizard.

Select any file from the Preview drop-down to view the file. The Table Preview displays the preview of the file you select from the drop-down.

Table Preview

	C7389	SMITH	CLERK	C7902	C17_DEC_80	C800	COLNAME_COL7_MISSING	C20
1	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
2	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
3	7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20
4	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
5	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30
6	7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10
7	7788	SCOTT	ANALYST	7566	19-APR-87	3000	(null)	20
8	7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
9	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
10	7876	ADAMS	CLERK	7788	23-MAY-87	1100	(null)	20
11	7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30
12	7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
13	7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	20

16. Click **Next** to view the Live Feed Settings tab of the wizard. On the Live Feed Settings tab, specify the following values in the field:

Live Table Feed Name *
ADP_2_FEED

Target Table Name *
ADP_2

Notification
 Enable for Notification

Scheduled
 Enable for Scheduling

Every 12 Hours on all Days
Start Date 12/07/2023 23:19
End Date 12/14/2023 23:19

- **Enable for Notification:** Select this option so that new or changed data in the data source will be loaded based on an Oracle Cloud Infrastructure notification. When you select this option, you can avoid delays that might occur when polling is initiated on a schedule (that is, if you selected the live table feed Scheduled option).

When you select the Enable for Notification option, you must also configure your object store bucket to emit notifications

- **Enable For Scheduling:** Select this option to set up a schedule for data feed. In the time interval fields, enter a number, and select a time type and the days on which to poll the bucket for new or changed files. For example, to poll every two hours on Monday, Wednesday, and Friday, enter **2**, select **Hours**, and then select **Monday, Wednesday, and Friday** in the appropriate fields. You can select **All Days, Monday to Friday, Sunday to Thursday**, or **Custom** from the Week Days drop-down. The Custom field enables you to select **Monday, Tuesday, Wednesday, Thursday and Friday**.

Select a start and end date. If you don't select a start date, the current time and date will be used as the start date. The end date is optional. However, without an end date, the feed will continue to poll.

The rest of the fields displayed in the wizard such as the **Live Table Feed Name**, **Target Table Name**, and **Consumer Group** are greyed out and disabled for selecting or editing.

Click **Save** to save and create a Live Table Feed from a data load cart.

Loading Data From Local Files

To load data from local files into your Oracle Autonomous AI Database, on the Data Load page, select **LOAD DATA** and **LOCAL FILE**.

Drag one or more files from your local file system navigator and drop them in the Data Load Cart. You can also click **Select Files** or the Select Files icon, select one or more files from the file system navigator, and then click **Open**.

You can add files in these file formats: AVRO, CSV, JSON, GeoJSON, TSV, delimited TXT, XLS, XLSX, XML, PNG, JPEG and PDF. For information on supported file formats, see Format Specifications for JSON, AVRO, and XML Files.

An item for each file appears in the cart. For an XLS or XLSX spreadsheet, the worksheets of the spreadsheet appear as individual items. The item shows the name of the source file or worksheet and its size, and the name of the table that is the target for the data load. The Data Load tool supports loading tables from only the first worksheet of a multi-worksheet XLSX file when the file is in an object store.

You can add more files to the cart by clicking the Select Files icon. You can add any number of files to the cart and load data from all of them in a single data load job.

To remove a source file from the Data Load Cart, click the Remove (trash can) icon for the source item. To remove all source files from the cart, click the Remove All (trash can) icon in the Data Load Cart menu bar.

To return to the Data Load page, click **Data Load** above the Data Load Cart menu bar.

Specify Processing Options

To specify settings for the data load job or preview the data in the source or the target, select the Settings (pencil) icon for the item in the Data Load Cart.

In the settings pane, on the **Settings** tab, you can view the name and size of the file in the title of the Load Data dialog box.

The **Name** field specifies the name of the target table. The value in the field varies depending on the selection in the **Options** field. If the option is **Create Table**, then the default target value is the name of source file or worksheet.

To specify a different name for the target table, enter it in the **Name** field. For the other target table choices in the **Options** field, the default value is <None>. Expand the drop-down list and select an existing table as the target.

In the **Options** field select **Create Table**, **Insert into Table**, **Replace Data**, **Drop Table and Create New Table**, or **Merge into Table**. Point to the question mark icon to see a brief description of the selected option.

Select a different schema from the **Schema** drop-down to create your target table in another schema.

Note

The **Schema** drop-down is available only if you have `PDB_DBA` role grant to you. To grant yourself a `PDB_DBA` role, you must log into your Database Actions instance and enter the following command in the **SQL** worksheet area displayed in the **SQL** tab under **Development** tools present in the Launchpad.

```
Grant PDB_DBA to Username;
```

This drop-down is available for **Create Table** and **Drop Table and Create New Table** options.

The **Source column name** option specifies whether to get the source and target column names from the file or to specify the column names manually. Getting the column names from the header of the source file is the default. If you select the **Column header** option, then the first row in the file is processed as column names. If you deselect the option, then the first row is processed as data. To specify column names manually, enter a name for each target column in the **Mapping** section. You can also select a data type for the column.

The **Start processing data at row** field specifies how many rows to skip when loading the source data into the target. If you have selected the **Column header** option, and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows after the first row are not loaded into the target. If you have deselected the **Column header** option, and if you enter a number greater than 0 in the **Start processing data at row** field, that number of rows including the first row are not loaded into the target.

To change the character set encoding for the contents of the file, select a value from the **Encoding** drop-down list.

To specify the characters that enclose text, select the double-quotes or single-quote character or `None` from the **Text enclosure** drop-down list.

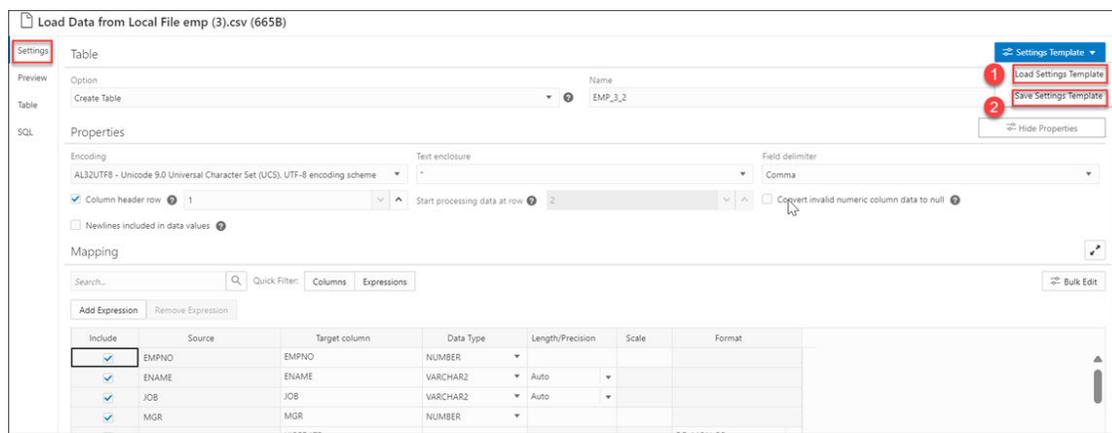
To change the delimiter character that separates columns in the source, expand the **Field delimiter** drop-down list and select a character. For example, if the file has columns delimited by semicolons, change the delimiter from the default comma delimiter to a semicolon.

To convert any invalid value in a numeric source column to a null value in the target column, select the Numeric column **Convert invalid data to null** option.

Settings Template

The save settings feature saves the configuration set in the Cart settings in the form of a JSON file. When opening the Settings template, you have the following options:

1. **Load Settings Template:** Loads a settings template from your local system.
2. **Save Settings Template:** Saves the current existing settings template.



You can use the **Load Settings Template** if you want to use an existing customized template present in your local.

1. From the Settings Template in the Settings tab of the Load Data page, select **Load Settings Template**.
2. You will see a Load Settings Template wizard, click the Settings Template JSON to load a JSON file from your system.
3. Clicking the Settings template JSON will open your local system. Click **OK** to load the JSON file.
4. After you load the JSON file, you can view the updates applied automatically to the settings tab which matches the JSON settings template you load from your local.

You can use the **Save Settings Template** to save the existing current Settings template.

1. From the Settings Template in the Settings tab of the Load Data page, select **Save Settings Template**.
2. The Template file editor appears. Click the Template File name and name the new template.
3. Click **OK** to finish saving the new name of the existing template.
4. You can test the configuration of the new template.

Bulk Edit Settings

You can use the Bulk edit settings to update all the columns at once from the mapping table. Use it to apply changes to the selection currently displayed in the results pane. You can search for the values of the column you want to edit in the search field and click the magnifier icon. The mapping table will display the results of the search. Select the Bulk Edit setting to update the column. The Bulk Edit setting allows you to:

- Update values of all the fields in a group.
- Find and replace, Add Prefix and Add suffix to target column name.
- Include the column(s) for loading data to the target table.
- Exclude the column(s) for loading data to the target table.

Searching the Column

The Bulk Edit setting updates the columns returned by the search field. The search box below the **Available Columns** on the **Bulk Edit Mapping** wizard filters the list of columns you wish to update in a bulk. You can also click the column you wish to select from the list of Available Columns. Choose any of the available options:

- >: This option enables you to move the column to Selected Columns.
- <: To remove the selected table from Selected Columns, select this option.
- >>: This option allows you to move all the tables to the Selected Columns screen.
- <<: To remove all the selected tables from Selected Columns select this option.

The Bulk Edit Mapping enables you to update the values of the following columns for all the searches returned by the search field:

- Data Type
- Target Column name
- Include Columns for loading
- Exclude columns for loading

Consider changing the Data Type of first five rows from VARCHAR2 to CHAR in the mapping table.

▼ **Mapping**

Search... Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source ↑	Target column ↑	Data Type	Length/Precision
<input checked="" type="checkbox"/>	Agency	AGENCY	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Claim	CLAIM	VARCHAR2 ▼	Auto ▼

- Select CHAR from the **Data Type** drop-down field.

Bulk Edit Mapping

1 Options 2 Preview

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type

CHAR

- CHAR **1**
- VARCHAR2
- CLOB
- NUMBER
- FLOAT
- BINARY_FLOAT
- BINARY_DOUBLE
- DATE
- TIMESTAMP

Back

- On the Available Columns list, click the columns you want to select. See [Searching the column](#) for more information.

Bulk Edit Mapping

1
2

Options **Preview**

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type

CHAR ▼

Length

Available Columns

- AGENCY
VARCHAR2
- AGENCY_TYPE
VARCHAR2
- DISTRIBUTION_CHANNEL
VARCHAR2
- PRODUCT_NAME
VARCHAR2
- CLAIM
VARCHAR2

Selected Columns

No items to display.

>>

>

<

<<

- You can view the columns you want to edit on the **Selected Columns** list.

Bulk Edit Mapping

1 Options 2 Preview

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type
CHAR

Length

Available Columns

Search...

No items to display.

Selected Columns

- AGENCY
VARCHAR2
- AGENCY_TYPE
VARCHAR2
- DISTRIBUTION_CHANNEL
VARCHAR2
- PRODUCT_NAME
VARCHAR2
- CLAIM
VARCHAR2

Back Next OK Cancel

Click **Next** to progress to the **Preview** tab.

- You can view the updated table in the Table Preview.

Bulk Edit Mapping

1 Options 2 Preview

Table Preview

Column Name	Data Type	Length/Precision	Format
AGENCY	CHAR		
AGENCY_TYPE	CHAR		
DISTRIBUTION_CHANNEL	CHAR		
PRODUCT_NAME	CHAR		
CLAIM	CHAR		

Back Next OK Cancel

Click **OK**.

- The **Mapping** table displays that the Data Type of the selected rows has changed from VARCHAR2 to CHAR.

Mapping

Search... Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source	Target column	Data Type
<input checked="" type="checkbox"/>	Agency	AGENCY	CHAR
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	CHAR
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	CHAR
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	CHAR
<input checked="" type="checkbox"/>	Claim	CLAIM	CHAR

Close

Specify Mappings

You can filter the results in the mapping table with Quick Filter field that enables you to filter out **Columns** or **Expressions**.

Select **Add Expression** to perform **Sentiment Analysis** or **Key Phrase extraction** with the source data. See [Use OCI Language Service Capabilities in Data Studio](#) for more details.

If you select the **Create Table** or **Drop Table and Create New Table** option and you are getting the source column names from the file header, then in the **Mapping** section either accept the default values for the target columns and data types or specify different values. To specify different values, in the target column, enter a name for the column. In the Data Type column, select a data type from the drop-down list. If you are not getting the source column names from the file header, then for each source column specify a name for the target column and select a data type from the Data Type drop-down list. For the Date data type, select a date format from the Format drop-down list.

Note

You will receive a tooltip error message with the exact reason for the error message when you complete editing a cell. The mapping grid cell will be highlighted with red to indicate an invalid value that must be fixed. The highlight is removed after you fix the invalid value. For example, you can view the following tooltip error message when the target column name is not filled in.

Include	Source column	Target column				
<input checked="" type="checkbox"/>	NAME		VARCHAR2	▼	Auto	▼
<input checked="" type="checkbox"/>	PLATFORM	PLATFORM	VARCHAR2	▼	Auto	▼

The target column name cannot be empty. A value is required.

For the **Merge into Table** option, for each source column, select a target column from the drop-down list. You must specify at least one column as a key column. To specify a column as a key column, select the **Merge Key** check box for the column. Merge keys are one or more columns that uniquely identify each row in the table. Merge keys must not contain any null values. For loading tables with primary keys, this option automatically enables the selection of primary key columns as the merge keys.

For the **Insert into Table** or **Replace Data** options, for each source column, select a target column from the drop-down list of existing columns.

Preview Source Data

To view a selection of data in the source file, select the **Preview** tab. The Preview tab displays a File menu which displays the data in tabular format with its values. You can copy the table. The **Load Preview** tab displays the source data.

Any modifications you make in the source preview do not affect the loading of data from the file.

Preview Target Data

For all options except **Create Table**, to view the existing data in the target table, select the **Load Preview** tab. The load preview displays the data in the target table before you run the data load job.

To close the settings pane, click **Close**.

Run the Data Load Job

When you have added all of the sources for the job and specified the settings for each source, to run the job click the Start icon in the Data Load Cart menu bar. In the Run Data Load Job dialog box, click **Start**. To stop the data load job, click the Stop icon.

Once the data load job starts, you can view the progress of the job in the Data Load dashboard. See [The Data Load Page](#) for more details.

When the data load job completes, the Data Load dashboard page displays the results of the job. At the top of the header of the table load, you can view the name of the table along with the total columns present in the table.

Click **Report** to view the total number of rows processed successfully and the count of rejected rows. You can also view the Start time. The SQL pane of the Report displays the equivalent SQL code of the job.

To view information about an item in the job, click the **Actions** icon on the Table Load.

To view a log of the load operation, click the Logging icon. You can save the log, clear it, or refresh it. Click **OK** to dismiss the log.

The list of tables on the Data Load page contains any new tables created. The target tables for the **Insert into Table**, **Replace Data**, **Drop Table and Create New Table**, and **Merge into Table** options contain the loaded data.

Fixing a data load job

After your data load job, you might see errors that you want to correct, or upon inspection, realize that you wanted to name a column differently. In such cases, you will view a warning sign on the Table load. Click the **Reload** icon to reload source with suggested fixes. Click **Actions** icon on the Table load and select **Edit** to make any changes to the data load job (i.e., change a column name).

Click **Apply** to apply any changes. Click **Close** to return to the Database Actions page.

Loading Data from Other Databases

To load data from tables in another database into your Oracle Autonomous AI Database, on the Data Load page, select **LOAD DATA** and **DATABASE**. Select a database link from the drop-down list. Drag one or more tables from the list of database tables and drop them in the Data Load Cart.

Each table appears as an item in the Data Load Cart. The item shows the name of the table and the number of rows in it, and the name of the table that is the target for the data load.

To remove a table from the Data Load Cart, click the Remove (trash can) icon for the item. To remove all tables from the cart, click the Remove All (trash can) icon in the Data Load Cart menu bar.

To add a remote database to the list of database links, create a database link to the remote database. For information on creating a database link, see Database Links in *Oracle AI Database Administrator's Guide*.

The databases available to you appear in the drop-down list of the database navigation pane of the Load Tables page.

You can filter the tables displayed in the navigation pane by entering a case-sensitive value in the search field at the top of the navigation tree and pressing Enter. To display all of the tables again, clear the search field and press Enter.

You can add any number of tables from the navigation pane to the Data Load Cart and load data from all of them in a single data loading job. You can set filters on the data for a table to load only the specified data.

Specify Processing Options

To specify settings for the data load job, preview the data in the source or the target, and see statistics about the data, click the Settings (pencil) icon for the item in the Data Load Cart.

In the settings pane, on the Settings tab, you can view the name and size of the file in the title of the Load Data dialog box.

The **Table** field specifies the name of the target table. The value in the field varies depending on the selection in the **Options** field. If the option is **Create Table**, then the default target value is the name of source table. To specify a different name for the target, enter it in the **Name** field. For the other options, the default value is <None>. Expand the drop-down list and select a table as the target.

In the **Options** field for the source, select **Create Table**, **Insert into Table**, **Replace Data**, **Drop Table and Create New Table**, or **Merge into Table**. Point to the question mark icon to see a brief description of the selected option.

Select a different schema from the **Schema** drop-down to create your target table in another schema.

Note

The **Schema** drop-down is available only if you have `PDB_DBA` role grant to you.

To grant yourself a `PDB_DBA` role, you must log into your Database Actions instance and enter the following command in the **SQL** worksheet area displayed in the **SQL** tab under **Development** tools present in the Launchpad.

```
Grant PDB_DBA to Username;
```

This drop-down is available for **Create Table** and **Drop Table and Create New Table** options.

If you select **Create Table**, then in the **Name** field accept the default name, which is the name of the source table, or enter a different name.

If you select one of the other options, then expand the drop-down list of the **Name** field and select a table as the target.

Settings Template

The save settings feature saves the configuration set in the Cart settings in the form of a JSON file. When opening the Settings template, you have the following options:

1. **Load Settings Template:** Loads a settings template from your local system.
2. **Save Settings Template:** Saves the current existing settings template.

The screenshot shows the 'Load Data from Local File emp (3).csv (665B)' configuration page. The 'Settings' tab is selected, and the 'Table' section shows 'Create Table' as the option and 'EMP_3_2' as the name. The 'Properties' section includes encoding (AL32UTF8), text enclosure (comma), and column header row (1). The 'Mapping' section shows a table with columns EMPNO, ENAME, JOB, and MGR being mapped to the target table. The 'Settings Template' dropdown menu is open, showing 'Load Settings Template' and 'Save Settings Template' options.

You can use the **Load Settings Template** if you want to use an existing customized template present in your local.

1. From the Settings Template in the Settings tab of the Load Data page, select **Load Settings Template**.
2. You will see a Load Settings Template wizard, click the Settings Template JSON to load a JSON file from your system.

3. Clicking the Settings template JSON will open your local system. Click **OK** to load the JSON file.
4. After you load the JSON file, you can view the updates applied automatically to the settings tab which matches the JSON settings template you load from your local.

You can use the **Save Settings Template** to save the existing current Settings template.

1. From the Settings Template in the Settings tab of the Load Data page, select **Save Settings Template**.
2. The Template file editor appears. Click the Template File name and name the new template.
3. Click **OK** to finish saving the new name of the existing template.
4. You can test the configuration of the new template.

Bulk Edit Settings

You can use the Bulk edit settings to update all the columns at once from the mapping table. Use it to apply changes to the selection currently displayed in the results pane. You can search for the values of the column you want to edit in the search field and click the magnifier icon. The mapping table will display the results of the search. Select the Bulk Edit setting to update the column. The Bulk Edit setting allows you to:

- Update values of all the fields in a group.
- Find and replace, Add Prefix and Add suffix to target column name.
- Include the column(s) for loading data to the target table.
- Exclude the column(s) for loading data to the target table.

Searching the Column

The Bulk Edit setting updates the columns returned by the search field. The search box below the **Available Columns** on the **Bulk Edit Mapping** wizard filters the list of columns you wish to update in a bulk. You can also click the column you wish to select from the list of Available Columns. Choose any of the available options:

- **>**: This option enables you to move the column to Selected Columns.
- **<**: To remove the selected table from Selected Columns, select this option.
- **>>**: This option allows you to move all the tables to the Selected Columns screen.
- **<<**: To remove all the selected tables from Selected Columns select this option.

The Bulk Edit Mapping enables you to update the values of the following columns for all the searches returned by the search field:

- Data Type
- Target Column name
- Include Columns for loading
- Exclude columns for loading

Consider changing the Data Type of first five rows from VARCHAR2 to CHAR in the mapping table.

Mapping

Search... Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source ↑	Target column ↑	Data Type	Length/Precision
<input checked="" type="checkbox"/>	Agency	AGENCY	VARCHAR2	Auto
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	VARCHAR2	Auto
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	VARCHAR2	Auto
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	VARCHAR2	Auto
<input checked="" type="checkbox"/>	Claim	CLAIM	VARCHAR2	Auto

- Select CHAR from the **Data Type** drop-down field.

Bulk Edit Mapping

Options Preview

Apply the same action to all the columns selected

Data Type Target Column Name Include column Exclude column

Data Type

CHAR

CHAR

VARCHAR2

CLOB

NUMBER

FLOAT

BINARY_FLOAT

BINARY_DOUBLE

DATE

TIMESTAMP

Back

- On the Available Columns list, click the columns you want to select. See [Searching the column](#) for more information.

Bulk Edit Mapping

1
2

Options
Preview

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type

CHAR ▼

Length

Available Columns

- AGENCY
VARCHAR2
- AGENCY_TYPE
VARCHAR2
- DISTRIBUTION_CHANNEL
VARCHAR2
- PRODUCT_NAME
VARCHAR2
- CLAIM
VARCHAR2

>>

>

<

<<

Selected Columns

No items to display.

- You can view the columns you want to edit on the **Selected Columns** list.

Bulk Edit Mapping

1 **Options** 2 **Preview**

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type
CHAR

Length

Available Columns
Search...
No items to display.

Selected Columns

- AGENCY
VARCHAR2
- AGENCY_TYPE
VARCHAR2
- DISTRIBUTION_CHANNEL
VARCHAR2
- PRODUCT_NAME
VARCHAR2
- CLAIM
VARCHAR2

Back **Next** OK Cancel

Click **Next** to progress to the **Preview** tab.

- You can view the updated table in the Table Preview.

Bulk Edit Mapping

1 Options 2 Preview

Table Preview

Column Name	Data Type	Length/Precision	Format
AGENCY	CHAR		
AGENCY_TYPE	CHAR		
DISTRIBUTION_CHANNEL	CHAR		
PRODUCT_NAME	CHAR		
CLAIM	CHAR		

Back Next OK Cancel

Click **OK**.

- The **Mapping** table displays that the Data Type of the selected rows has changed from VARCHAR2 to CHAR.

Mapping

Search... Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source	Target column	Data Type
<input checked="" type="checkbox"/>	Agency	AGENCY	CHAR
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	CHAR
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	CHAR
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	CHAR
<input checked="" type="checkbox"/>	Claim	CLAIM	CHAR

Close

Specify Mappings

If you select the **Create Table** or the **Drop Table and Create New Table** option, then in the **Mapping** section either accept the default values for the target columns or specify different values. For the target column, enter a name for the column.

Note

You will receive a tooltip error message with the exact reason for the error message when you complete editing a cell. The mapping grid cell will be highlighted with red to indicate an invalid value that must be fixed. The highlight is removed after you fix the invalid value. For example, you can view the following tooltip error message when the target column name is not filled in.

Include	Source column	Target column			
<input checked="" type="checkbox"/>	NAME		VARCHAR2	▼	Auto
<input checked="" type="checkbox"/>	PLATFORM	PLATFORM	VARCHAR2	▼	Auto

The target column name cannot be empty. A value is required.

For the **Insert into Table** or **Replace Data** options, select a target column from the drop-down list of existing columns.

For the **Merge into Table** option, for each source column, select a target column from the drop-down list. You must specify at least one column as a key column. To specify a column as a key column, select the **Merge Key** check box for the column. Merge keys are one or more columns that uniquely identify each row in the table. Merge keys must not contain any null values. For loading tables with primary keys, this option automatically enables the selection of primary key columns as the merge keys.

Preview

To view the data in the source table, in the settings pane select the **Source Table** tab. The source preview displays the data in the table.

Table

For all options except **Create Table**, to view the existing data in the target table, in the settings pane select the **Target Table** tab. The target preview displays the data in the table before you run the data load job.

SQL

The SQL tab displays the SQL commands that will be run to complete this data load job.

Note

You can see the SQL code even before the table is created.

Run the Data Load Job

When you have added all of the source tables for the job and specified the settings for each table, to run the job click the Start icon in the Data Load Cart menu bar. To stop the data load job, click the Stop icon.

Once the data load job starts, you can view the progress of the job in the Data Load dashboard.

When the data load job completes, the Data Load dashboard page displays the results of the job. At the top of the header of the table load, you can view the name of the table along with the total columns present in the table.

Click **Report** to view a report of the total rows loaded and failed for a specific table. You can view the name of the table, the time the table was loaded and the time taken to process the load.

At the header of the Table Load, you can view the name of the table with total number of columns loaded. When you expand the Table Load you can view the options you used to load the source data with the count of rows loaded.

To view information about an item in the job, click the Settings icon in the item. The settings pane has the same **Settings, Source, Table, SQL, Job Report** and **Data Definition** tabs as the settings pane before running the job, except that the target preview now contains the data loaded by the data load job. To close the settings pane, click **Close**.

To view a log of the load operation, click the Log icon. You can save the log, clear it, or refresh it. Click **OK** to dismiss the log.

The list of tables on the Data Load / Explore page contains any new tables created. The target tables for the **Insert into Table, Replace Data, Drop Table and Create New Table**, and **Merge into Table** options contain the loaded data.

Fixing a data load job

After your data load job, you might see errors that you want to correct, or upon inspection, realize that you wanted to name a column differently. In such cases, you will view a warning sign on the Table load. Click the **Reload** icon to reload source with suggested fixes. Click **Actions** icon on the Table load and select **Edit** to make any changes to the data load job (i.e., change a column name).

Click **Apply** to apply any changes. Click **Close** to return to the Database Actions page.

- [Load Data from Oracle and Non-Oracle Databases using Database Links](#)
This section of the document describes how to create database links and then transfer data from various non-Oracle databases into your Autonomous AI Database using Data Studio.

Load Data from Oracle and Non-Oracle Databases using Database Links

This section of the document describes how to create database links and then transfer data from various non-Oracle databases into your Autonomous AI Database using Data Studio.

To load data from Oracle and Non-Oracle Databases using Database Links, you must first create a Database Link.

Run the following steps to load data from Oracle databases, or non-Oracle databases into Autonomous AI Database using Database Links:

1. Log in to your Database Actions instance. On the **Development** menu, click **SQL**. You can view the SQL Worksheet.
2. **Create Credentials and Database Links to access the Autonomous AI Database.**

You will use the `DBMS_CLOUD.CREATE_CREDENTIAL` procedure to create credentials where the `username` and `password` values you specify are the credentials for the target database.

The `DBMS_CLOUD.CREATE_CREDENTIAL` procedure stores the credentials in an encrypted format.

After you create credentials, you will create database links to load data in Data Studio from non-Oracle Databases using the Data Load tool.

Following are the sample codes you will run to create credentials and then use the same credentials to create the database links:

a. Create Database Links from MySQL:

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'MYSQLPE_CRED',
    username => 'admin',
    password => <enter password here> );

  DBMS_CLOUD_ADMIN.CREATE_DATABASE_LINK(
    db_link_name => 'MYSQLPE_DBLINK',
    hostname => 'scottmysql.scottprivate.com',
    port => '3306',
    service_name => 'scott',
    ssl_server_cert_dn => NULL,
    credential_name => 'MYSQLPE_CRED',
    private_target => TRUE,
    gateway_params => JSON_OBJECT('db_type' value 'MYSQL'));
END;
/
```

b. Create Database Links from Microsoft Azure:

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'AZURE_CRED',
    username => 'oracle',
    password => <enter password here> );

  DBMS_CLOUD_ADMIN.CREATE_DATABASE_LINK(
    db_link_name => 'AZURE_DBLINK',
    hostname => 'scott-server.database.windows.net',
    port => '1433',
    service_name => 'scottdb',
    ssl_server_cert_dn => NULL,
    credential_name => 'AZURE_CRED',
    gateway_params => JSON_OBJECT('db_type' value 'AZURE'));
END;
/
```

c. Create Database Links from Snowflake:

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'SNOWFLAKE_CRED',
    username => 'SCOTT',
    password => <enter password here> );

  DBMS_CLOUD_ADMIN.CREATE_DATABASE_LINK(
    db_link_name => 'SNOWFLAKE_DBLINK',
    hostname => 'https://abcdefg-hijl2345.snowflakecomputing.com',
    port => '443',
    service_name => 'SCOTTDB',
    credential_name => 'SNOWFLAKE_CRED',
```

```

gateway_params => JSON_OBJECT('db_type' value 'SNOWFLAKE'),
directory_name => NULL,
ssl_server_cert_dn => NULL);
END;
/

```

d. Create Database Links from Oracle Cloud Object Storage using Native OCI Credentials

```

BEGIN
  DBMS_CLOUD.GET_OBJECT(
    object_uri => 'https://objectstorage.us-ashburn-1.oraclecloud.com/n/
abcdefghijklmnop12ij/b/scott-bucket/o/cwallet.sso',
    credential_name => 'OCI_NATIVE_CRED',
    directory_name => 'REG_WALLET_DIR');

  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'ADBS_CRED',
    username => 'ADMIN',
    password => <enter password here> );

  DBMS_CLOUD_ADMIN.CREATE_DATABASE_LINK(
    db_link_name => 'ADBS_DBLINK',
    hostname => 'adb.us-ashburn-1.oraclecloud.com',
    port => '1522',
    service_name =>
'abcdefghijklmnop12ijkl_scottadbs_high.adb.oraclecloud.com',
    credential_name => 'ADBS_CRED',
    directory_name => 'REG_WALLET_DIR');
END;
/

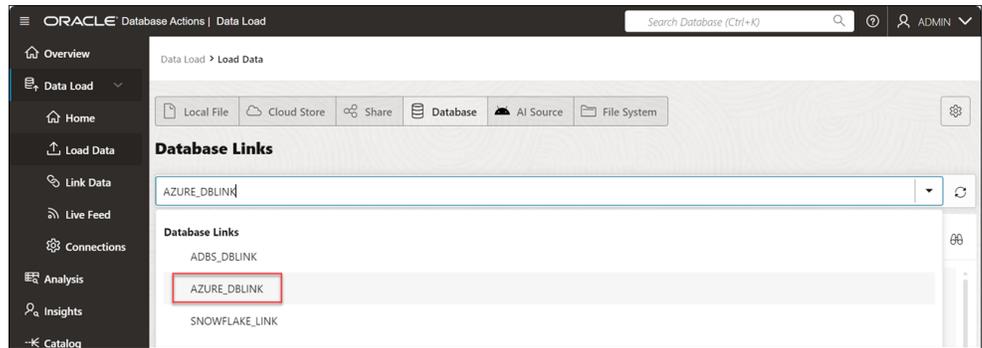
```

For more details on the parameters and their values, see the [Create Database Link](#) procedure.

① Note

- You require ADMIN privileges to run the DBMS_CLOUD_ADMIN.CREATE_DATABASE_LINK procedure.
- You can use a vault secret credential for the target database credential in a database link. See [Use Vault Secret Credentials](#) for more information.

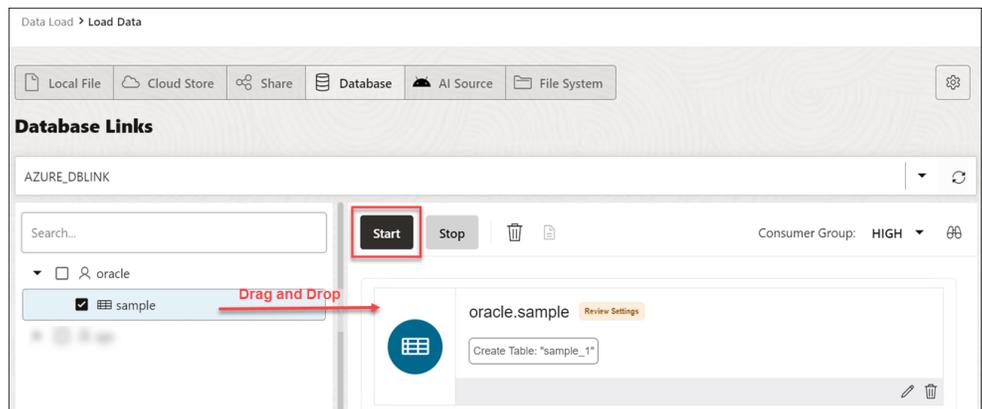
- Use the Data Load tool to LOAD data from Oracle and non-Oracle Databases:**
 You can use the database link you created to load access data on Data Studio. After you create the database links, you can view and use the list of database links you created in the Data Load tool and access data from Oracle and non-Oracle databases.
 - Click **Data Load** under the Data Studio suite of tools and select the **Load Data** card.
 - Click the **Database** tab and select the database link from the list of available database links you created. In this example, we will use AZURE_DBLINK.



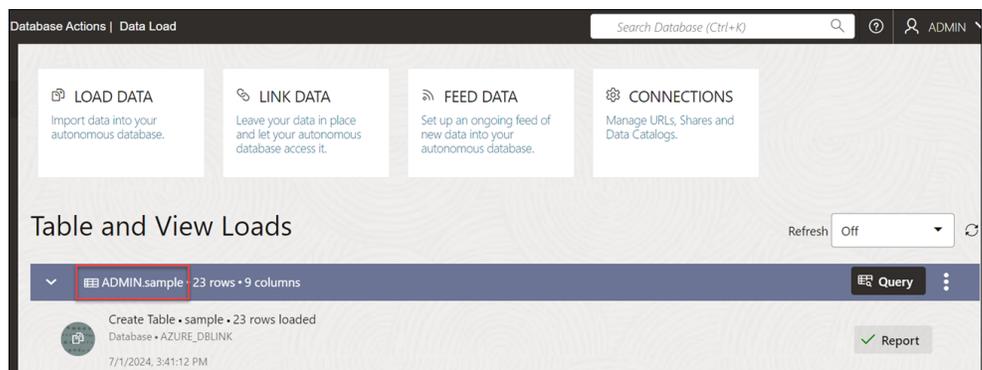
Note

You cannot view the database link you create from **MySQL** since it resides within a Virtual Cloud Network (VCN), and the Autonomous AI Database would need to be either located in or connected to the Private Subnet within that VCN.

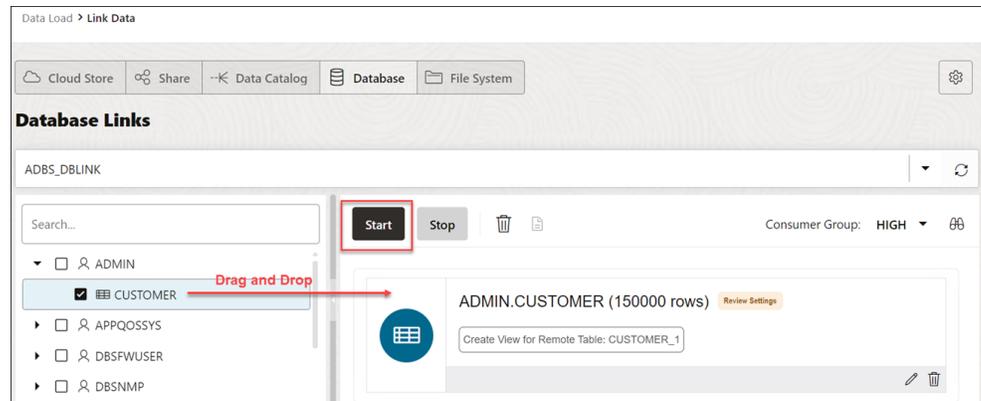
- Drag the `sample` table from the database and drop it in the data load cart to load the data in your Autonomous AI Database.



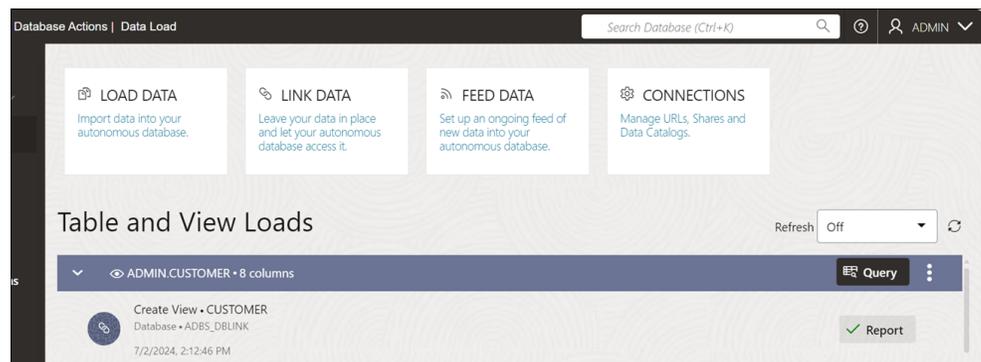
- Click **Start** to start the loading process.
- After the table is loaded in the tool, you can view the result in the [Data Load Dashboard](#) under the **Table and View Loads** section.



- b. **Use the Data Load tool to LINK data from Oracle and non-Oracle Databases:**
Alternatively, you can use the Data Load to link data that creates a view of the data by directly accessing cloud databases via the Database Link.
- From the **Data Load** menu under Data Studio, select **Link Data**.
 - Click the **Database** tab. In this example, we will use the `ADBS_DBLINK` link.
 - Drag the `customer` table from the database and drop it in the data link cart to link the data in your Autonomous AI Database.



- Click **Start** to start linking the data in Data Studio.
- After the data is linked, you will view the results of the data link job in the [Data Load dashboard](#) under the **Table and View Loads** section.



You have successfully loaded the data from Oracle and non-Oracle databases using Database Links in Data Studio.

Loading Data from Cloud Storage

You can load data from a cloud store to a table in your Autonomous AI Database.

You can load files in these file formats: AVRO, CSV, JSON, GeoJSON, ORC, Delimited TXT, XLSX, PRQ, GZ, GNU ZIP, Tab-Separated Values and Oracle Business Intelligence Cloud Connector (BICC) extracts. The main output of a BICC extract is one or more CSV files that contain the extracted business data in a tabular format. Alongside the CSV files, there may be metadata or manifest files used internally by BICC or downstream processes to track the

extract details, but the primary deliverables are the CSV data files. For information on supported file formats, see [Format Specifications for JSON and AVRO Files](#). The Data Load tool supports loading tables from multiple worksheets XLSX files when the file is in a cloud store.

You can set filters on the data for a table to load only the specified data. For example, to limit the files to only those that are CSV files, enter *.CSV in the file extension filter.

Configure and run a data load job from the Load Cloud Object page. To open that page:

1. Open the Database Actions and select **Data Load**.
2. Select **LOAD DATA** and **Cloud Store**.

On the left side of the page is a *navigator pane*, where you choose a cloud store connection and the folders or files containing the data. On the right of the page is the data load *cart*, where you stage the files and folders for the data load job. You can set options for the data load job before running it. The Autonomous AI Database comes with predefined CPU/IO shares assigned to different consumer groups. You can set the consumer group to either low, medium or high while executing a data load job depending on your workload.

To load files from a cloud store into your database, do the following:

- [Manage Cloud Storage Links for Data Load Jobs](#)
- [Prepare the Data Load Job](#)
- [Add Files or Folders for the Data Load Job](#)
- [Enter Details for the Data Load Job](#)
- [Run the Data Load Job](#)
- [View Details About the Data Load Job After It Is Run](#)
- [View the Table Resulting from the Data Load Job](#)

- [Manage Cloud Storage Links for Data Load Jobs](#)

Before you can load data from a cloud store, you must establish a connection to the cloud store you want to use. You can select cloud store location from the cloud store locations field.

- [Prepare the Data Load Job](#)
- [Add Files or Folders for the Data Load Job](#)
- [Enter Details for the Data Load Job](#)

Enter the details about the data load job in the Load Data from Cloud Storage pane.

- [Run the Data Load Job](#)
Once you've added data sources to the data load cart and entered details about the data load job, you can run the job.
- [View Details About the Data Load Job After It Is Run](#)
- [View the Table Resulting from the Data Load Job](#)

Manage Cloud Storage Links for Data Load Jobs

Before you can load data from a cloud store, you must establish a connection to the cloud store you want to use. You can select cloud store location from the cloud store locations field.

On the Load Data page when you select Cloud Store:

1. Click the **Create Cloud Store locations** menu besides the cloud store locations text field. This opens an Add Cloud Store Location dialog box. See [Managing Connections](#) to add cloud store location.

See [Managing Connections](#).

To return to the Load Cloud Object page, click **Data Load** in the breadcrumbs at the top of the page and then navigate back to the page.

Prepare the Data Load Job

As you'll see below in [Enter Details for the Data Load Job](#), the first decision you'll make when configuring a data load job is how to load the source data into a new or existing table in the database. The choices are:

- Create a table and insert data loaded from the source into the new table.
- Insert data loaded from the source into an existing table.
- Delete all data in an existing table and then insert new data from the source into the table.
- Drop a table, create a new table, and then insert data loaded from the source into the new table.
- Merge data from the source into a table by updating existing rows in the table and inserting new rows into the table.

You may have to adjust your source data or your target table so that the source data loads correctly into the external target table. The number, order, and data types of columns in the source must match those in the target. Consider:

- If you're creating a new table or if the columns in your source exactly match the columns in an existing target, you don't have to do any special preparation.
- If the columns in your source don't match the columns in an existing target, you must edit your source files or target table so they do match.
- If you're loading multiple files, you must make sure that:
 - All the source files are of the same type, for example, CSV, JSON, etc.
 - The number, order, and data types of the columns in all the source files match (and that they match the target, if you're loading into an existing table).
- If you want to partition by date:
 - The source file must contain data where the data type is date or timestamp.
 - You must load a folder containing two or more data sources.
 - The names of the files in the folder must indicate a date or dates, for example, `MAR-1999.csv` or `2017-04-21.xlsx`.

Add Files or Folders for the Data Load Job

Add files from the cloud store to the data load cart, where you can edit the details of the data load job. To add the files:

1. From the list at the top of the navigator pane on the left, select the bucket from the drop-down with your source data.

The list shows links that were established on the Manage Cloud Storage page. If you haven't yet registered the cloud store you want to use, click the **Create Cloud Store**

Location button at the top of the page and register a connection. See [Manage Cloud Storage Links for Data Load Jobs](#), above.

2. Drag one or more items from the file navigator on the left and drop them into the cart on the right.
 - You can add files, folders, or both. A card is added to the cart for each file or folder you drag into it. The card lists the name of the source file or folder and a proposed name for the target table.
 - If you add a folder that contains multiple files, all the files must be of the same type, that is, CSV, TXT, etc.

When you add the folder to the cart, a prompt is displayed that asks if you want to load all the objects from the multiple source files into a single target table. Click **Yes** to continue or **No** to cancel.

- When you add multiple individual files or multiple folders to the cart, the data represented by each card will be loaded into a separate table, but all the items in the cart will be processed as part of the same data load job. The multiple files you load must have same file extension.
- You can add files or folders from a different bucket, but if you do that, you're prompted to remove all files that are already in the cart before proceeding. To select files from a different bucket, select the bucket from the drop-down list in the navigator pane on the left and then add the file(s), as described above.
- You can drop files or folders into the data load cart and then navigate away from the Data Load Object page. When you return to the page, those items remain on the page.

You can remove items from the cart before running the data load job:

- To remove an item from the cart, click the **Remove** icon on the card for the item.
- To remove all items from the cart, click **Remove All** in the data link cart menu bar at the top of the pane besides the **Start** and **Stop** icon.

Enter Details for the Data Load Job

Enter the details about the data load job in the Load Data from Cloud Storage pane.

On the card in the data link cart, click the **Actions** icon and select the **Settings** to open the Load Data from Cloud Storage pane for that job. The pane contains:

- [Settings Tab - Table Section](#)
- [Settings Tab - Properties Section](#)
- [Settings Tab - Mapping Section](#)
- [Preview Tab](#)
- [Table Tab](#)
- [Close Button - Save and Close the Pane](#)

Settings Tab - Table Section

Set details about the target table in the **Table** section.

- **Option:** Select an item from the **Option** list to specify how the data should be loaded into a new or existing table. The processing options are:

- **Create Table:** Creates a table and inserts the data into the new table. When you select this option, the **Name** field on the **Settings** tab is filled with a default name, based on the name of the source file or folder. You can change it if you want.
 - **Insert into Table:** Inserts data loaded from the source into an existing table. When you select this option, the **Name** field on the **Settings** tab presents a list of the tables in the current schema. Select the table into which you want to insert the data.
 - **Replace Data:** Deletes all data in the existing table and then inserts new data from the source into the table. When you select this option, the **Name** field on the **Settings** tab presents a list of the tables in the current schema. Select the table you want to use.
 - **Drop Table and Create New Table:** Drops the table (if it already exists), creates a new table, and then inserts the new data into the table. When you select this option, the **Name** field on the **Settings** tab presents a list of the tables in the current schema. Select the table you want to use.
 - **Merge into Table:** Updates existing rows and inserts new rows in the table. When you select this option, the **Name** field on the **Settings** tab presents a list of the tables in the current schema. Select the table you want to use.
- Select a different schema from the **Schema** drop-down to create your target table in another schema.

Note

The **Schema** drop-down is available only if you have `PDB_DBA` role grant to you.

To grant yourself a `PDB_DBA` role, you must log into your Database Actions instance and enter the following command in the **SQL** worksheet area displayed in the **SQL** tab under **Development** tools present in the Launchpad:

```
Grant PDB_DBA to USER;
```

You can now view the **Schema** drop-down. It is only available for **Create Table** and **Drop Table and Create New Table** options.

- **Name:** The name of the target table.
- **Partition Column:**

List Partitions and Date-based partitions are the different types of partitions available in data loading.

List partitioning is required when you specifically want to map rows to partitions based on discrete values.

To partition according to a specific column, click the **Partition Column** drop-down list and select the column you want to use for the partitioning.

You will have N files per partition value, all partitioned by the partition column you select.

Note

- For linked files (from external tables) there is also a requirement that for each file, the list partitioning column can contain only a single distinct value across all of the rows.
- If a file is list partitioned, the partitioning key can only consist of a single column of the table.

Date-based partitioning is available when you load a folder containing two or more data sources that contain date or timestamp data.

To partition according to date, click the **Partition Column** drop-down list and select the **DATE** or **TIMESTAMP** column you want to use for the partitioning.

Settings Tab - Properties Section

Specify options to control how the source data is interpreted, previewed, and processed. These options vary, depending on the type of source data.

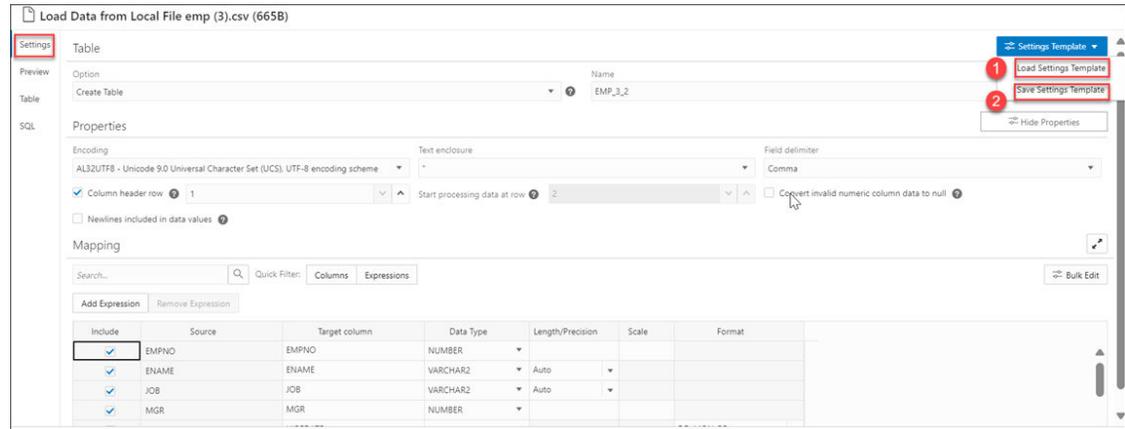
- **Encoding:** Select a character encoding type from the list. This option is available when the loaded file is in plain text format (CSV, TSV, or TXT). The default encoding type is UTF-8.
- **Text enclosure:** Select the character for enclosing text: " (double-quote character), ' (single-quote character) or **None**. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Field delimiter:** Select the delimiter character used to separate columns in the source. For example, if the source file uses semicolons to delimit the columns, select **Semicolon** from this list. The default is **Comma**. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Column header row:** Select the **Column header row** checkbox to use the column names from the source table in the target table.
 - By selecting this option you can indicate what row in the file contains column names. The rows in the **Mapping** section, below, are filled with those names (and with the existing data types, unless you change them).
 - If you deselect this option, the first row is processed as data. To specify column names manually, enter a name for each target column in the **Mapping** section. (You will also have to enter data types.)
- **Start processing data at row:** Specifies the number of rows to skip when loading the source data into the target:
 - If you select the **Column header row** option under **Source column name** (see below) and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows after the first row are not loaded into the target.
 - If you deselect the **Column header row** option under **Source column name**, and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows including the first row are not loaded into the target.
- **Numeric column:** Select the **Convert invalid data to null** checkbox to convert an invalid numeric column value into a null value.
- **Newlines included in data values:** Select this option if there are newline characters or returns to the beginning of the current line without advancing downward in the data fields. Selecting this option will increase the time taken to process the load. If you do not select

this option when loading the data, the rows with newlines in the fields will be rejected. You can view the rejected row in the Job Report panel.

Settings Template

The save settings feature saves the configuration set in the Cart settings in the form of a JSON file. When opening the Settings template, you have the following options:

1. **Load Settings Template:** Loads a settings template from your local system.
2. **Save Settings Template:** Saves the current existing settings template.



You can use the **Load Settings Template** if you want to use an existing customized template present in your local.

1. From the Settings Template in the Settings tab of the Load Data page, select **Load Settings Template**.
2. You will see a Load Settings Template wizard, click the Settings Template JSON to load a JSON file from your system.
3. Clicking the Settings template JSON will open your local system. Click **OK** to load the JSON file.
4. After you load the JSON file, you can view the updates applied automatically to the settings tab which matches the JSON settings template you load from your local.

You can use the **Save Settings Template** to save the existing current Settings template.

1. From the Settings Template in the Settings tab of the Load Data page, select **Save Settings Template**.
2. The Template file editor appears. Click the Template File name and name the new template.
3. Click **OK** to finish saving the new name of the existing template.
4. You can test the configuration of the new template.

Bulk Edit Settings

You can use the Bulk edit settings to update all the columns at once from the mapping table. Use it to apply changes to the selection currently displayed in the results pane. You can search for the values of the column you want to edit in the search field and click the magnifier icon. The mapping table will display the results of the search. Select the Bulk Edit setting to update the column. The Bulk Edit setting allows you to:

- Update values of all the fields in a group.
- Find and replace, Add Prefix and Add suffix to target column name.

- Include the column(s) for loading data to the target table.
- Exclude the column(s) for loading data to the target table.

Searching the Column

The Bulk Edit setting updates the columns returned by the search field. The search box below the **Available Columns** on the **Bulk Edit Mapping** wizard filters the list of columns you wish to update in a bulk. You can also click the column you wish to select from the list of Available Columns. Choose any of the available options:

- >: This option enables you to move the column to Selected Columns.
- <: To remove the selected table from Selected Columns, select this option.
- >>: This option allows you to move all the tables to the Selected Columns screen.
- <<: To remove all the selected tables from Selected Columns select this option.

The Bulk Edit Mapping enables you to update the values of the following columns for all the searches returned by the search field:

- Data Type
- Target Column name
- Include Columns for loading
- Exclude columns for loading

Consider changing the Data Type of first five rows from VARCHAR2 to CHAR in the mapping table.

▼ **Mapping**

Search... Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source ↑	Target column ↑	Data Type	Length/Precision
<input checked="" type="checkbox"/>	Agency	AGENCY	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	VARCHAR2 ▼	Auto ▼
<input checked="" type="checkbox"/>	Claim	CLAIM	VARCHAR2 ▼	Auto ▼

- Select CHAR from the **Data Type** drop-down field.

Bulk Edit Mapping

1 Options 2 Preview

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type

CHAR

- CHAR **1**
- VARCHAR2
- CLOB
- NUMBER
- FLOAT
- BINARY_FLOAT
- BINARY_DOUBLE
- DATE
- TIMESTAMP

Back

- On the Available Columns list, click the columns you want to select. See [Searching the column](#) for more information.

Bulk Edit Mapping

1
2

Options
Preview

Apply the same action to all the columns selected

Data Type	Target Column Name	Include column	Exclude column
-----------	--------------------	----------------	----------------

Data Type

CHAR ▼

Length

Available Columns

- AGENCY
VARCHAR2
- AGENCY_TYPE
VARCHAR2
- DISTRIBUTION_CHANNEL
VARCHAR2
- PRODUCT_NAME
VARCHAR2
- CLAIM
VARCHAR2

>>

>

<

<<

Selected Columns

No items to display.

- You can view the columns you want to edit on the **Selected Columns** list.

Bulk Edit Mapping

1 Options 2 Preview

Apply the same action to all the columns selected

Data Type Target Column Name Include column Exclude column

Data Type
CHAR

Length

Available Columns
Search...
No items to display.

Selected Columns

- AGENCY VARCHAR2
- AGENCY_TYPE VARCHAR2
- DISTRIBUTION_CHANNEL VARCHAR2
- PRODUCT_NAME VARCHAR2
- CLAIM VARCHAR2

Back Next OK Cancel

Click **Next** to progress to the **Preview** tab.

- You can view the updated table in the Table Preview.

Bulk Edit Mapping

1 Options
2 Preview

Table Preview

Column Name	Data Type	Length/Precision	Format
AGENCY	CHAR		
AGENCY_TYPE	CHAR		
DISTRIBUTION_CHANNEL	CHAR		
PRODUCT_NAME	CHAR		
CLAIM	CHAR		

Back Next OK Cancel

Click **OK**.

- The **Mapping** table displays that the Data Type of the selected rows has changed from VARCHAR2 to CHAR.

Mapping

Quick Filter: Columns Expressions Errors

Add Expression Remove Expression Bulk Edit

Include	Source	Target column	Data Type
<input checked="" type="checkbox"/>	Agency	AGENCY	CHAR
<input checked="" type="checkbox"/>	Agency Type	AGENCY_TYPE	CHAR
<input checked="" type="checkbox"/>	Distribution Channel	DISTRIBUTION_CHANNEL	CHAR
<input checked="" type="checkbox"/>	Product Name	PRODUCT_NAME	CHAR
<input checked="" type="checkbox"/>	Claim	CLAIM	CHAR

Close

Settings Tab - Mapping Section

The settings in the **Mapping** section control how data from the source files are loaded into the rows of the target database table. For each row, the data from the column listed under **Source column** will be loaded into the column listed under **Target column**.

As mentioned above, the contents of the **Mapping** table change according to what processing option you chose in the **Table** section and which properties you set in the **Properties** section.

You can filter the results in the mapping table with **Quick Filter** field that enables you to filter out **Columns** or **Expressions**.

Select **Add Expression** to perform **Sentiment Analysis** or **Key Phrase extraction** with the source data. See [Use OCI Language Service Capabilities in Data Studio](#) for more details.

- Include:** This check box ensures that the row you select is loaded from the source column to the target column.

- **Source column:** Lists the columns from the source file.

If the **Column header row** option under **Properties** is selected, **Source column** shows the names of the columns in the source file. If the **Column header row** option is not selected, generic names like **COLUMN_1**, **COLUMN_2**, etc., are used. This field is always read only.

You can view two source columns `FILE$NAME` and `SYSTIMESTAMP`. The `FILE$NAME` column enables you to locate the source file containing a particular data record. For example, you load a source file that contains a list of files. The file names in the file list refer to the department names across the organization. For instance, a *finance.txt* file contains data from the Finance department. In the mapping, you can use string data types to extract the department name from the output of the file name column. You can use the extracted department name to process data differently for each department.

The `SYSTIMESTAMP` column allows us to view the current timestamp in the database.

Note

- `FILE$NAME` and `SYSTIMESTAMP` source columns are not included by default. You must check the **Include** check box and run the load for the target table to display these two columns.
- When you are creating a livefeed, the `FILE$NAME` and `SYSTIMESTAMP` source columns appear in the Mapping table by default.

- **Target column:** Lists the columns in the target table. Accept, select, or enter a column in the target table. You need to make sure that the target column is not empty. The target column name cannot have duplicate name as another target column. The target column length must not be beyond 128 bytes. 128 byte limit is a database limit.

The contents of this column differ, depending on what you selected for the table processing **Option** and whether you selected for the **Column header row** option.

- If (1) the processing option is **Create Table** or **Drop Table and Create New Table** and (2) the **Column header row** option *is* selected, then the **Target column** uses the names of the columns in the source file. You can change the name of a target column by replacing the provided name with a new one.
- If (1) the processing option is **Create Table** or **Drop Table and Create New Table** and (2) the **Column header row** option is *not* selected, then generic names like **COLUMN_1**, **COLUMN_2**, etc., are used. You can change the name of a target column by replacing the provided name with a new one.
- If (1) the processing option is **Insert into Table**, **Replace Data**, or **Merge Into Table** and (2) the **Column header row** option *is* selected, then the **Target column** has a drop-down list of all the columns in the target table, with their data types. By default, the column with the name corresponding to the source column is selected, but you can select a different one from the list.
- If (1) the processing option is **Insert into Table**, **Replace Data**, or **Merge Into Table** and (2) the **Column header row** option is *not* selected, then the **Target column** has a drop-down list of all the columns in the target table, with their data types. Select a column from the list to use as the target column.

Note

If you're loading multiple files from a folder in a single data load job, only the first file will be shown in the **Mapping** section. However, as long as the column names and data types match, the data from all source files will be loaded.

- **Data Type:** Lists the data type to use for data in that column. This column is displayed only for **Create Table** or **Drop Table and Create New Table**. The contents change depending on whether the **Get from file header** option is selected.
 - If the **Get from file header** option is selected, **Data type** shows the data types of the columns in the source file (for **Create Table**) or in the existing table (for **Drop Table and Create New Table**). If you want to change the data type for the target, click the name and select a different one from the list.
 - If the **Column header row** option is *not* selected, **Data type** shows all available data types. Select the data type to use for the target column from the list.
- **Length/Precision** (Optional): For columns where the **Data Type** is **NUMBER**, enter the length/precision for the numbers in the column. Precision is the number of significant digits in a number. Precision can range from 1 to 38.

For columns where Data Type is VARCHAR2, the **Auto** value in **Length/Precision** field enables the Auto Size feature.

With the Auto-Size column Width feature, you can automatically size any column to fit the largest value in the column. Select **Auto** from the **Length/Precision** drop-down values or pick a value from the drop-down list.

- **Scale** (Optional): For columns where the **Data Type** is **NUMBER**, enter the scale for the numbers in the column. Scale is the number of digits to the right (positive) or left (negative) of the decimal point. Scale can range from ranges from -84 to 127.
- **Format:** If the data type in the **Data type** column is **DATE** or one of the **TIMESTAMP** types, select a format for that type from the **Format** drop-down list.
- **Merge Key:** This option is used only for the processing option **Merge into Table**.

For the **Merge into Table** option, you must specify at least one column to use as a key column. Merge keys are one or more columns that uniquely identify each row in the table. To specify a key column, select the **Merge Key** checkbox for the column. Merge keys must not contain any null values. For loading tables with primary keys, this option automatically enables the selection of primary key columns as the merge keys.

Preview Tab

The **Preview** tab displays the source data in tabular form in the **Load Preview** menu. The display reflects the settings you chose in the **Properties** section.

The **File** menu in the **Preview** tab displays the source data with all the columns with their names. You can copy the text.

Table Tab

The **Table** tab displays what the target table is expected to look like after the data has been loaded. If you chose the **Create Table** processing option, no table is shown.

SQL Tab

The **SQL** tab displays the SQL commands that will be run to complete this data load job.

Note

You can see the SQL code even before the table is created.

Close Button - Save and Close the Pane

After entering all the details for the data load job, click **Close** at the bottom of the page. This saves the details you entered and returns you to the Load Data from Cloud Storage pane. To close the page without saving your entries, press **Escape**.

Run the Data Load Job

Once you've added data sources to the data load cart and entered details about the data load job, you can run the job.

To run the job:

1. If you haven't already done so, click the **Close** button in the **Load Data from Cloud Storage** pane to save your settings and close the pane. If any of the settings are invalid, an error message reports the problem. Fix the problem and click **Close**.

2. Click  **Start** in the data load cart menu bar. To stop the data load job, click  **Stop**.

When you have added all of the sources for the job and specified the settings for each source, to run the job click the Start icon in the Data Load Cart menu bar. In the Run Data Load Job dialog box, click **Start**. To stop the data load job, click the Stop icon.

Once the data load job starts, you can view the progress of the job in the Data Load dashboard.

When the data load job completes, the Data Load dashboard page displays the results of the job. At the top of the header of the table load, you can view the name of the table along with the total columns present in the table.

Click **Job Report** to view the total number of rows processed successfully and the count of rejected rows. You can also view the Start time. The SQL pane of the Job Report displays the equivalent SQL code of the job.

To view information about an item in the job, click the **Actions** icon on the Table Load.

To view a log of the load operation, click the Logging icon. You can save the log, clear it, or refresh it. Click **OK** to dismiss the log.

View Details About the Data Load Job After It Is Run

To view details about the data load job after it is run, click the **Actions** icon on the **Table and View Loads** section of the Data Load dashboard.

You can view [Table details](#), [Gather Statistics](#), [Register to Cloud Link](#), [Create Analytic View](#), [Export Data to Cloud](#), [Edit Table](#) and delete the table.

When the data load job completes, the Data Load dashboard page displays the results of the job. At the top of the header of the table load, you can view the name of the table along with the total columns present in the table.

Click **Report** to view the total number of rows processed successfully and the count of rejected rows. You can also view the Start time. The SQL pane of the Report displays the equivalent SQL code of the job.

After your data load job, you might see errors that you want to correct, or upon inspection, realize that you wanted to name a column differently. In such cases, you will view a warning sign on the Table load. Click the **Reload** icon to reload source with suggested fixes. Click **Actions** icon on the Table load and select **Edit** to make any changes to the data load job (i.e., change a column name).

View the Table Resulting from the Data Load Job

To view the new tables or tables modified by the data load job, you can:

1. Fix your data load job. After your data load job, you might see errors that you want to correct, or upon inspection, realize that you wanted to name a column differently. In such cases, click the **Reload** option on the selected Table Load to reload cards from your recent cart and edit them as you did before your first attempt. The Reload icon reloads the source data with the fixes suggested by the tool. Click the **Actions** icon on the Table header, click **Table** and select **Edit** to make any changes to the data load job (i.e., change a column name).
2. You can use the **Edit** icon as mentioned in the above paragraph to review the table.

Linking Data

You can link to data in remote databases or in cloud storage buckets.

When you link to data in a remote database or in cloud storage, the target object produced is an external table or a view. When you select that target table or view on the Data Load Jobs page, the source preview for the object shows the current data in the source object.

Linking to columns in a remote database or to files in cloud storage can be useful when the source data is being continually updated. For example, if you have a database table with columns that are updated for each new sales transaction, and you have run a data load job that links columns of the source table to targets in your Oracle Autonomous Database, then those targets have the new sales data as it is updated in the source.

See:

- [Linking to Other Databases](#)
- [Linking to Objects in Cloud Storage](#)
- [Linking to Directory](#)
- [Linking to Share](#)
- [Linking to OCI Data Catalog](#)

- [Linking to Other Databases](#)

To link to data in tables in another database from your Oracle Autonomous Database, on the Data Load Dashboard, click **LINK DATA**, then click **Database**. Select a database link from the drop-down list. Drag one or more tables from the list of database tables and drop them in the Data Load Cart.

- [Linking to Objects in Cloud Storage](#)

When you create a link to files in a cloud store bucket from your Oracle Autonomous AI Database, you create an external table that links to the files in the cloud store.

- [Linking to File System](#)
You can link to file system directories from your Autonomous Database.
- [Linking to Share](#)
You can subscribe to a Share Provider for linking data.
- [Linking to Data Catalog](#)
You can link your Data Catalogs to Autonomous Database.

Linking to File System

You can link to file system directories from your Autonomous Database.

On the Data Load dashboard page, click **Link Data** and select **File System**. On the left side of the page is a navigator pane, where you choose a directory with the folders or files containing the data. On the right of the page is the data link cart, where you stage the files and folders for the data link job. You can set options for the data link job before running it. The Autonomous Database comes with predefined CPU/IO shares assigned to different consumer groups. You can set the consumer group to either low, medium or high while executing a data load job depending on your workload.

Drag and drop the files or folders from the directory you select to add them to linking job.

Enter Details for the Data Link Job

Enter the details about the data link job in the Link Data from Directory dialog box. For more details, refer to the [Linking to Objects in Cloud Storage](#) chapter.

Linking to Share

You can subscribe to a Share Provider for linking data.

On the Data Load Dashboard page, click **Link Data** and select **Share**. For more details, refer to the [Loading data from a Share](#) chapter.

Linking to Data Catalog

You can link your Data Catalogs to Autonomous Database.

Click **OCI Data Catalog** in the Link Data page under the Data Load menu to register a Data Catalog for linking data. Refer to the [Register Data Catalog](#) chapter for more details on this.

Linking to Other Databases

To link to data in tables in another database from your Oracle Autonomous Database, on the Data Load Dashboard, click **LINK DATA**, then click **Database**. Select a database link from the drop-down list. Drag one or more tables from the list of database tables and drop them in the Data Load Cart.

Each table appears as an item in the Data Load Cart. The item shows the name of the table and the number of rows in it, and the name of the table that is the target for the data load.

To remove a table from the Data Load Cart, select the Remove (trash can) icon for the item. To remove all tables from the cart, click the Remove All (trash can) icon in the Data Load Cart menu bar.

To add a remote database to the list of databases, create a database link to the remote database. For information on creating a database link, see Database Links in *Oracle® Database Database Administrator's Guide*.

The databases available to you appear in the drop-down list of the database navigation pane of the Link Tables page.

Specify the Target, Create Filters, and View Mappings

To specify settings for the data load job, preview the data in the source or the target, and see statistics about the data, click the Settings (pencil) icon for the item in the Data Load Cart.

In the settings pane, on the Settings tab, the **Source** field displays the name of the table and the number of rows in the table.

Preview Source Data

To view the data in the source table, in the settings pane select the **Source Preview** tab. The source preview displays the data in the table.

View Statistics

To view statistics about the source table, in the settings pane select the **Statistics** tab. It may take a moment for the statistics to appear. The statistics include the size of the table, the number of rows and columns, the column names, data types, number of distinct values, and other information. Below the details about the columns is a bar graph that displays the top unique values for the selected column.

Preview Target Data

To view the data in the target view, in the settings pane select the **Target Preview** tab. The target preview displays the data in the target view. If the view does not yet exist, then the target data is the same as the source data, regardless of any filters set for the data load job.

Run the Data Load Job

When you have added all of the source tables for the job and specified the settings for each table, to run the job click the Start icon in the Data Load Cart menu bar. In the Run Data Load Job dialog box, click **Start**. To stop the data load job, click the Stop icon.

At the top of the page, the **Status** shows the number of items for which the load has completed over the number of items in the job, and the total time elapsed for the job. When the data load job completes, the Link Tables page displays the results of the job.

To view information about an item in the job, click the Settings (circled i) icon in the item. The settings pane has the same **Settings**, **Preview**, and **Target Preview** tabs as the settings pane before running the job, except that the **Target Preview** now contains the data loaded by the data load job. To close the settings pane, click **Close**.

To view a log of the load operation, click the Logging icon. You can save the log, clear it, or refresh it. Click **OK** to dismiss the log.

The list of views on the Data Load page contains any new views created. A preexisting view that was the target for the data load job now contains the loaded data.

On the Database links page, click the **Reload Cart** button. Clicking the button enables you to add the data sources to the data load the cart again.

Click **Done** to return to the Database Actions page.

Linking to Objects in Cloud Storage

When you create a link to files in a cloud store bucket from your Oracle Autonomous AI Database, you create an external table that links to the files in the cloud store.

You can link to files in these file formats: AVRO, CSV, JSON, GeoJSON, Parquet, ORC, Delimited TXT. For information on supported file formats, see [Format Specifications for JSON, AVRO, and XML Files](#).

Configure and run a data link job from the Link Cloud Object page. To open that page:

1. Open the Database Actions launchpad, click **Data Studio** tab and select the **Data Load** menu. See, [The Data Load Page](#).
2. Select **LINK DATA** and **CLOUD STORE**.

On the left side of the page is a *navigator pane*, where you choose a cloud store connection and the folders or files containing the data. On the right of the page is the data load *cart*, where you stage the files and folders for the data link job. You can set options for the data link job before running it. The Autonomous Database comes with predefined CPU/IO shares assigned to different consumer groups. You can set the consumer group to either low, medium or high while executing a data load job depending on your workload.

To link to files from a cloud store, do the following:

- [Manage Cloud Storage Links for Data Link Jobs](#)
- [Prepare the Data Link Job](#)
- [Add Files or Folders for the Data Link Job](#)
- [Enter Details for the Data Link Job](#)
- [Run the Data Link Job](#)
- [View Details About the Data Link Job After It Is Run](#)
- [View the Table Resulting from the Data Link Job](#)

- [Manage Cloud Storage Links for Data Link Jobs](#)

Before you can link to data in a cloud store, you must establish a connection to the cloud store you want to use.

- [Prepare the Data Link Job](#)
- [Add Files or Folders for the Data Link Job](#)
- [Enter Details for the Data Link Job](#)

Enter the details about the data link job in the Link Data from Cloud Storage pane.

- [Run the Data Link Job](#)

Once you've added data sources to the data link cart and entered details about the data link job, you can run the job.

- [View Details About the Data Link Job After It Is Run](#)

You can view the progress of the job on the Data Load dashboard.

- [View the Table Resulting from the Data Link Job](#)

After running a data link job, you can view the table created by the data link job on the Data Load dashboard.

Manage Cloud Storage Links for Data Link Jobs

Before you can link to data in a cloud store, you must establish a connection to the cloud store you want to use.

On the Link Cloud Object page:

1. Click the Manage cloud store icon besides the field where you enter the cloud store location. Select **+ Create Cloud Store Location**.
2. Enter your information in the **Add Cloud Store Location** pane. See to add cloud storage location.

See [Managing Connections](#).

To return to the Link Cloud Object page, click **Data Load** in the breadcrumbs at the top of the page and then navigate back to the page.

Prepare the Data Link Job

You may have to adjust your source data or your target table so that the source data links correctly to the external target table. Consider:

- If you're linking to multiple files, you must make sure that:
 - All the source files are of the same type, for example, CSV, JSON, etc.
 - The number, order, and data types of the columns in all the source files match.
- If you want to partition by date:
 - The source file must contain data where the data type is date or timestamp.
 - You must load a folder containing two or more data sources.
 - The names of the files in the folder must indicate a date or dates, for example, `MAR-1999.csv` or `2017-04-21.xlsx`.

Add Files or Folders for the Data Link Job

Add files from the cloud store to the data link cart, where you can edit the details of the data link job. To add the files:

1. From the list at the top of the navigator pane on the left, select the bucket with your source data.

The list shows links that were established on the Manage Cloud Storage page. If you haven't yet registered the cloud store you want to use, click the **Connections** button under the Data Load menu in Data Studio suite of tools and register a connection.
2. Drag one or more items from the file navigator on the left and drop them into the cart on the right.
 - You can add files, folders, or both. A card is added to the cart for each file or folder you drag into it. The card lists the name of the source file or folder and a proposed name for the target table.
 - If you add a folder that contains multiple files, all the files must be of the same type, that is, CSV, TXT, etc.

When you add the folder to the cart, a prompt is displayed that asks if you want to load all the objects from the multiple source files into a single target table. Click **Yes** to continue or **No** to cancel.

- When you add multiple individual files or multiple folders to the cart, the data represented by each card will be loaded into a separate table, but all the items in the cart will be processed as part of the same data load job.
- You can add files or folders from a different bucket, but if you do that, you're prompted to remove all files that are already in the cart before proceeding. To select files from a different bucket, select the bucket from the drop-down list in the navigator pane on the left and then add the file(s), as described above.
- You can drop files or folders into the data load cart and then navigate away from the Data Link Object page. When you return to the page, those items remain on the page, but you may receive a message, "Remove All Data Link Items. Changing to another Cloud storage location requires all items to be removed from the data load job. Do you wish to continue?" Click **Yes** to remove the items from the cart. Click **No** to keep the items in the cart. Then you can continue to work.

You can remove items from the cart before running the data link job:

- To remove an item from the cart, select **Remove** on the card for the item Data Link cart menu bar at the top of the pane.
- To remove all items from the cart, click **Remove All** in the data link cart menu bar at the top of the pane.

Enter Details for the Data Link Job

Enter the details about the data link job in the Link Data from Cloud Storage pane.

On the card in the data link cart, click **Settings** to open the Link Data from Cloud Storage pane for that job. The pane contains:

- [Settings Tab - Table Section](#)
- [Settings Tab - Properties Section](#)
- [Settings Tab - Mapping Section](#)
- [File Tab](#)
- [Table Tab](#)
- [SQL Tab](#)
- [Close Button - Save and Close the Pane](#)

Settings Tab - Table Section

Set details about the target table in the **Table** section.

- **Name:** The name of the target table.
- **Partition Column:**

List Partitions and Date-based partitions are the different types of partitions available in data linking.

List partitioning is required when you specifically want to map rows to partitions based on discrete values.

To partition according to a specific column, click the **Partition Column** drop-down list and select the column you want to use for the partitioning.

You will have N files per partition value, all partitioned by the partition column you select.

Note

- For linked files (from external tables) there is also a requirement that for each file, the list partitioning column can contain only a single distinct value across all of the rows.
- If a file is list partitioned, the partitioning key can only consist of a single column of the table.

Date-based partitioning is available when you link a folder containing two or more data sources that have columns that contain date or timestamp data.

To partition according to date, click the **Partition Column** drop-down list and select the **DATE** or **TIMESTAMP** column you want to use for the partitioning.

- **Validation Type:** Validation examines the source files, optional partitioning information, and report rows that do not match the format options specified. Select **None** for no validation; select **Sample** to perform validation based on a sample of the data; or select **Full** to perform validation based on all the data.
- **Use Wildcard:** This check box enables use of wildcard characters in search condition to retrieve specific group of files that matches the filter criteria. You can use a wildcard character, such as an asterisk (*) that searches, filters, and specifies groups of files that detect and add new files to the external table.

For example, if you enter file*, then file01, file02, file03, and so on are considered to match the keyword. The asterisk (*) matches zero or more characters of the possibilities, to the keyword.

Note

The wildcard support is incompatible with partitioning. The validation of source file fails if you use wildcards with partitioned data.

Settings Tab - Properties Section

Specify options to control how the source data is interpreted, previewed, and processed. These options vary, depending on the type of source data.

- **Encoding:** Select a character encoding type from the list. This option is available when the linked file is in plain text format (CSV, TSV, or TXT). The default encoding type is UTF-8.
- **Text enclosure:** Select the character for enclosing text: " (double-quote character), ' (single-quote character) or **None**. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Field delimiter:** Select the delimiter character used to separate columns in the source. For example, if the source file uses semicolons to delimit the columns, select **Semicolon** from this list. The default is **Comma**. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Start processing data at row:** Specifies the number of rows to skip when linking the source data to the target external table:

- If you select the **Column header row** option under **Source column name** (see below) and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows after the first row are not linked to the target.
- If you deselect the **Column header row** option under **Source column name**, and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows including the first row are not linked to the target.
- **Source column name:** Select the **Column header row** checkbox to use the column names from the source table in the target table.
 - If you select this option, the first row in the file is processed as column names. The rows in the **Mapping** section, below, are filled with those names (and with the existing data types, unless you change them).
 - If you deselect this option, the first row is processed as data. To specify column names manually, enter a name for each target column in the **Mapping** section. (You will also have to enter data types.)
- **Numeric column:** Select the **Convert invalid data to null** checkbox to convert an invalid numeric column value into a null value.
- **Newlines included in data values:** Select this option if there are newline characters or returns to the beginning of the current line without advancing downward in the data fields. Selecting this option will increase the time taken to process the load. If you do not select this option when loading the data, the rows with newlines in the fields will be rejected. You can view the rejected row in the Job Report panel.

Settings Tab - Mapping Section

The settings in the **Mapping** section control how data from the source files are linked to the rows of the target external table. For each row, the data from the column listed under **Source column** will be linked to the column listed under **Target column**.

- **Source column:** Lists the columns from the source file.

If the **Column header row** option under **Properties** is selected, **Source column** shows the names of the columns in the source file. If the **Column header row** option is not selected, generic names like **COLUMN_1**, **COLUMN_2**, etc., are used. This field is always read only.

You can view two source columns `FILE$NAME` and `SYSTIMESTAMP`. The `FILE$NAME` column enables you to locate the source file containing a particular data record. For example, you load a source file that contains a list of files. The file names in the file list refer to the department names across the organization. For instance, a *finance.txt* file contains data from the Finance department. In the mapping, you can use string data types to extract the department name from the output of the file name column. You can use the extracted department name to process data differently for each department.

The `SYSTIMESTAMP` column allows us to view the current timestamp in the database.

Note

- `FILE$NAME` and `SYSTIMESTAMP` source columns are not included by default. You must check the **Include** check box and run the load for the target table to display these two columns.
- When you are creating a livefeed, the `FILE$NAME` and `SYSTIMESTAMP` source columns appear in the Mapping table by default.

- **Target column:** Lists the columns in the target table.
 - If the **Column header row** option *is* selected, then the **Target column** uses the names of the columns in the source file. You can change the name of a target column by replacing the provided name with a new one. You need to make sure that the target column is not empty. Target column name must not be a duplicate of another target column. The target column name cannot have duplicate name as another target column. The target column length must not be beyond 128 bytes. 128 byte limit is a database limit.
 - If the **Column header row** option is *not* selected, then generic names like **COLUMN_1**, **COLUMN_2**, etc., are used. You can change the name of a target column by replacing the provided name with a new one.

Note

If you're linking multiple files from a folder in a single data link job, only the first file will be shown in the **Mapping** section. However, as long as the column names and data types match, the data from all source files will be linked.

- **Data Type:** Lists the data type to use for data in that column. The contents change depending on whether the **Get from file header** option is selected.
 - If the **Column header row** option *is* selected, **Data type** shows the data types of the columns in the source file. If you want to change the data type for the target, click the name and select a different one from the list.
 - If the **Column header row** option is *not* selected, **Data type** shows all available data types. Select the data type to use for the target column from the list.
- **Length/Precision (Optional):** For columns where the **Data Type** is **NUMBER**, enter the length/precision for the numbers in the column. Precision is the number of significant digits in a number. Precision can range from 1 to 38.
For columns where Data Type is VARCHAR2, the **Auto** value in **Length/Precision** field enables the Auto Size feature.
With the Auto-Size column Width feature, you can automatically size any column to fit the largest value in the column. Select **Auto** from the **Length/Precision** drop-down values or pick a value from the drop-down list.
- **Scale (Optional):** For columns where the **Data Type** is **NUMBER**, enter the scale for the numbers in the column. Scale is the number of digits to the right (positive) or left (negative) of the decimal point. Scale can range from ranges from -84 to 127.
- **Format:** If the data type in the **Data type** column is **DATE** or one of the **TIMESTAMP** types, select a format for that type from the from the **Format** drop-down list.

Preview Tab

The **Load Preview** menu in the **Preview** tab displays the source data in tabular form. The display reflects the settings you chose in the **Properties** section. The **File** menu displays source data with the column names.

If you dragged a folder containing multiple files into the data link cart and then clicked  **Settings** for that card, the **Preview** pane includes a **Preview Object (File)** drop-down list at the top of the pane that lists all the files in the folder. Select the source file you want to preview from that list.

Table Tab

The **Table** tab displays what the target table is expected to look like after the data has been linked.

SQL Tab

The **SQL** tab displays the SQL commands that will be run to complete this data link job.

Note

You can see the SQL code even before the table is created.

Close Button - Save and Close the Pane

After entering all the details for the data link job, click **Close** at the bottom of the page. This saves the details you entered and returns you to the Link Data from Cloud Storage pane.

Run the Data Link Job

Once you've added data sources to the data link cart and entered details about the data link job, you can run the job.

To run the job:

1. If you haven't already done so, click the **Close** button in the **Link Data from Cloud Storage** pane to save your settings and close the pane. If any of the settings are invalid, an error message reports the problem. Fix the problem and click **Close**.

2. Click  **Start** in the data link cart menu bar. To stop the data link job, click  **Stop**.

When the data link job completes, the **Data Load Dashboard** page displays the results of the job under **Table and View Loads** section.

Once the data link job starts, you can view the progress of the job in the Data Load dashboard.

View Details About the Data Link Job After It Is Run

You can view the progress of the job on the Data Load dashboard.

When the data load job completes, the Data Load dashboard page displays the results of the job. At the top of the header of the table load, you can view the name of the table along with the total columns present in the table.

Click **Job Report** to view the total number of rows processed successfully and the count of rejected rows. You can also view the Start time. The SQL pane of the Job Report displays the equivalent SQL code of the job.

To view information about an item in the job, click the **Actions** icon on the Table Load.

To view a log of the load operation, click the Logging icon. You can save the log, clear it, or refresh it. Click **OK** to dismiss the log.

View the Table Resulting from the Data Link Job

After running a data link job, you can view the table created by the data link job on the Data Load dashboard.

Fix your data load job. After your data load job, you might see errors that you want to correct, or upon inspection, realize that you wanted to name a column differently. In such cases, click the **Reload** option on the selected Table Load to reload cards from your recent cart and edit them as you did before your first attempt. The Reload icon reloads the source data with the fixes suggested by the tool. Click the **Actions** icon on the Table header, click **Table** and select **Edit** to make any changes to the data load job (i.e., change a column name).

Feeding Data

You can run a live table feed on demand, on a schedule, or as the result of a notification.

A Live Table Feed automates loading of data into a table in your database. Files automatically load as they appear in object storage and the Live Table Feed system ensures that files are only loaded once. The loading can happen manually, by a schedule, or even by notifications delivered directly from Object Storage.

The bucket can contain files in these formats: AVRO, CSV, JSON, GeoJSON, Parquet, ORC, Delimited TXT. All of the files must have the same column signature.

About the Live Feed Page

On the Database Actions - Data Load Dashboard page select **FEED DATA** to display the Live Feed page. On this page, you can:

- [Manage Cloud Storage Connections for Live Table Feeds](#)
- [Create a Live Table Feed Object](#)
- [List, Filter, and Sort Live Table Feed Objects](#)
- [Find and View Live Table Feed Objects](#)
- [Run a Live Table Feed](#)
- [Delete a Live Table Feed](#)

Manage Cloud Storage Connections for Live Table Feeds

Before you create a live table feed, you must establish a connection to the cloud store you want to use:

1. Click **Connections** under the Data Load menu. For instructions, see [Managing Connections](#).

Create a Live Table Feed Object

To create a live table feed object,

1. On the Live Feed page, click the **+ Create Live Table Feed** button to display the Live Feed Settings pane. Enter information on the Data Source tab as follows:
 - **Cloud Store Location:** Select the Cloud Store Location from the drop-down. Select the cloud connection for the bucket containing the file you want to use for feeding data.

On the **Basic** mode, you can view the following options:

- **Folders:**
Select the folder containing the file(s) you want to use for feeding data on the object store. Select **Entire Bucket** to load all the files in your bucket into your table. The folders are listed and organized in the drop-down based on how you create folders or directories and store your files. For example, you can create a sales folder to store *sales1.csv* and *sales2.csv* files.
- **Extensions:**
Enter an extension to limit the live table feed to only those files in the bucket that match the extension. For example, to limit the files to only those that are CSV files, select **CSV**.

On the **Advanced** mode, you can view the following options:
 - **Object Filter (glob):** Enter a file glob to limit the live table feed to only those files in the bucket that match the glob. For example, to limit the files to only those that are CSV files, enter ***.CSV**.
 - Under the Live Feed File Preview section, you can view a preview of the file you select in the previous step.

Click **Next** to progress to the **Table Settings** tab.

On the **Option** field, select any of the two available options:

- **Load Table:** This option appends the feed into the target table. It adds new rows to your table for every file it views.
- **Merge into Table:** This option merges live feed data into the target table. Click **Merge Key** in the Mapping section when the row already exists, and the loaded key matches the newly added feed. This avoids insertion of a new row.
- **Load Collection:** You can define your livefeed over JSON files with this option. By selecting this option, you will not view the mapping section and Add Expression section. The tool loads the source data into a JSON collection which you can extract later.

Target Table Name: Accept the default name or enter a different name. This is the name of the target table that data from the live feed will be loaded into in your Autonomous Database instance. If the table does not exist, Live Feed will attempt to guess the correct columns. You can pre-create the table in which you wish the Live Feed to load into. This is for higher accuracy.

The Table Settings tab specifies options to control how the source data is interpreted, previewed, and processed. These options vary, depending on the type of source data.

- **Encoding:** Select a character encoding type from the list. This option is available when the linked file is in plain text format (CSV, TSV, or TXT). The default encoding type is UTF-8.
- **Text enclosure:** Select the character for enclosing text: " (double-quote character), ' (single-quote character), or None. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Field delimiter:** Select the delimiter character used to separate columns in the source. For example, if the source file uses semicolons to delimit the columns, select **Semicolon** from this list. The default is **Comma**. This option is visible only when the selected file is in plain text format (CSV, TSV, or TXT).
- **Start processing data at row:** Specifies the number of rows to skip when linking the source data to the target external table:
If you select the **Column header row** option under **Source column name** (see below) and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows after the first row are not linked to the target.

If you deselect the **Column header row** option under **Source column name**, and if you enter a number greater than 0 in the **Start processing data at row** field, then that number of rows including the first row are not linked to the target.

Column header row: Select the **Column header row** checkbox to use the column names from the source table in the target table.

If you select this option, the first row in the file is processed as column names. The rows in the **Mapping** section, below, are filled with those names (and with the existing data types, unless you change them).

If you deselect this option, the first row is processed as data. To specify column names manually, enter a name for each target column in the **Mapping** section. (You will also have to enter data types).

Select the **Convert invalid data to null** checkbox to convert an invalid numeric column value into a null value.

Newlines included in data values: Select this option if there are newline characters or returns to the beginning of the current line without advancing downward in the data fields. Selecting this option will increase the time taken to process the load. If you do not select this option when loading the data, the rows with newlines in the fields will be rejected. You can view the rejected row in the Job Report panel.

- Edit or update the table settings in the **Mapping** section: In this pane, the mapping of the source to target columns is displayed. The contents of the **Mapping** table change according to what processing option you chose in the **Table** section and which properties you set in the **Properties** section. You can filter the results in the mapping table with **Quick Filter** field that enables you to filter out **Columns** or **Expressions**.

Select **Add Expression** to perform **Sentiment Analysis** or **Key Phrase extraction** or **Language Detection** or **Text Translation** with the source data. See [Use OCI Language Service Capabilities in Data Studio](#) for more details.

- Select the Include check box at the beginning of a row to add the column to the target table.
- Select or enter values for column attributes such as Target Column Name, Column Type, Precision, Scale, Default, Primary Key and Nullable.
- You need to review the suggested data type and if needed, modify it by entering the data type directly into the target cell. Review the generated mapping table code based on the selections made in the previous screens.

Click **Merge Key** in the Mapping section when the row already exists, and the loaded key matches the newly added feed. This avoids insertion of a new row. This option shows up when you select the **Merge into Table** option.

Click **Next** to progress to the Preview tab.

- The Preview Pane displays the changes you make to the table.
- Click **Next** to progress to the Live Feed Settings tab.

On the Live Feed Settings tab, specify the following field values:

- **Live Table Feed Name:** Accept the default name or enter a different name to identify this live table feed.
- **Enable for Notification:** Select this option so that new or changed data in the data source will be loaded based on an Oracle Cloud Infrastructure *notification*. When you

select this option, you can avoid delays that might occur when polling is initiated on a schedule (that is, if you selected the live table feed **Scheduled** option).

When you select the **Enable for Notification** option, you must also:

- Configure your object store bucket to emit notifications
- Create a Notifications service subscription topic
- Create an Events service rule
- Copy the notification URL
- Create a Notifications service subscription
- Confirm that notifications are allowed

For complete instructions, see [Creating a Notification-Based Live Table Feed](#).

- **Enable for Scheduling:** Select this option to set up a schedule for running the live table feed object; that is, to poll the data source on a regular basis:
 - In the time interval fields, enter a number, and select a time type and the days on which to poll the bucket for new or changed files. For example, to poll every two hours on Monday, Wednesday, and Friday, enter 2, select **Hours**. You can select **All Days**, **Monday to Friday**, **Sunday to Thursday**, or **Custom** from **Week Days** drop-down. The Custom field enables you to select **Monday**, **Tuesday**, **Wednesday**, **Thursday** and **Friday** in the appropriate fields.
 - Select a start and end date with start and end time. If you don't select a start date, the current time and date are used as the start date. The end date is optional. However, without an end date, the live feed will continue to poll.

Select a consumer group from the drop-down, namely, low, medium and high.

2. Click **Create** to create the Live Table feed object.

Show code: Select this option to view the PL/SQL code equivalent of the Create Live Table Feed wizard. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Live Table Feed wizard.

List, Filter, and Sort Live Table Feed Objects

When you open the Live Feed page, existing live table feed objects are displayed as cards on the page. They are identified as LIVE_TABLE_FEED entities.

To filter live table feed objects:

1. Click the search field at the top of the page to display filter options. By default, the live table feed objects from the current user's schema are shown. As soon as you start typing in the search field, the feed tool returns the values which contain the letters you type. You can remove the filter by deleting the content from search box and clicking the cross icon that appears next to the search box.
2. To include objects from other schemas, select the drop-down next to the search field, under **Schema**. To remove a schema from the filter list, deselect the box next to its name.
3. To show objects from all available schemas, select **All** from the **Schema** drop-down.

To sort live table feed objects

1. Click the **Sort by** button at the top right of the page.
2. Select a sorting option. To sort ascending, click the icon with the up arrow. To sort descending, click the icon with the down arrow.

Find and View Live Table Feed Objects

To search for available live table feed entities in the selected schemas, enter a value in the search field at the top of the page and press **Enter**. The display then includes only the entities whose names contain the characters in the search field. To clear the search field, click the Clear search results (X) icon in the search field.

To remove a schema or sorting value from the selected filters, deselect the schema or sorting value in the filter panel, or click the Remove filter (X) icon for the schema or sorting value above the display of live table feed objects. To close the filter panel, click the Hide filter panel (X) icon in the panel.

To refresh the display of live table feeds, click the Refresh icon at the top of the page.

Edit a Live Table Feed Object

To edit details of a live table feed object,

1. On the Live Feed page, find the card for the live table feed whose details you want to edit.
2. Click the Actions icon (three dots) on the card and select **Edit Live Table Feed**. You can edit the following options:
 - Enter information on the Data Source tab as follows:
 - **Cloud Store Location:** Select the Cloud Store Location from the drop-down. Select the cloud connection for the bucket containing the file you want to use for feeding data.
 - On the **Basic** mode, you can view the following options:
 - **Folders:**
Select the folder containing the file you want to use for feeding data on the object store. Select **Entire Bucket** to upload all the files to your bucket. The folders are listed and organized in the drop-down based on how you create folders or directories and store your files. For example, you can create a sales folder to store *sales1.csv* and *sales2.csv* files.
 - **Extensions:**
Enter an extension to limit the live table feed to only those files in the bucket that match the extension. For example, to limit the files to only those that are CSV files, select **CSV**.

On the **Advanced** mode, you can view the following options:
 - **Object Filter (glob):** Enter a file glob to limit the live table feed to only those files in the bucket that match the glob. For example, to limit the files to only those that are CSV files, enter ***.CSV**.
 - On the **Live Feed Settings** tab, edit the following fields:
 - **Enable for Notification:** Select this option so that new or changed data in the data source will be loaded based on an Oracle Cloud Infrastructure *notification*. When you select this option, you can prevent any delays that might occur when polling is initiated on a schedule (that is, the live table feed **Scheduled** option).

When you select the **Enable for Notification** option, you must also:

- Copy the live table feeds notification URL
- Configure your cloud store to emit notifications
- Configure Oracle Cloud Infrastructure to route events to the endpoint used for the live table feed.

- Create a rule.
- Create a subscription.
- Confirm that notifications are allowed at the live feed service.

For complete instructions, see [Creating a Notification-Based Live Table Feed](#).

- **Scheduled:** Select this option to set up a schedule for running the live table feed object; that is, to poll the data source on a regular bases:
 - In the time interval fields, enter a number, and select a time type and the days on which to poll the bucket for new or changed files. For example, to poll every two hours on Monday, Wednesday, and Friday, enter 2, select **Hours**. You can select **All Days**, **Monday to Friday**, **Sunday to Thursday**, or **Custom** from **Week Days** drop-down. The Custom field enables you to select **Monday**, **Tuesday**, **Wednesday**, **Thursday** and **Friday** in the appropriate fields.
 - Select a start and end date with start and end time.

3. Click **Save**.

Run a Live Table Feed

You can run a live table feed on demand, on a schedule, or as the result of a notification.

To run a live table feed on demand:

1. On the Live Feed page, find the card for the live table feed you want to run.
2. Click the Actions icon (three dots) on the card and select **Run Live Table Feed Immediately (Once)**.

To run a live table feed on a schedule:

You can set a schedule for running live table feeds on the **Create Live Table Feed** pane (when creating a new table feed) or the **Edit Live Table Feed** pane (when editing an existing table feed). See [Create a Live Table Feed Object](#) or [Edit a Live Table Feed Object](#).

To run a live table feed as the result of a notification:

See [Creating a Notification-Based Live Table Feed](#).

Select the **Scheduled** check box to display the schedule options and then set the schedule by selecting the options you want.

To view live table feed run details:

1. On the Live Feed page, find the card for the live table feed whose run details you want to see.
2. Click the Actions icon (three dots) on the card and select **Live Table Feed Run Details**. The **Objects** tab on the Live Table Feed Run Details pane displays information about the jobs, such as when the run occurred, the objects involved in the run, the table owner, the table Name, status of the live feed, the rows loaded and the rows rejected, and other details. Click the **All** tab to view more details, such as the event type.

Delete a Live Table Feed

1. On the Live Feed page, find the card for the live table feed job you want to delete.
2. Click the Actions icon (three dots) on the card and select **Delete Live Table Feed**.

- [Creating a Notification-Based Live Table Feed](#)
You can load data through a live table feed based on an Oracle Cloud Infrastructure *notification*.
- [Creating a Notification-Based Live Table Feed using Amazon Simple Storage Service \(S3\)](#)
You can integrate Amazon Simple Storage Service (S3) and Oracle Cloud Infrastructure (OCI) to automate the process of live feed notifications when storage objects it is observing have updates. The following section provides instructions for creating event notifications in your Amazon S3 bucket where your data files are stored.
- [Creating a Notification-Based Live Table Feed using Microsoft Azure](#)
A notification-based Live Table Feed is an interface between Oracle Cloud Infrastructure and a third-party cloud message queuing service such as Azure Event Grid.
- [Send Email using Live Feed Tool](#)
You can configure the Live Feed tool to send you email notifications when certain specific live feed events occur.

Creating a Notification-Based Live Table Feed

You can load data through a live table feed based on an Oracle Cloud Infrastructure *notification*.

In addition to being able to run a live table feed on demand or on a schedule, as described in [Feeding Data](#), you can also run a feed as the result of a notification. When data in the source bucket is changed, a notification is sent which triggers a run of the table feed. With a notification-based live table feed, you can avoid any delay that might come from running on-demand or scheduled live table feed jobs.

Note

- Notification-based live table feeds aren't available on the Oracle Cloud Infrastructure free tier. You must be on a paid tenancy with appropriate permissions on your account to use this feature.
- Notification-based live table feeds aren't available on Oracle Autonomous Data Warehouse (ALK) databases that are configured using a private endpoint.

To create a notification-based live table feed:

- [Step 1: Configure your object store bucket to emit notifications](#)
- [Step 2: Create a Notifications service subscription topic](#)
- [Step 3: Create an Events service rule](#)
- [Step 4: Create and configure a live table feed to use notifications, and copy the notification URL](#)
- [Step 5: Create a Notifications service subscription](#)
- [Step 6: Confirm that the endpoint can receive notifications](#)

 **Tip**

To complete those steps, you will alternate between Oracle Cloud Infrastructure Console pages and Oracle Database Actions pages. You may find it convenient to open the Cloud Console in one browser page or tab and Database Actions in another, so it's easy to move back and forth.

Step 1: Configure your object store bucket to emit notifications

Where: Oracle Cloud Infrastructure Console: Object Storage & Archive Storage - Buckets page

Configure the bucket containing your source data so that it will emit notifications when the data changes. You can set this option when you create a bucket or you can set it in an existing bucket.

1. Open the Cloud Console navigation menu and click **Storage**. Under **Object Storage and Archive Storage**, click **Buckets**.
2. **If you're creating a new bucket:**
 - a. On the Buckets page, click the **Create Bucket** button to create a new bucket, as described in [Managing Buckets](#). In the **Create Bucket** wizard, select the **Emit Object Events** option, along with the other options for your new bucket.
 - b. Click **Create**.

If you're using an existing bucket:

- a. On the Buckets page, click the name of the bucket you want to use, as described in [Managing Buckets](#).
- b. On the Bucket Details page, click the **Edit** link next to **Emit Object Events**.
- c. Select the **Emit Objects Events** check box, and then click **Save Changes**.

Step 2: Create a Notifications service subscription topic

Where: Oracle Cloud Infrastructure Console: Notifications - Topics page

1. Open the Cloud Console navigation menu and click **Developer Services**. Under **Application Integration**, click **Notifications**.
2. Click **Create Topic**, enter a name and optional description, and then click **Create**.

Step 3: Create an Events service rule

Where: Oracle Cloud Infrastructure Console: Events - Rules page

1. Open the Cloud Console navigation menu and click **Observability & Management**. Under **Events Service**, click **Rules**.
2. Click **Create Rule**, and fill out the Create Rule page as described in [Managing Rules for Events](#).
 - Under **Rule Conditions**, select:
 - **Condition:** Event Type
 - **Service Name:** Object Storage
 - **Event Type:** Object - Create
 - Under **Actions**, select:

- **Action Type:** Notifications
- **Notifications Compartment:** Select the compartment to use for the notifications.
- **Topic:** Select the name of the topic you created above, in [Step 2: Create a Notifications service subscription topic](#).

3. Click **Create Rule**.

Step 4: Create and configure a live table feed to use notifications, and copy the notification URL

Where: Database Actions: Live Feeds page

You can configure a new or an existing live table feed to use notifications:

1. Go to the Database Actions Live Feeds page, as described in [Feeding Data](#).
2. Create or edit a live table feed object, as described in [Create a Live Table Feed Object](#) or [Edit a Live Table Feed Object](#). Select the **Enable for Notification** option
3. Click **Create** or **Save**.
4. Click the Actions (three vertical dots) icon on the card for your live feed, and select **Show Confirmation URL**.
5. In the **Notification URL** dialog box, click the **Copy** icon to copy the URL to the clipboard. You may want to copy it to a temporary file, so you can retrieve it later. You'll use this URL in the next step, [Step 5: Create a Notifications service subscription](#).

Step 5: Create a Notifications service subscription

Where: Oracle Cloud Infrastructure Console: Notifications - Subscriptions page

1. Return to the Oracle Cloud Infrastructure Console. Open the navigation menu and click **Developer Services**. Under **Application Integration**, click **Notifications**.
2. On the Notifications page, click the **Subscriptions** tab (on the left side of the page), the status will be **Active**.
3. Click **Create Subscription** and fill in the **Create Subscription** page:
 - **Subscription topic:** Select the subscription topic you created in [Step 2: Create a Notifications service subscription topic](#).
 - **Protocol:** HTTPS (Custom URL)
 - **URL:** Paste in the URL you copied in [Step 4: Create and configure a live table feed to use notifications, and copy the notification URL](#).
 - Click **Create**. The subscription will be listed in the **Subscriptions** table in a state of "Pending."

Step 6: Confirm that the endpoint can receive notifications

Where: Database Actions: Live Feeds page

1. Return to the Database Actions Live Feeds page and find the card for the live table feed you are configuring for a notification-based feed.
2. You will view the live feed card in an Active status.

You will receive email notifications when certain specific live feed events occur.

See [Send Email using Live Feed Tool](#) for more details.

Once you finish the above steps, any new files uploaded to the bucket will automatically be loaded into the live table feed table.

Creating a Notification-Based Live Table Feed using Amazon Simple Storage Service (S3)

You can integrate Amazon Simple Storage Service (S3) and Oracle Cloud Infrastructure (OCI) to automate the process of live feed notifications when storage objects it is observing have updates. The following section provides instructions for creating event notifications in your Amazon S3 bucket where your data files are stored.

✓ Tip

To complete these steps, you will need to alternate between Amazon Web Services (AWS) Management console and Oracle Database Actions pages. You may find it convenient to open the Amazon Web Services in one browser page or tab and Database Actions in another, so it is easy to move back and forth.

To create a notification-based live feed with Amazon S3 as cloud storage you must:

- [Step 1: Create your object store bucket in Amazon S3](#)
- [Step 2: Create Access Keys](#)
- [Step 3: Add an OCI Cloud Storage using Amazon S3](#)
- [Step 4: Create and configure a live table feed to use notifications and copy the notification URL](#)
- [Step 5: Create a notifications service subscription topic](#)
- [Step 6: Enable and configure event notifications using the Amazon S3 console](#)
- [Step 7: Create a notifications service subscription](#)
- [Step 8: Confirm that the endpoint can receive notifications](#)

Step 1: Create your object store bucket in Amazon S3

Where: Amazon Web Services (AWS) Management console

Configure and create your bucket containing source data so that it emits notifications when the data changes.

1. Log in to AWS Management console and open the Amazon S3 console.
2. On the home page click the **Create Bucket** icon.
3. In **Bucket name**, enter a valid name for your bucket. For example: testbucket. After you create the bucket, you cannot change its name.
4. In **Region**, select the Amazon Web Services (AWS) Region from the dropdown. For example: us-west-2
5. In Bucket settings for Block Public Access, select the **Block Public Access** settings that you want to apply to the bucket. It is recommended to keep all settings enabled unless you know that you need to turn any of them off.
6. Select **Advanced settings**, and accept all the default options if you want to enable S3 Object Lock. This step is optional.

7. Select **Create bucket**.

Step 2: Create Access Keys

Where: AWS Management console

To access Amazon Simple Notification Service (SNS), you must have credentials that Amazon Web Services (AWS) can use to validate your requests. These credentials must have permissions to access Amazon SNS topics. The following steps provide you details on steps to create access keys using AWS Identity and Access Management (IAM) for security purposes.

1. Log in to AWS Management console and open Amazon Identity and Access Management (IAM) console.
2. On the navigation menu, select **Users**.
3. Select your **user name**.
4. In the Security Credentials tab, select **Create access key**.
5. Copy the **Access key ID** and **Secret access key** in the display. Paste them in a clipboard.
6. To download the keys, select **Download.csv file** icon. This way you can store the file in a secure location.

Step 3: Add an Amazon S3 Cloud Storage Link

Where: Database Actions: Manage Cloud page

Before you create a live table feed, you must establish a connection to the cloud store you want to use.

1. Click the **Manage Cloud Store** button at the top of the page to go to the Manage Cloud page. For further instructions on adding source files residing in cloud storage provided by Amazon S3, refer to *Create an Amazon S3 Cloud Storage Link* topic in [Managing Connections](#).

① Note

Paste the Access key ID and Secret access key generated in the previous step ([Step 2: Create Access Keys](#)) to their respective text fields in the **Add Cloud Storage** page.

Step 4: Create and configure a live table feed to use notifications, and copy the notification URL

Where: Database Actions: Live Feeds page

Creating a live table feed enables you to load data in real time from external storage sources to your table in ADB. External storage you use include as Oracle Object Store, AWS S3 or Microsoft Azure containers.

You can configure a new or an existing live table feed to use notifications:

1. Go to the Database Actions Live Feeds page, as described in [Feeding Data](#).
2. Create or edit a live table feed object, as described in [Create a Live Table Feed Object](#) or [Edit a Live Table Feed Object](#). Select the **Enable for Notification** option
3. Click **Create** or **Save**.

4. Click the **Actions** (three vertical dots) icon on the card for your live feed, and select **Show Notification URL**.
5. In the **Notification URL** dialog box, click the **Copy** icon to copy the URL to the clipboard. You may want to copy it to a temporary file, so you can retrieve it later. You will use this URL in the subsequent step ([Step 7: Create a notifications service subscription](#)).

Step 5: Create a notifications service subscription topic

Where: Amazon Simple Notification Service (SNS) console

You receive Amazon S3 notifications using Amazon Simple Notification Service (Amazon SNS) topic. You need to add a notification configuration to your bucket using an Amazon SNS topic. SNS topics are shared locations which are used to send notifications of various events that happen in AWS buckets.

During creation, you select a topic name and topic type. After creating a topic, you cannot change the topic type or name. All other configuration choices are optional during topic creation, which you can edit later.

To access any AWS service, you must first create an AWS account.

Navigate to the AWS Management console, and then select **Create an AWS Account**.

Follow the instructions as provided in the [Amazon SNS link](#) to create your first IAM administrator user and group. Now you can log in to any of the AWS services as an IAM user.

1. Log in to Amazon SNS console as an IAM user.
2. On the **Topics** page, select **Create topic**.
3. Specify the following fields on the **Create topic** page, in the **Details** section.
 - **Type:**Standard (Standard or FIFO)
 - **Name:** notify-topic. For a FIFO topic, add fifo to the end of the name.
 - **Display Name:** This field is optional.
4. Expand the **Encryption** section and select **Disable encryption**.
5. Expand the **Access policy** section and configure additional access permissions, if required. By default, only the topic owner can publish or subscribe to the topic. This step is optional. Edit the JSON format of the policy based on the topic details you enter. Here is a sample of Access policy in JSON format.

```
{ "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    { "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": { "AWS": "*"
    }, "Action": [
        "SNS:Publish",
        "SNS:RemovePermission",
        "SNS:SetTopicAttributes",
        "SNS>DeleteTopic",
        "SNS:ListSubscriptionsByTopic",
        "SNS:GetTopicAttributes",
        "SNS:AddPermission",
        "SNS:Subscribe"
      ]
    }
  ],
```

```

        "Resource": "arn:aws:sns:us-west-2:555555555555:notify-topic", //us-
west-2 is the region
        "Condition": {
            "StringEquals": {
                "AWS:SourceOwner": "555555555555"
            }
        },
        {
            "Sid": "s3_policy", //This field accepts string values
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "arn:aws:sns:us-west-2:555555555555:notify-topic", //
notify-topic is the topic name
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "555555555555" //This is the Account ID
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:testbucket /*testbucket is the
s3 bucket name. You will get notifications only when file is uploaded to
this
                    bucket.*/
                }
            }
        }
    ]
}

```

6. Expand the **Delivery retry policy (HTTP/S)** section to configure how Amazon SNS retries failed message delivery attempts. This step is optional.
7. Expand the **Delivery status logging** section to configure how Amazon SNS logs the delivery of messages to CloudWatch. This step is optional.
8. Expand **Tags** section to add metadata tags to the topic. This step is optional.
9. Select **Create topic**.
10. The topic's Name, ARN (Amazon Resource Name), and Topic owner's AWS account ID are displayed in the Details section.
11. Copy the topic ARN to the clipboard.

Step 6: Enable and configure event notifications using the Amazon S3 console

Where: Amazon S3 Management console

You can enable Amazon S3 bucket events to send a notification message to a destination whenever those events occur. You configure event notifications for your S3 bucket to notify OCI when there is an update or new data available to load. The following steps explain the procedure to be followed in Amazon S3 console to enable event notifications.

1. Log in to Amazon S3 Management console and sign in as an IAM (Amazon Identity and Access Management) user.
2. In the **Buckets** list, select the name of the bucket i.e. testbucket. This is the bucket that you had created in [Step 1: Create your object store bucket in Amazon S3](#).
3. Select **Properties** icon.
4. Navigate to the **Event Notifications** section and select **Create event notification** icon.
5. In the **General configuration** section, specify the following values for event notification.
 - **Event name:** bucket-notification
 - **Prefix:** This value is to filter event notifications by prefix. It is an optional value. This is added to filter event activity.
 - **Suffix:** This value is to filter event notifications by suffix. It is an optional value. This is added to filter event activity.
6. In the **Event types** section, select one or more event types that you want to receive notifications for. If you are unsure of what event types to pick, then select the **All object create events** option.
7. In the **Destination** section, select **SNS Topic** as the event notification destination.

Note

Before you can publish event notifications, you must grant the Amazon S3 the necessary permissions to call the relevant API. This is so that it can publish notifications to a Lambda function or an SNS topic.

8. After you select SNS topic as the event notification destination, select the SNS topic i.e. notify-topic from dropdown. This is the topic you created in [Step 5: Create a notifications service subscription topic](#).
9. Select **Save changes**.

Step 7: Create a notifications service subscription

Where: Amazon SNS console

Every Amazon SNS topic has a set of subscriptions. Once a message is published to a topic, SNS handles distributing the message to all its subscribers. The subscribers can be AWS Lambda functions, HTTP(S) endpoints, email addresses and mobile phone numbers capable of receiving SMS messages.

Amazon SNS matches the topic to a list of subscribers who have subscribed to that topic and delivers the message to each of those subscribers.

1. Log in to Amazon SNS console.
2. In the left navigation pane, select **Subscriptions**.
3. Select **Create subscription** on the subscriptions page.
4. In the **Details** section of the **Create subscription** page, specify the following values.
 - **Topic ARN:** Paste the ARN value copied from previous step ([Step 5: Create a notifications service subscription topic](#)).
 - **Protocol:** HTTPS

- **Endpoint:** Paste the endpoint value you copied while creating the live table feed in previous step ([Step 4: Create and configure a live table feed to use notifications and copy the notification URL](#)).
5. Expand the **Subscription filter policy** section to configure a filter policy. This step is optional.
 6. Expand the **Redrive policy (dead-letter queue)** section to configure a dead-letter queue for the subscription. This step is optional.
 7. Select **Create subscription**.

Note

HTTP(S) endpoints, email addresses, and AWS resources in other AWS accounts require confirmation of the subscription before they can receive messages.

Step 8: Confirm that the endpoint can receive notifications

Where: Database Actions: Live Feeds page

1. Return to the Database Actions Live Feeds page and find the card for the live table feed you are configuring for a notification-based feed.
2. Click the **Actions** (three vertical dots) icon on the card, and select **Show Confirmation URL**.
3. In the **Confirmation URL** dialog box, click the link to confirm the URL. This does not close this dialog box. If the link is successful, a message is displayed that confirms the subscription is active.
4. Return to the **Confirmation URL** dialog box and select the Check only when the cloud store confirmation process is complete check box, and click **OK**.

Once you finish the above steps, any new files uploaded to the bucket will automatically be loaded into the live table feed table.

For more information on how to enable and configure event notifications using the Amazon S3 console, refer [Enabling and configuring event notifications using the Amazon S3 console](#).

Creating a Notification-Based Live Table Feed using Microsoft Azure

A notification-based Live Table Feed is an interface between Oracle Cloud Infrastructure and a third-party cloud message queuing service such as Azure Event Grid.

The following section explains the procedure to generate automatic live feed messages using Microsoft (MS) Azure as the cloud storage. When there is an update in the container and the notification conditions are met, a log message is generated and displayed in the live feed in Oracle Cloud Infrastructure.

To create a notification-based live feed with Microsoft Azure as cloud storage you must:

- [Step 1: Create a resource group in Microsoft Azure](#)
- [Step 2: Create a storage account in Microsoft Azure](#)
- [Step 3: Create Access Keys](#)
- [Step 4: Create a container](#)
- [Step 5: Add cloud storage using Microsoft Azure cloud store](#)

- [Step 6: Create and configure a live table feed to use notifications and copy the notification URL](#)
- [Step 7: Enable Event Resource Provider](#)
- [Step 8: Create Event subscription](#)
- [Step 9: Confirm that the endpoint can receive notifications](#)

✔ **Tip**

To complete the steps above, you will need to alternate between Microsoft Azure portal and Oracle Database Actions pages. You may find it convenient to open the Microsoft Azure portal in one browser page or tab and Database Actions in another, so it is easy to move back and forth.

Step 1: Create a resource group in Microsoft Azure

Where: Microsoft Azure Portal

Resource groups are logical containers where you can manage Azure resources like storage accounts. Resource groups are created so you can deploy, update and delete them as a group. You can create a resource group by following these steps:

1. On the Azure portal, click the **Resource groups** button.
2. Select **Add**.
3. Enter the following values:
 - **Subscription:** Select your Azure subscription, such as Microsoft Azure Enterprise.
 - **Resource group:** Enter a new resource group name, such as resource-group.
 - **Region:** Select your location, such as US west.
4. Click **Review+create**.
5. Click **Create**. It takes a few seconds to create a resource group.

Step 2: Create a storage account in Microsoft Azure

Where: Microsoft Azure Portal

An Azure storage account contains all your storage data objects like blobs, tables, disks etc. You can create a storage account inside the resource group. It provides a unique namespace for your data. To create a storage account, do the following:

1. From the left portal menu, select **Storage accounts** to display a list of your storage accounts.
2. On the Storage accounts page, click the **Create** icon.
3. On the Basic tab, provide the following information for your storage account.
 - **Subscription:** Microsoft Azure Enterprise
 - **Resource group:** resource-group
 - **Storage account name:** teststorage
 - **Region:** Select your location, such as US west.
 - **Redundancy:** Locally-redundant storage (LRS)

4. You can select **Review+create** to accept the default options and proceed to validate the account.
5. After the validation passes, you can proceed to click on **Create storage** account. In case the validation fails, the portal indicates which settings must be modified.

Step 3: Create access Keys

Where: Microsoft Azure Portal

You must grant Microsoft Azure the permissions necessary to obtain access keys on your storage locations. The access keys specific to the storage account are generated automatically after the storage account is created in the previous step. The following steps describes the procedure to create access keys.

1. In **Security+Networking** , select **Access keys**. Your account access keys appear with the complete connection string for each key.
2. Select **Show keys** to show your access keys and connection string for each key and to copy values.
3. Copy the connection string value under key1. This value will be pasted in **Azure storage account access key** text field of next step ([Step 5: Add cloud storage using Microsoft Azure cloud store](#)).
4. Copy the storage account name i.e. teststorage and paste it in **Azure storage account name** text field of next step ([Step 5: Add cloud storage using Microsoft Azure cloud store](#)).
5. Test the credentials to see if it works or not.

Step 4: Create a container

Where: Microsoft Azure Portal

A container is a location (also known as buckets in Amazon S3 and OCI) which holds Azure Blob (Binary large object) storage. Follow these steps to create a container.

1. Navigate to your new storage account in the Azure portal.
2. In the left menu for the storage account, scroll to the **Data storage** section, then select **Containers**.
3. Click the **+Container** icon.
4. Enter the name for your new container. The container name must be lowercase, must start with a letter or number, and can include only letters, numbers, and the dash character.
5. Set the level of **Public Access Level** to Private. The default level is Private.
6. Select **Create** to create the container.

Step 5: Add cloud storage using Microsoft Azure cloud store

Where: Database Actions: Manage Cloud page

1. Click the **Manage Cloud Store** button at the top of the page to go to the Manage Cloud page. For further instructions on adding source files residing in cloud storage provided by Microsoft Azure cloud storage, refer to *Create an Microsoft Azure Cloud Storage Link* topic in [Managing Connections](#) section.

Note

Paste the connection string value under key 1 of previous step [Step 3: Create Access Keys](#) in **Azure storage account access key** text field of **Add Cloud Storage** page. Also paste the storage account name generated in the previous step [Step 3: Create Access Keys](#) in **Azure storage account name** text field of **Add Cloud Storage** page.

Step 6: Create and configure a live table feed to use notifications and copy the notification URL

Where: Database Actions: Live Feeds page

The Live table feed object enables data to be loaded from Microsoft Azure cloud storage with no polling delay. This object creates an integration between Oracle Cloud Interface and Microsoft Azure.

You can configure a new or an existing live table feed to use notifications:

1. Go to the Database Actions Live Feeds page, as described in [Feeding Data](#).
2. Create or edit a live table feed object, as described in [Create a Live Table Feed Object](#) or [Edit a Live Table Feed Object](#). Select the **Enable for Notification** option
3. Click **Create** or **Save**.
4. Click the **Actions** (three vertical dots) icon on the card for your live feed, and select **Show Confirmation URL**.
5. In the **Notification URL** dialog box, click the **Copy** icon to copy the URL to the clipboard. You may want to copy it to a temporary file, so you can retrieve it later. You will use this URL in the subsequent step, ([Step 8: Create Event subscription](#)).

Step 7: Enable Event Resource Provider

Where: Microsoft Azure Portal

If this is the first time you are using Event Grid, you must enable Event Grid resource provider.

1. Select **Subscriptions** on the left menu.
2. Select the subscription you are using for Event Grid i.e. Microsoft Azure Enterprise.
3. On the left menu, under Settings, select **Resource Providers**.
4. Search **Microsoft.EventGrid**.
5. Select **Register**.

It takes a minute for the registration to finish.

Step 8: Create Event subscription

Where: Microsoft Azure Portal

You create an Event Subscription by configuring the subscription and specifying the endpoint that will receive the notifications.

1. Select the storage account you created in [Step 2: Create a storage account in Microsoft Azure](#).
2. Select the **Events** icon in the left navigation pane.
3. Click on **+Event Subscription**.

The **Create Event Subscription** window appears.

1. Specify the following fields in **Event Subscription** details section:
 - **Name:** Eventssub. This is the name of the Event Subscription we create.
 - **Event Schema:** Event Grid Schema
2. Specify the following fields in **Topic Details** section:
 - **Topic Type:** Storage account
 - **System Topic Name:** eventtopic.
3. Specify the following fields in **Event Types** section:
 - **Event Type:** MicrosoftStorage.BlobCreated
4. Specify the following fields in **Endpoint Details** section:
 - **Endpoint Type:** Web Hook
 - **Endpoint:** Paste the Notification URL copied in [Step 6: Create and configure a live table feed to use notifications and copy the notification URL](#).
5. Select **Create**.

This way Microsoft Azure creates a system topic first and then the Event Subscription for the topic.

Step 9: Confirm that the endpoint can receive notifications

Where: Database Actions: Live Feeds page

1. Return to the Database Actions Live Feeds page and find the card for the live table feed you are configuring for a notification-based feed as created in [Step 6: Create and configure a live table feed to use notifications and copy the notification URL](#).
2. Click the **Actions** (three vertical dots) icon on the card, and select **Show Confirmation URL**.
3. In the **Confirmation URL** dialog box, click the link to confirm the URL. This does not close this dialog box. If the link is successful, a message is displayed that confirms the subscription is active.

Note

The **Confirmation URL** link expires after a few minutes. You must make sure that you click the link before it expires.

4. Return to the **Confirmation URL** dialog box and select the **Check only when the cloud store confirmation process is complete** check box, and click **OK**.

Once you finish the above steps, upload a new file to the Microsoft Azure container you created in [Step 4: Create a container](#).

1. Navigate to the container you created.
2. Select the **Container** to show a list of blobs it contains.
3. Select **Upload** button to open your local repository and browse the file you need to upload as a block blob.
4. Select the **Upload** button to upload the blob.
5. You can now view the new blob listed within the container.

6. Return to the Database Actions Live Feeds page and find the card for the live table feed you are configuring for a notification-based feed.
7. Click the **Actions** (three vertical dots) icon on the card, and select **Live table feed Run Details**.

You should be able to view logs for the blob uploaded to the Live Feed table from the Microsoft Azure storage in **Live table feed Run Details** window.

For more details on how to create a topic and subscription in Azure portal, refer to [Azure Event Grid Notifications](#).

Send Email using Live Feed Tool

You can configure the Live Feed tool to send you email notifications when certain specific live feed events occur.

Prerequisite

1. Before subscribing to notifications, make sure that you have first created a live feed.
2. **Configure the mail service:**
For the live feed tool to send notifications, you need to configure the Simple Mail Transfer Protocol (SMTP) Settings in the Data Studio Settings.

Refer to the [Send Email on Autonomous Database](#) chapter for more details.

Note

The Live Feed tool uses the UTL_SMTP package in the back end to provide an interface. While UTL_SMTP itself does not inherently support encryption like OAuth2, which is required by some providers (e.g., Microsoft 365), using a secure connection (SSL/TLS) can encrypt the communication between your application and the SMTP server. However, UTL_SMTP does not natively support OAuth2, which is becoming increasingly common for authentication with large email providers.

The following rules apply while configuration:

- The SMTP server port must be 25 or higher. The only ports likely to be used here are 587 (most common for modern setups, typically with StartTLS), 465 (Secure Sockets Layer) or 25 (no encryption).
- The SMTP server host name must not be blank.
- The E-mail address field must not be blank.
- Typically, clients will need to provide authentication credentials to access their SMTP server. Additionally, most setups will likely use StartTLS for encryption. However, the specific requirements can vary depending on how the company or provider has configured their system.

You can configure the email notifications in the following locations:

[Livefeed page](#)

[Data Load Dashboard](#)

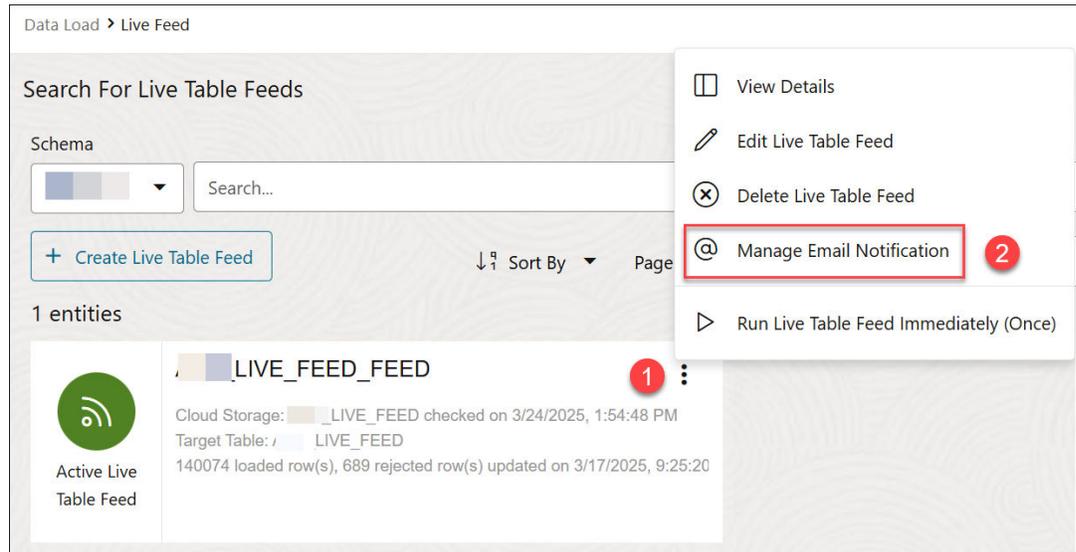
[Catalog page](#)

Configure Email Notifications on the Live Feed tool

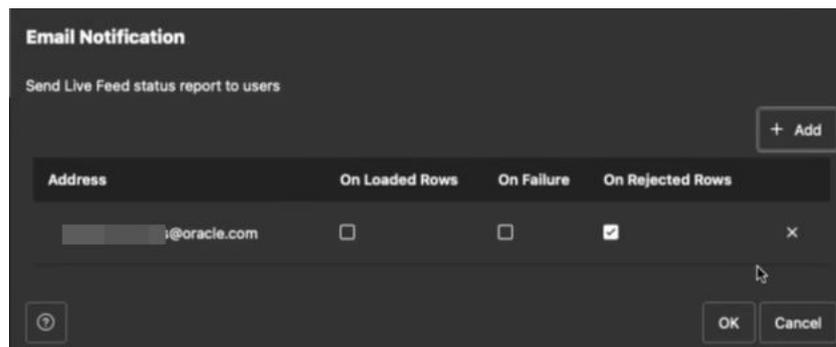
After you have created a Live Table Feed using the Live Feed tool, navigate to the Live Feed tool .

You will view the Live Table Feed card on the Live Feed tool page.

1. Click the **Actions** icon on the card and select **Manage Email Notification**.



2. On the Email Notification dialog, select **+Add** to add your email address where you will receive the notification.



3. Enter the email address.
4. Select any of the available options to receive email notifications based on the following conditions:
 - **On Loaded Rows:** Select this option if you want to be notified in case the Live Feed successfully imports new data into your table.
 - **On Failure:** Select this option if you want to be notified in case the Live Feed fails to run due to an error. For example, if the storage location for your files is missing, or your access credentials are no longer valid.

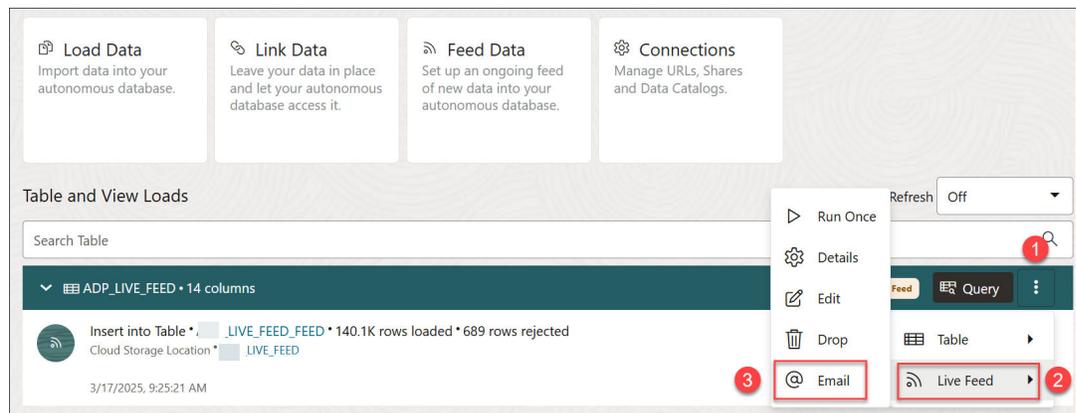
- **On Rejected Rows:** Select this option if you want to be notified in case the Live Feed encounters and rejects incorrect data. For instance, when there is a data type mismatch, such as attempting to import the word `Blue` into a column that must consist of only numbers.

5. Click **OK**.

Configure Email Notifications on the Data Load Dashboard

Once you have created a Live Feed, you will view the status of the feed on the Data Load Dashboard.

1. Click the **Actions** menu on the table header, click **Live Feed** and select **Email**.



2. On the Email Notification dialog, select **+Add** to add your email address where you will receive the notification.
3. Enter the email address.
4. Click **OK**.

Configure Email Notifications on the Catalog Tool

You can view the Live Feed entity on the Catalog page after you create it.

1. Select **Manage Email Notifications** the Live Feed entity.
2. **Continue with the wizard:** Progress to the next steps in the **Manage Email Notifications** wizard by following the same process you used previously.

After you have configured the email notifications, the subscriber with the email address specified in the Email Management notification dialog will now receive notifications according to the settings you configure during the live feed creation process.

DBMS_LIVE_FEED Package Reference

This chapter provides information about the packages you use with the Data Load Tool in Data Studio. The `DBMS_LIVE_FEED` topic also covers the procedures included in the `DBMS_LIVE_FEED` package.

Topics:

[Summary of DBMS_LIVE_FEED Subprograms](#)

- [DBMS_LIVE_FEED Procedures](#)

- [Summary of DBMS_LIVE_FEED Subprograms](#)
The DBMS_LIVE_FEED package simplifies common operations like running a live table feed on demand, on a schedule, or as the result of a notification driven updates to a table based on new objects in object store buckets.

Summary of DBMS_LIVE_FEED Subprograms

The DBMS_LIVE_FEED package simplifies common operations like running a live table feed on demand, on a schedule, or as the result of a notification driven updates to a table based on new objects in object store buckets.

The following table lists the DBMS_LIVE_FEED subprograms and briefly describes them.

DBMS_LIVE_FEED Package Subprograms

Subprogram	Description
create_for_storage_link	This procedure creates a live feed for storage link.
create_for_directory	This procedure creates a live feed for directory.
create_for_url	This procedure creates a live feed from URL.
modify_live_feed	This procedure modifies a live feed.
drop_live_feed	This procedure drops a live feed.
load_new_content	This procedure loads new content in an existing Live Feed.
get_notification_key	This procedure retrieves Notification Key from a Live Feed.
get_confirmation_url	This procedure receives confirmation URL from a Live feed.
get_notification_state	This procedure receives notification state from a Live Feed.

Create For Storage Link Procedure

This procedure creates a live table feed from a cloud storage link. A cloud storage link is a named association between an OCI bucket URI, and a local credential name.

Syntax

```
procedure create_for_storage_link(live_feed_name      IN VARCHAR2,
                                table_name           IN VARCHAR2,
                                storage_link_name    IN VARCHAR2,
                                schema               IN VARCHAR2
                                DEFAULT NULL,
                                table_schema         IN VARCHAR2
                                DEFAULT NULL,
                                column_list          IN CLOB
                                DEFAULT NULL,
                                object_type          IN VARCHAR2
                                DEFAULT NULL,
                                object_format       IN VARCHAR2
                                DEFAULT NULL,
                                object_filter       IN VARCHAR2
                                DEFAULT NULL,
                                object_filter_type  IN VARCHAR2
                                DEFAULT NULL,
                                scheduled            IN BOOLEAN
                                DEFAULT NULL,
```

```

                                scheduler_interval IN VARCHAR2
DEFAULT NULL,
                                notifications      IN BOOLEAN
DEFAULT NULL,
                                job_class          IN VARCHAR2
DEFAULT NULL,
                                start_date        IN TIMESTAMP
WITH TIME ZONE DEFAULT NULL,
                                end_date          IN TIMESTAMP
WITH TIME ZONE DEFAULT NULL);

```

DBMS_LIVE_FEED Parameters

If you are accessing DBMS_LIVE_FEED procedures, use the following parameters.

Parameter	Description
live_feed_name	The name of the Live feed which you want to create.
storage_link_name	The name of the cloud storage link. The name of the link should follow standard Oracle naming conventions.
table_name	The name of the Autonomous Database target table into which the live feed will load data from Cloud Object Storage.
Object_filter	The name of the target table that data from the live feed will be loaded into in your Autonomous Database instance to access the Cloud Object Storage.
object_filter_type	The regular expression to limit the live table feed to only those files in the bucket that match the expression. . It could be a regular expression or a glob, depending on the object_filter_type parameter.
scheduled	The default is TRUE. If TRUE, you can set up a schedule for running the live table feed object.
notifications	The default is FALSE. When TRUE, the Live Feed tool allows your livefeed to be notified by object storage that there are new files to be loaded.
schema	The name of the schema. The name of the schema of the livefeed. Only the current schema is supported at the moment.
table_schema	The schema of the target table.
object_type	The type of the object. The type of files you can load are CSV, JSON and Parquet.
object_format	This is one of the SQL Loader options. It accepts format received by DBMS_CLOUD . COPY_DATA procedure.
scheduler_interval	This parameter displays the interval at which the livefeed runs using the DBMS_SCHEDULER syntax.
job_class	The Scheduler job class. The values are TPURGENT , TP , HIGH , MEDIUM and LOW.
start_date	The starting date for the live feed job.
end_date	The ending date of the Live feed job.

Note

The parameters of all the procedures described below are listed in the above table.

Example

In this example, a live feed named MYLIVEFEED is created on the given URL.

```
SQL> BEGIN
  dbms_live_feed.create_for_storage_link(
    live_feed_name => 'MYLIVEFEED',
    storage_link_name => 'DATATOOLS_LF_BUCKET',
    table_name => 'MALTESE',
    object_filter => 'fndcalday?.csv',
    object_filter_type => 'GLOB',
    scheduled => false,
    notifications => false);
end;
/;
PL/SQL procedure successfully completed.
```

Create For Directory Procedure

This procedure creates a live table feed from a directory.

Syntax

```
procedure create_for_directory(live_feed_name      IN VARCHAR2,
                              table_name         IN VARCHAR2,
                              directory_name     IN VARCHAR2,
                              schema             IN VARCHAR2 DEFAULT NULL,
                              table_schema      IN VARCHAR2 DEFAULT NULL,
                              column_list       IN CLOB      DEFAULT NULL,
                              object_type       IN VARCHAR2 DEFAULT NULL,
                              object_format     IN VARCHAR2 DEFAULT NULL,
                              object_filter      IN VARCHAR2 DEFAULT NULL,
                              object_filter_type IN VARCHAR2 DEFAULT NULL,
                              scheduled         IN BOOLEAN  DEFAULT NULL,
                              scheduler_interval IN VARCHAR2 DEFAULT NULL,
                              notifications     IN BOOLEAN  DEFAULT NULL,
                              job_class         IN VARCHAR2 DEFAULT NULL,
                              start_date        IN TIMESTAMP WITH TIME
                              ZONE DEFAULT NULL,
                              end_date          IN TIMESTAMP WITH TIME
                              ZONE DEFAULT NULL);
```

Create For URL Procedure

This procedure creates a live table feed from a URL.

Syntax

```
procedure create_for_url(live_feed_name      IN VARCHAR2,
                        table_name         IN VARCHAR2,
                        url                 IN VARCHAR2,
                        credential_name     IN VARCHAR2,
                        schema             IN VARCHAR2 DEFAULT NULL,
                        table_schema      IN VARCHAR2 DEFAULT NULL,
                        column_list       IN CLOB      DEFAULT NULL,
```

```

object_type          IN VARCHAR2 DEFAULT NULL,
object_format        IN VARCHAR2 DEFAULT NULL,
object_filter        IN VARCHAR2 DEFAULT NULL,
object_filter_type   IN VARCHAR2 DEFAULT NULL,
scheduled            IN BOOLEAN  DEFAULT NULL,
scheduler_interval   IN VARCHAR2 DEFAULT NULL,
notifications        IN BOOLEAN  DEFAULT NULL,
job_class            IN VARCHAR2 DEFAULT NULL,
start_date           IN TIMESTAMP WITH TIME ZONE
DEFAULT NULL,
end_date             IN TIMESTAMP WITH TIME ZONE
DEFAULT NULL);

```

Modify Live Feed Procedure

This procedure modifies a live table feed.

Syntax

```

procedure modify_live_feed(live_feed_name  IN VARCHAR2,
                           schema          IN VARCHAR2 DEFAULT NULL,
                           scheduled       IN BOOLEAN  DEFAULT NULL,
                           scheduler_interval IN VARCHAR2 DEFAULT NULL,
                           notifications    IN BOOLEAN  DEFAULT NULL,
                           job_class       IN VARCHAR2 DEFAULT NULL,
                           start_date      IN TIMESTAMP WITH TIME ZONE
DEFAULT NULL,
                           end_date        IN TIMESTAMP WITH TIME ZONE
DEFAULT NULL);

```

Example

In this example, a live feed named MYLIVEFEED is modified.

```

begin
  dbms_live_feed.modify_live_feed(
    live_feed_name => 'MYLIVEFEED',
    scheduled => true,
    scheduler_interval => 'FREQ=daily;INTERVAL=1',
    notifications => true);
end;
/

```

Drop Live Feed Procedure

This procedure drops a live table feed.

Syntax

```

procedure drop_live_feed(live_feed_name  IN VARCHAR2,
                        schema            IN VARCHAR2 DEFAULT
NULL);

```

Example

In this example, a live feed named MYLIVEFEED is dropped.

```
begin
  dbms_live_feed.drop_live_feed(live_feed_name => 'MYLIVEFEED');
end;
/
```

Load New Content Procedure

This procedure loads new content in the feed.

Syntax

```
procedure load_new_content(live_feed_name IN VARCHAR2,
                           schema          IN VARCHAR2 DEFAULT NULL);
```

Get Notification Key Procedure

This procedure receives Live Table Feed Notification Key. It is used for a live feed with notifications enabled to construct a notification URL to provide a cloud service. Follow the instructions in the [Creating a Notification-Based Live Table Feed](#) chapter to create a notification based live feed.

Syntax

```
function get_notification_key(live_feed_name IN VARCHAR2,
                              schema         IN VARCHAR2 DEFAULT
NULL)
  return VARCHAR2;
```

Get Confirmation URL

This procedure receives confirmation URL which verifies that the endpoint can receive notifications.

You must visit this URL to enable notifications you receive from a cloud service. It is not possible to confirm the enabling of notifications automatically due to database firewall restrictions. Click this URL to enable notifications else you will not receive any notifications.

Syntax

```
function get_confirmation_url(live_feed_name IN VARCHAR2,
                              schema         IN VARCHAR2 DEFAULT
NULL)
  return VARCHAR2;
```

Get Notification State

This procedure receives the state of notification of the Live Feed job execution.

Syntax

```
function get_notification_state(live_feed_name IN VARCHAR2,
                               schema          IN VARCHAR2 DEFAULT
```

```
NULL)  
    return VARCHAR2;
```

13

The Catalog Tool

The Catalog Tool allows you browse, search and discover data in, and connected to, the Oracle Autonomous Database.

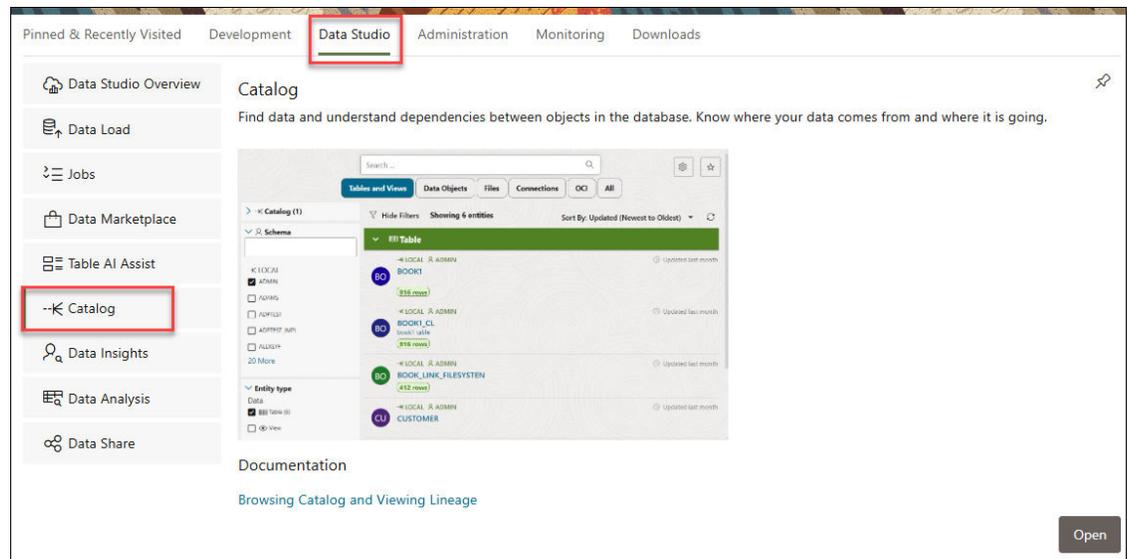
The Autonomous Database Catalog is a multi-catalog tool that provides a way to search for data and other objects in your currently connected Autonomous Database, and also in a wide range of other connected systems.

The Catalog tool enables you to search, find, load, or link connected data assets from anywhere in the cloud and beyond. It also enables you to mount new catalogs over other systems and search for data and other items in the connected Autonomous Database.

The Catalog tool can be used to mount new catalogs over:

- Other Autonomous Databases in your tenancy.
- Any other database that can be connected using a DB Link, for example an on-premises database that your Autonomous Database can connect to.
- Shared data, for example data shared from DataBricks using Delta Sharing.
- Existing external data catalogs such as AWS Glue, or the OCI Data Catalog.

To reach the Catalog tool, select the **Catalog** menu in the **Data Studio** tab of the Launchpad.



or click the Selector  icon and select **Catalog** from the Data Studio menu in the navigation pane.

Note

If you do not see the Catalog card then your database user is missing the required DWROLE role.

Run the following command to grant DWROLE to the user:

```
GRANT DWROLE TO ANALYST;
```

The following topics describe the features, capabilities, and technical architecture of the Catalog tool:

- [About the Catalog Page](#)
Use the Catalog Tool to browse, search and discover data either in your Oracle Autonomous Database, or connected to it and get information about the entities in and available to your Oracle Autonomous Database. You can see the data in an entity, the sources of that data, the objects that are derived from the entity, and the impact on derived objects from changes in the sources.
- [Browse and Search Catalogs](#)
The Catalog Tool aims to allow you to find data quickly and easily, whether or not the data is currently resident in the connected Autonomous Database.
- [Sorting and Filtering Entities](#)
On the Catalog page, you can sort and filter the displayed search results.
- [Filter Entities](#)
By selecting different quick filters, the faceted results of the entity types also vary.
- [Display Entity Search](#)
If you do not select a quick filter, the **Tables and Views** filter is selected by default.
- [Catalog Settings](#)
The Catalog Settings can manage the general settings, add, and view query scopes, create and view saved searches and change the look and feel of the Catalog page.
- [Manage Catalogs](#)
You can now enable multiple catalogs using the Catalog Tool.
- [Viewing Entity Details](#)
When you have found an entity you are interested in, you can click the entity to view its details, or click one of the other available actions from the menu that appears on the right hand side when you hover over that row.
- [Registering Cloud Links to Access Data](#)
Cloud Links enable you to remotely access read only data on an Autonomous AI Database instance.
- [Export Data to Cloud](#)
Use the **Export Data to Cloud** menu in the table actions of the Browse Catalog page to export data as text from an Autonomous Database to a cloud Object Store. The text format export options are CSV, JSON, Parquet, or XML.
- [Gathering Statistics for Tables](#)
You can generate statistics that measure the data distribution and storage characteristics of tables.

- [Editing Tables](#)
You can create and edit objects using Edit Table wizard available from the **Edit** menu in **Actions** (three vertical dots) besides the table entity.
- [Manage Catalogs with DBMS_CATALOG](#)
The `DBMS_CATALOG` package provides a comprehensive set of procedures, functions, and types for managing database catalogs for Oracle Autonomous AI Database.

About the Catalog Page

Use the Catalog Tool to browse, search and discover data either in your Oracle Autonomous Database, or connected to it and get information about the entities in and available to your Oracle Autonomous Database. You can see the data in an entity, the sources of that data, the objects that are derived from the entity, and the impact on derived objects from changes in the sources.

The Catalog page lists and displays details about:

- Database objects such as tables, columns, views, cloud objects, analytic views, packages, procedures, shares, share recipients, catalogs, catalog storage links, live table feed, share providers.
- Other types of object connected to the database, such as Cloud Storage Links, Cloud Storage Buckets, Cloud Objects (files on connected Cloud Storage), Compartments, Data Catalog Links, Catalog Storage Links, Autonomous Database.
- Database data dictionary objects such as tables, columns, views, database links, analytic views, packages, and procedures that have been created by the database tools or a database application such as SQL Developer.

The Catalog tool now provides a unified search and discovery of the following data assets that are connected to the Autonomous Database:

- Data in connected databases – for example, other Autonomous Databases in the same tenancy.
- Data in connected cloud storage systems – for example, in OCI, AWS S3, Azure, or Google Cloud Platform.
- Data in registered data catalogs – for example, in OCI Data Catalog, or AWS Glue.
- Data in registered data shares – for example, shared from Data Bricks.

Catalog Tool Terminology

The following are some major terms related to the Catalog Tool:

- **Catalog:** A catalog consists of one or many schemas, each with one or many tables. A catalog simply represents the set of data in that database as it is organized into schemas and tables. The way you connect to the catalog determines the level of access to those schemas and tables. For other types of catalog, for example, catalogs that are mounted over external data catalogs or shares, the objects in the external system are still organized into schemas and tables but may not exist physically in such a form. For example, a catalog that is mounted over an AWS Glue data catalog may contain tables that actually exist as multiple files on AWS S3 cloud storage.
- **Mounting a Catalog:** This is the process of adding a new catalog connection to an external system so that the metadata in that system can be represented in the Autonomous Database catalog in a standard form.

A **Local** catalog is all the objects in or directly connected to your Autonomous Database as described in the above points. In addition to the local catalog, Catalog Explorer can add in additional catalogs so that you can discover data from other sources.

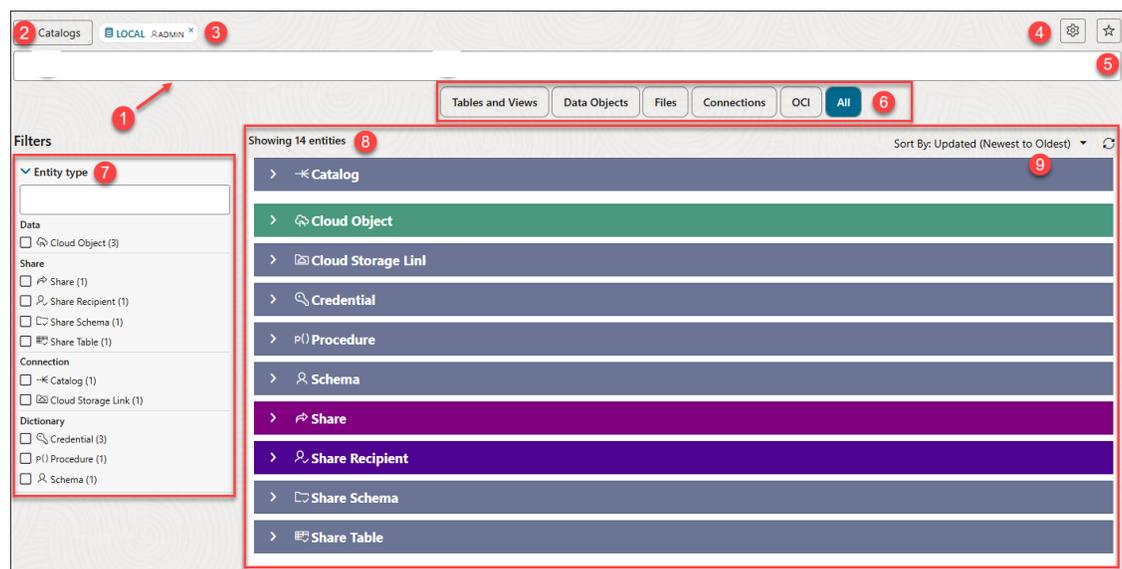
Additional catalogs can be mounted over:

- [Mount Autonomous Database as Catalog](#): You can add a catalog to another autonomous database in your tenancy, which will allow you to search and discover all of the objects in that database based on the permissions for connection.
- [Mount Database Link](#): The tool can mount an existing Database Link. Where a database link already exists, for example to an on-premises Oracle database, this can be mounted as a catalog. As with adding a catalog over an Autonomous Database, this option enables all the objects that are available to connection for search and discovery.
- [Mount Data Share](#): The tool can mount registered live share or delta share as a catalog.
- [Mount Amazon Glue](#): A catalog can be mounted over an AWS Glue Data Catalog database region.
- [Mount OCI Data Catalog](#): The tool can mount data registered in OCI Data Catalog.
- [Mount a Catalog over an Iceberg Data Catalog](#): The tool can add a catalog over an Iceberg Data Catalog.

Once a new catalog has been mounted, you can add the catalog in the Catalog tool, and search and discover data from that catalog, or across many catalogs.

When you first open the Browse catalog page, it contains the **Search catalog** field, the **Show Catalog Settings**  button, **Catalogs** icon, a list of recently viewed objects, and several suggested searches. You can enter a search string, click on one of the recent objects to see its details, or click on **Show search suggestions** icon to view suggestions on the right side of the page.

As soon as you select Catalog menu from Data Studio menu bar, the page is displayed as below.



When you hover your mouse over a particular catalog entity, it will show a floating toolbar of common actions. The list is context-sensitive depending on the capabilities available for the particular entity. For example, tables in the current database offer more possibilities than tables in remote catalogs.

The Catalog page contains:

1. Search field

You can search for the entities in the search field. For more details see [Browse and Search Catalogs](#).

2. Catalogs field

Clicking this icon opens the **Catalogs** dialog box.

You can view a list of all mounted catalogs and schemas in this dialog box. You can also add catalogs from the **Manage** tab from the **Catalogs** dialog box.

Select specific schemas from a database of your choice to refine your search.

- To show tables of only the **ADMIN** schema from the default **local** schema:
 - On the **Catalogs** dialog box, click the **Select** tab, and type **ADMIN** from the **Select Schemas** search field.

Note

The default LOCAL catalog is already selected.

The screenshot shows the 'Catalogs' dialog box with the following elements:

- Buttons: Refresh, Add
- Tabs: **Select** (highlighted), Manage
- Text: Select Catalogs and its Schemas to explore entities
- Options:
 - All Catalogs
 - LOCAL
 - All Schemas
- Search field: ADMIN X
- Option: -- YETI
- Buttons: Apply, Cancel

- Select the **Manage** tabs to [Manage Catalogs](#).

3. Default Database name and Schema name

This field displays the default database name along with the schema name.

4. Catalog Settings

Refer to [Catalog Settings](#) for detailed information.

5. Saved Search Suggestions

Click to open or close the **Saved Search Suggestions** panel.

6. Filters option

Click any of the filters icon to filter and refine the search results.

Refer to [Browse and Search Catalogs](#) for detailed information.

You can also **Query Scopes** to filter options to search for catalogs and save the search.

Refer to **Query Scopes** in [Catalog Settings](#) for more information.

7. Faceted Filters panel

Select one or more filter values to limit the entities shown on the page. Only those entities that match the filter values are shown. That is, the items returned by a search are filtered by these filter settings. The filters also change depending on the quick filters you select.

See [Filter Entities](#).

8. Display area

The area beneath the **Search catalog** field displays the entities returned by a search and that match the filter criteria set in the **Filters** panel. You can sort the entities by clicking the **Sort By** button and then setting sort values.

Hover besides the name of the entity to view details about the entity.

9. Toolbar

The toolbar appears after you run an initial search. It contains these buttons:

- **Sort By** 

To select sorting values, click the **Sort By** button to open the list of options. Then click the **Ascending** or **Descending** icon next to **Name** or **Updated**. For example, if you select the **Ascending** icon next to **Name** the entities will be sorted in alphabetical order by entity name. If you select **Descending** icon next to **Updated**, the entities will be sorted in reverse of the latest updated entity..

Click **Reload** to refresh the choices in the list.

See [Sorting and Filtering Entities](#) for more information.

Browse and Search Catalogs

The Catalog Tool aims to allow you to find data quickly and easily, whether or not the data is currently resident in the connected Autonomous Database.

Using the Catalog tool, you can discover data in connected systems, preview it, and choose whether or not to load, link or feed the data into the local database.

You can view the search bar on the Catalog's home page. To enter a search term, click on the search field located at the top. You can filter the search entry by selecting a quick filter displayed below the search field.

You must first select the type of assets you want to search from the filter buttons at the top of the screen, then search for objects by name.

Selecting the quick filters helps you display filtered searches in the display area. You can combine multiple filters. They are:

- **Tables and Views**

This filter searches only for the database tables (both internal and external tables) and views (this includes both standard database views and analytic views)

- **Data Objects**
This filter searches for data objects in the database and on connected cloud storage. This is a superset of the **Tables and Views** and **Files** scopes.
- **Files**
This filter searches only for files in connected cloud storage.
- **OCI**
This filter searches for other Oracle Cloud Infrastructure objects, such as other Autonomous Databases, buckets on OCI cloud storage, or a registered OCI Data Catalog. This field is available if you have an OCI credential. You can create an OCI credential from the [Data Studio Settings](#) icon.
- **Connections**
This filter searches for connections that are registered in Data Studio, such as connections to external Data Catalogs or databases.
- **All**
This filter searches for all objects in the catalog. Since there are many thousands of such objects, this scope may be slower to use.

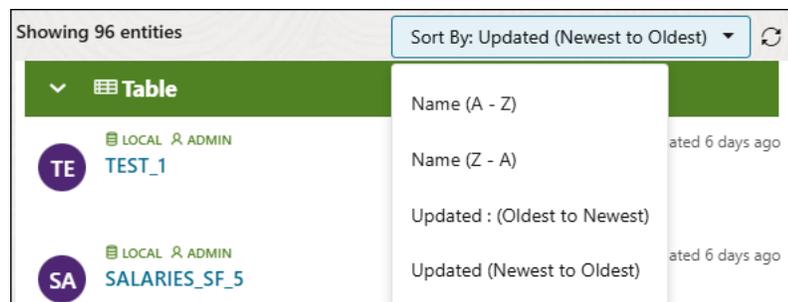
For example, to find all connections that may contain data related to testing, click **Connections** and search for *test*:

This displays all connections that contain test data.



Sorting and Filtering Entities

On the Catalog page, you can sort and filter the displayed search results.



Sort Entities

To set a sort order for the entities on the page,

1. Click **Sort by** on the toolbar.

2. Find the value you want to use to sort, for example **Entity Name**.
3. Click the **Name (A - Z)** (Ascending) icon or the **Name (Z - A)** (Descending) icon next to that value.
4. You can set only one parameter. You can also set **Updated: (Oldest to Newest)** or **Updated: (Newest to Oldest)** based on the time you update the entity.

Filter Entities

Restrict which entities that are returned by a search are displayed on the page by setting filters in the **Filters** suggestions panel on the top of the Catalog page. Select one or more filter values. Only those entities that are returned by a search and that match the filter values are shown. Selecting all or none of the options shows all entities returned by the search.

By default, system tables and private tables are not displayed. To display them, click **Catalog**

Settings  in the toolbar and then click the **Show system tables** slider or the **Show private tables** slider, or both.

Filter Entities

By selecting different quick filters, the faceted results of the entity types also vary.

The faceted search displays various options in entity types based on different categories. You can select multiple entity types in various categories for more specific search results. It displays the count value in bracket.

The following table lists the entity types and filter condition subcategories with their description:

Table 13-1 Entity Types and Filter Condition

Field	Description
Filter for	
Entity Type	Specifies the name of the entity type under which you select entity filters
Filter condition subcategory	Data
Data Entity type values	<ul style="list-style-type: none"> • Table • View • Analytic View • Column • File • Cloud Object
Share Entity type values	<ul style="list-style-type: none"> • Share • Share Recipient • Cloud Link Namespace • Share Schema • Share Table

Table 13-1 (Cont.) Entity Types and Filter Condition

Field	Description
Connection Entity type values	<ul style="list-style-type: none"> • Catalog • Cloud Storage Link • Data Catalog Link • Live Table Feed • Share Provider • Cloud Link • Data Catalog Asset • Database Link • Share Link
Dictionary Entity Type Values	<ul style="list-style-type: none"> • Attribute • Attribute Dimensions • Directory • Credential • Function • Level • Mining Model • Procedure • Synonym • Schema
OCI	<ul style="list-style-type: none"> • Autonomous Database • Bucket • Compartment • Data Dictionary • Database Connection • Data Catalog • Vault • Vault Secret

Display Entity Search

If you do not select a quick filter, the **Tables and Views** filter is selected by default.

You can also view the Table after you enter the search term, a search is performed across **Tables and Views**.

Expand the Table header to view the Table name.



1. You can view the table name. The table header icon color uniquely reflects the entity. All entities within the same schema or catalog have the same entity color.
2. **Auto-generate description:** You can also Auto generate a description by clicking the **Add description** link.

On the **Add Description** wizard, click **Auto-Generate**. You need to set up an AI profile using the [Data Studio Settings](#) icon to configure the profile. Click **OK** on the **Add description** wizard. You can now view the auto generated description. Your profile must be AI enabled to generate auto description.

3. Viewing Entity Details

You can view the entity details of the selected entity by just clicking it. You can view the menu of the available actions by hovering over the entity. The icons vary from entity to entity.

For **Table and Views** type of entity, you can view the set of available actions of the entity:

- **View Details:** Select **View Details** to view details about the table.
- **Gather Statistics:** Select **Gather Statistics** to display new statistics after you modify the table's structure. Refer to the [Gathering Statistics for Tables](#) for more detailed information.
- **Create Share:** Select **Create Share** to share from the associated table.
- **Register to Cloud Link:** Select **Register to Cloud Link** to register the table for remote access for a selected audience you define. See [Registering Cloud Links to Access Data](#) section for more information.
- **Create Analytic View:** Select **Create Analytic View** to create an Analytic View from the selected associated table. See [Creating Analytic Views](#)
- **Query:** Selecting **Query** takes you to the Data Analysis page with the selected table.
- **Export Data to Cloud Storage Location:** Select [Export Data to Cloud](#) to export data to a cloud object store.
- **Edit:** Select **Edit** to edit the properties of the table. Refer to [Editing Tables](#) for more information.
- **Drop:** Select **Drop** to delete the table.

Catalog Settings

The Catalog Settings can manage the general settings, add, and view query scopes, create and view saved searches and change the look and feel of the Catalog page.

You can alter the Catalog tool's user interface and page layout through themes and theme styles in the Catalog Settings.

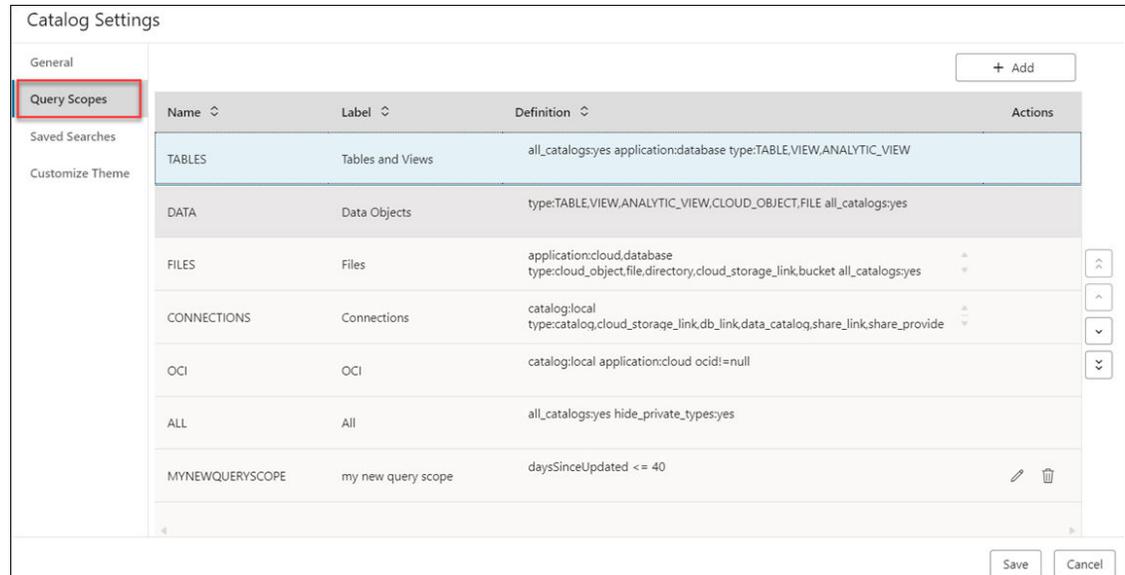
Select **Show Catalog Settings** on the Catalog page to set the behavior of the Catalog. When you set these options, they take place immediately and are also saved as the default behavior for the page. Clicking **Show Catalog Settings** opens the General tab where you can view the following options.

On the General tab, you can view the following options:

- **Show system tables**
Select this option to include system tables in the search results.
- **Show private tables**
Select this option to include private tables in the search results.
- **Page size**
Select the number of entities to display on the page.

Gradually progress to use the **Query Scopes** tab to view, create, and delete query scopes. You can search for catalogs and save this search using query scopes.

Selecting Query Scope tab icon enables you to create new query scopes based on your preferences.



You can view, create, and edit the previously saved query scopes in this wizard. The query scopes are categorized based on by whom its created.

Click **+Add** to create a Query Scope. Click **Cancel** to cancel its creation.

Specify the following fields in the Query Scope tab.

- **Name:** Enter the name of the Query Scope. This is a mandatory field.
- **Label:** This is a mandatory field. Enter a descriptive name here. You will use this field to refer to a query scope.
- **Definition:** Enter the Oracle Autonomous Database Data Definition Language (DDL) that creates the search entity. This is the same search criteria you enter in the Search Catalog field.

Once you have added the new query scope, it is visible in the list of query scopes in the Catalog Settings wizard.

Progress to the Saved Searches tab.

You can save your time from redefining the same search again in the future. You can diagnose problems faster since you are just few clicks away from accessing a saved search. Here is how you can create a saved search.

- Select this tab to save the five previous search queries that you entered into the **Search catalog** field. When the **Search catalog** field is empty and you click in the field, the last five search queries are listed in the drop-down list. (Any predefined searches selected from the **Suggestions** panel won't appear in this list, unless you edited them to make a new search.)

The **Saved Searches** tab saves the current search query as per the criteria you specify or create a new saved search. Click **+Add** to create a new saved search.

- On the **Saved Searches** tab, click **+Add** to create a new search.
 - On the **Create New Search** dialog, specify the **Title**, **Scope**, **Definition** and **Description** (optional field) in their respective fields. For example, you want to search for a specific Entity type and a specific owner.
 - Select **Save**.

After the creation of the saved search, it appears on the list of Saved Searches.

You can change the columns displayed in the search results by clicking the pencil icon in the Actions column. Click the **delete** icon in the Actions column to delete the search you save. The saved searches you create are available for selection in the Saved Search panel in the right of the Catalog page.

Progress to the Customize Theme tab to expand each entity type to select the properties and preview it.

- – On the **Customize Theme** tab, you can select the entity type whose appearance you want to define by editing the options in this tab. You can view the following menu options as you move from left to right.

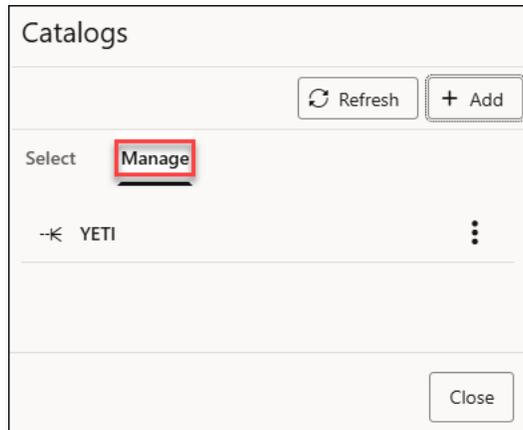
Options	Description
Entity Type	This option describes the type of entity you want to edit. You can expand the entity type to select properties and to preview what the change looks like.
Is actions toolbar?	This option when enabled displays the actions menu items as floating toolbar on hover, else displays the actions menu.
Header background color	This option enables you to set the background color of the title of the entity header. This is the same as the text color for the properties of the entity listed.
Text color	This option enables you to set the text colour of the entity title.
Text highlighter	This option enables you to set the highlight colour of the box surrounding the properties of the entity.

Click **Save** in the Catalog Settings dialog to save all your preferences you set for the Catalog page.

Manage Catalogs

You can now enable multiple catalogs using the Catalog Tool.

You can mount additional catalogs from other Autonomous Databases in the same tenancy, so that you can search for data across multiple databases. Select the **Manage** tab from **Catalogs** dialog box.



Mount Autonomous Database as Catalog

You will use an existing Autonomous Database to create a Database link and provide catalog name to mount it.

Note

To mount an Autonomous Database as a Catalog, you must have the `CREATE DATABASE LINK` and `CREATE ANY DIRECTORY` system privileges. The ADMIN user is granted `CREATE DATABASE LINK` and `CREATE ANY DIRECTORY` system privileges. The ADMIN user can grant `CREATE DATABASE LINK` and `CREATE ANY DIRECTORY` system privileges to other users.

To add an Autonomous Database as a catalog, you need to perform the following operations:

- Store your object storage credentials in your Autonomous Database
- Create a database link or use an existing database link
- Mount the database link as a catalog

Refer to [Connections](#) to view the connections that are established from the Data Studio to the cloud, catalogs and shares.

1. On the Catalog page, select **Manage Catalog**. Click **+Add**.
2. On the **Add Catalog** page, select **Autonomous Database** as Catalog Source.

Add Catalog

1 **Catalog source** 2 **Autonomous Database**

- Autonomous Database**
Add a catalog over another Autonomous Database in your tenancy.
[Show more...](#)
- Database Link**
Add a catalog over another database to which you have a Database Link already configured.
- Data Share**
Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.
- Amazon Glue**
Add a catalog over an Amazon Web Services Glue data catalog region.
- OCI Data Catalog**
Add a catalog over an OCI Data Catalog.
- Iceberg catalog**
Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

Back Next Add Cancel

3. Select the **Database name** from the drop-down. For example, *TESTING*.
4. You can view the following fields after you select the database name. Specify the following field values:

Add Catalog

1 **Catalog source** 2 **Autonomous Database**

Database name *
TESTING

Database link will be created first and then it will be mounted as Catalog

Catalog name *
.TESTING

Database link name *
.TESTING_LINK

Database credential * ⓘ
AWSGLUE_CRED Create Credential

Hide Advanced Options

Wallet directory name *
WALLET_DIR-TESTING

Show code

Test Create Back Next Cancel

- **Catalog Name:** Enter the name of the catalog you want to mount the database link as. For example, *TESTING*.
- **Database link name:** Enter the name of the database link the tool creates using the existing database. For example, *TESTING_LINK*.
- **Database credential:** Select the credential from the drop-down which is required to create the database link. For example, *AWSGLUE_CRED*. Click **Create Credential** in case there is no existing credentials.
- Under **Show Advanced Options**, select the **Wallet Directory name** from the drop-down. You need to have permission to access the directory. For example, *WALLET_DIR-TESTING*.

Note

The wallet file, along with the Database user ID and password provide access to data in the target Oracle Autonomous Database.

- Click **Test** to test the creation of the Database Link.

Click **Create** to create the Autonomous Database as Catalog.

After successful creation, you can view the Autonomous Database listed under Manage Catalog.

Mount an existing Database Link as Catalog

You can instantiate a catalog schema as a local database user. If a database link has already been set up, you can mount a catalog over the linked database in order to search and discover data sets in that database.

Mounting a catalog requires the following information:

- Name of the database link
- Cloud credentials required to access the data

You can add a database link to one of the databases and mount its metadata as a new catalog.

1. On the Catalog page, select **Manage Catalog**.
2. Click **+Add**.
3. On the **Add Catalog** page, select **Database Link** from Catalog Source.

The screenshot shows the 'Add Catalog' interface. At the top, a progress bar indicates four steps: 1. Catalog source, 2. Share provider settings, 3. Select shares, and 4. Catalog details. Below the progress bar, there are several catalog source options:

- Autonomous Database**: Add a catalog over another Autonomous Database in your tenancy. (Show more...)
- Database Link**: Add a catalog over another database to which you have a Database Link already configured. (This option is highlighted in blue.)
- Data Share**: Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.
- Amazon Glue**: Add a catalog over an Amazon Web Services Glue data catalog region.
- OCI Data Catalog**: Add a catalog over an OCI Data Catalog.
- Iceberg catalog**: Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

At the bottom right of the page, there are four buttons: Back, Next, Add, and Cancel.

4. Select the **Database Link** tab.

Add Catalog

1 Catalog source 2 Database Link

Database link name
TESTING_LINK

Catalog name
TESTING_LINK

Create Back Next Cancel

5. Select the **Database Link name** from the drop-down. For example, *AA-TESTING_LINK*.
6. Enter the name of the catalog in the **Catalog name** field. For example, *AA-TESTING_LINK*.
7. Click **Create**.
You can view the newly mounted catalog in the **Manage Catalogs** dialog box.

After you have successfully mounted a new catalog over a database connected using a Database link, you can select the catalog in Catalog Explorer and search for data in it.

Mount either Live Share or Delta Share Provider as Catalog

You can mount catalogs from registered data shares. This enables the consumers to access tables defined by other providers in other databases. The external tables need to be defined just once.

1. On the Catalog page, select **Manage Catalog**. Click **+Add**.
2. On the **Add Catalog** page, select **Data Share** from Catalog Source.

Add Catalog

1 Catalog source 2 Share provider settings 3 Select shares 4 Catalog details

Autonomous Database
Add a catalog over another Autonomous Database in your tenancy.
Show more...

Database Link
Add a catalog over another database to which you have a Database Link already configured.

Data Share
Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.

Amazon Glue
Add a catalog over an Amazon Web Services Glue data catalog region.

OCI Data Catalog
Add a catalog over an OCI Data Catalog.

Iceberg catalog
Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

Back Next Add Cancel

- Click **Next** to progress to the **Share provider settings** tab.
Under Share Source, select any of the available options:

- **Select from Live Share Providers**
- **Delta Share Provider JSON**

In this example, you will select share source from Live Share Providers.

Under **Share Provider Details**, specify the following details:

- **Provider Name:** Enter the name of the share provider. For example, *TEST-PROVIDER*.
- **Description:** Enter any description. This field is optional.

The screenshot shows the 'Add Catalog' wizard with four steps: 1. Catalog source, 2. Share provider settings, 3. Select shares, and 4. Catalog details. The current step is 'Share provider settings'. The 'Share Source' dropdown is set to 'LIVE SHARE'. Below it, the radio button for 'Select from Live Share Providers' is selected. The 'Share Provider Details' section includes a 'Provider Name' field with 'TEST-PROVIDER' and a 'Description' field with the text 'This describes the provide'. At the bottom right, the 'Next' button is highlighted with a red box.

- Click **Next** to progress to the **Select shares** tab.
- On the Share provider settings tab, drag and drop the share you want to mount as a catalog from the Available Shares to the Selected Shares.
- Click **Next** to progress to the Catalog details tab. On the Catalog details tab, specify the following details:
 - **Catalog name:** Enter a catalog name of your choice.
 - **Catalog description:** Enter a description.
- Select **Create** to mount a Live Share or Delta Share as a catalog.

After you have successfully mounted a new catalog over a Share, you can select the catalog in Catalog Explorer and search for data in it.

Mount Amazon Glue as Catalog

You can mount catalogs from registered AWS Glue data connections.

- On the Catalog page, select **Manage Catalog**.
- Click **+Add**.
- On the **Add Catalog** page, select **Amazon Glue** from Catalog Source.

The screenshot shows the 'Add Catalog' wizard with the following content:

- Step 1: Catalog source**
 - Autonomous Database: Add a catalog over another Autonomous Database in your tenancy. (Show more...)
 - Database Link: Add a catalog over another database to which you have a Database Link already configured.
 - Data Share: Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.
 - Amazon Glue**: Add a catalog over an Amazon Web Services Glue data catalog region. (This option is highlighted in blue.)
 - OCI Data Catalog: Add a catalog over an OCI Data Catalog.
 - Iceberg catalog: Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.
- Step 2: Amazon Glue Catalog**
- Step 3: Catalog details**

Buttons at the bottom right: Back, Next, Add, Cancel.

4. Click **Next** to proceed to the **Amazon Glue Catalog** tab.
5. On the **Amazon Glue Catalog** tab, specify the following details:
 - **Credential for Data Catalog Connection:** Select a credential from the drop-down.
 - **Region:** Select a region from the drop-down.
6. Click **Next** to progress to the **Catalog details** tab. Specify the following field values on the **Catalog details** tab:
 - **Catalog Name:** Enter the name that you want to use in Data Studio to refer to the catalog.
 - **Catalog Description:** Enter a description of the catalog.
7. Click **Create** to mount the AWS Glue Data Catalog region as a catalog.

After you have successfully mounted a new catalog over an AWS Glue Data Catalog region, you can select the catalog in Catalog Explorer and search for data in it.

Mount OCI Data Catalog as Catalog

You can mount catalogs from a registered OCI Data Catalog.

1. On the Catalog page, select **Manage Catalog**. Click **+Add**.
2. On the **Add Catalog** page, select **OCI Data Catalog** from Catalog Source.

Add Catalog

1 Catalog source 2 OCI Data Catalog 3 Catalog details

Autonomous Database
Add a catalog over another Autonomous Database in your tenancy.
[Show more...](#)

Database Link
Add a catalog over another database to which you have a Database Link already configured.

Data Share
Add a catalog over a set of data that has been shared with you, using either Live Sharing (from Autonomous Database), or Delta Sharing.

Amazon Glue
Add a catalog over an Amazon Web Services Glue data catalog region.

OCI Data Catalog
Add a catalog over an OCI Data Catalog.

Iceberg catalog
Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

Back Next Add Cancel

3. Click **Next** to proceed to the **OCI Data Catalog** tab.
4. On the OCI Data Catalog tab, specify the following field values:
 - **Credential for Data Catalog Connection:** Select a credential from the drop-down. This field is mandatory. For example, *BASICOCICRED*.
 - **Region:** Select a region from the drop-down. This field is mandatory. For example, *us-phoenix-1*.
 - **Data Catalog ID:** Select a Data Catalog ID from the drop-down. A Data Catalog ID is the ID of the account to which the data catalog belongs. For example, *catalog1*.

Select **Use separate credential for Object Storage** to use a different credential other than the selected one.

Add Catalog

1 Catalog source 2 OCI Data Catalog 3 Catalog details

Credential for Data Catalog Connection *
BASICOCICRED [Create Credential](#)

Region *
us-phoenix-1

Data Catalog ID
catalog1

Use separate credential for Object Storage

Select Credential *
AWSGLUE

Back Next Cancel

5. Click **Next** to progress to the **Catalog details** tab.
6. On the **Catalog details** tab, specify the following field values:
 - **Catalog Name:** Enter the name that you want to use in Data Studio to refer to the catalog. For example, *TEST_CATALOG*.
 - **Catalog Description:** Enter a description of the catalog.

The screenshot shows the 'Add Catalog' wizard with three steps: 1. Catalog source, 2. OCI Data Catalog, and 3. Catalog details. The current step is 'OCI Data Catalog'. The summary information is as follows:

Field	Value
Credential for Data Catalog Connection	BASICOCICRED
Region	us-phoenix-1
Catalog ID	[Redacted]
Bucket credentials	[Redacted]
Catalog name	TEST-CATALOG
Catalog description	This is Catalog Description

At the bottom right, the 'Create' button is highlighted with a red box.

7. Click **Create** to mount the OCI Data Catalog as catalog.

After you have successfully mounted a new catalog over an OCI Data Catalog, you can select the catalog in Catalog Explorer and search for data in it.

Mount a Catalog over an Iceberg Data Catalog

You can add a catalog over an Iceberg Data Catalog. Integrating a catalog with Iceberg data involves connecting with supported catalogs such as Unity and Polaris. To integrate Iceberg Data Catalogs via Data Studio, you need to ensure your Iceberg Catalogs are properly configured and accessible.

To configure and access your Iceberg Catalog, you must have the following:

1. **Iceberg Endpoint:** Set up Polaris or Unity Catalog using the Iceberg REST API. This involves specifying the catalog type, URI, and warehouse location. You will need the REST endpoint for your external Iceberg catalog. This is the URL that provides access to the Iceberg metadata.
Example: *https://your-iceberg-catalog.com/api/v1/iceberg*
2. **Token Endpoint:** The */v1/oauth/tokens* endpoint is provided by Iceberg REST catalogs to issue **Bearer Token**.
3. **Client ID & Client Secret:** These are credentials provided by the OAuth server to identify your application (in this case, the Unity and Polaris Catalog integration). You will use these credentials to generate Bearer Token.
4. **Bearer Token:** To authenticate requests to the Iceberg REST endpoint, you will use OAuth with the Client Credentials. The Catalog tool calls the token endpoint to receive **Bearer Token**.

5. **Namespace:** A namespace is equivalent to a schema in a database. This location will serve as the repository for your Iceberg tables within the catalog. Iceberg catalogs group tables into namespaces, which can be nested within other namespaces. The Iceberg specification supports arbitrary levels of nesting. For example, a SALES table could be defined within a STORES namespace, which is defined in a RETAIL namespace, which is itself defined within a CORPORATE namespace. The Iceberg location of such a table would be CORPORATE.RETAIL.STORES.SALES.

The following is a sample configuration to create a REST Catalog integration that uses OAuth to connect to Apache Iceberg tables:

```
CREATE OR REPLACE CATALOG INTEGRATION tabular_catalog_int
  CATALOG_SOURCE = ICEBERG_REST
  TABLE_FORMAT = ICEBERG
  CATALOG_NAMESPACE = 'default'
  REST_CONFIG = (
    CATALOG_URI = 'https://api.tabular.io/ws'
    CATALOG_NAME = '<tabular_warehouse_name>'
  )
  REST_AUTHENTICATION = (
    TYPE = OAUTH
    OAUTH_TOKEN_URI = 'https://api.tabular.io/ws/v1/oauth/tokens'
    OAUTH_CLIENT_ID = '<oauth_client_id>'
    OAUTH_CLIENT_SECRET = '<oauth_secret>'
    OAUTH_ALLOWED_SCOPES = ('catalog')
  )
  ENABLED = TRUE;
```

Here, CATALOG_URI is the REST end point to access the warehouse.

The REST_AUTHENTICATION procedure consists of all the parameters you need to create the BEARER TOKEN and use it to create the credential.

See [Configure a catalog integration for Apache Iceberg™ REST catalogs](#) for more information on this.

To mount a Polaris Catalog over an Iceberg Data Catalog:

1. On the Catalog page, select **Manage Catalog**. Click **+Add**.
2. On the **Add Catalog** dialog, select **Iceberg catalog** from the **Catalog source** tab.

Add Catalog

1 Catalog source 2 Iceberg catalog 3 Select namespace

Database Link
Add a catalog over another database to which you have a Database Link already configured.

Data Share
Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.

Amazon Glue
Add a catalog over an Amazon Web Services Glue data catalog region.

OCI Data Catalog
Add a catalog over an OCI Data Catalog.

Iceberg catalog
Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

Back Next Add Cancel

Click **Next**.

3. Specify the following field values on the Catalog source tab of the Add Catalog dialog:

- **Catalog name:** Enter the name of the Catalog you want to mount the Iceberg Catalog as.
For example, *Polaris*.
- **Iceberg catalog type:** Select *Polaris* from the drop-down field.
- **Iceberg catalog endpoint:** It specifies your Polaris Catalog account URL. The endpoint for Snowflake's Polaris Catalog is usually in the format of `https://<account_locator>.snowflakecomputing.com/polaris/api/catalog`. For more information about account identifier formats, see the [Snowflake documentation](#).
- **Iceberg catalog credential:** Select a credential from the list of available credentials in the drop-down list. If you do not have a credential, you can create one. Select **Create Credential**. On the Create Credential dialog, specify the following field values:
 - **Credential Name:** Enter the name of your credential. For example, *POLARIS_CRED*.
Under **Get New Bearer Token**,
 - * **Access token end point:** Enter the Iceberg REST specification endpoint. This generates access token using OAuth2 defined as `/v1/oauth/tokens`. This endpoint was designed to facilitate authentication by exchanging client credentials (client ID and secret) for an access token.

Note

The `/v1/oauth/tokens` endpoint is deprecated in Iceberg version 1.6.0 and will be removed in version 2.0. You are encouraged to rely on external OAuth2 providers instead. See [OAuth Authentication](#) for more details.

- * **Grant type:** Select *Client credentials*.
- * **Client ID:** Specify the Client ID.
- * **Client Secret:** Specify the Client Secret.

- * **Scope:** *PRINCIPAL_ROLE:ALL*.
Click **Get Token**.

The **Bearer token** field gets populated.

Click **Create Credential**.

POLARIS_CRED is displayed as Iceberg catalog credential.

Create Credential

Credential Name *

Get New Bearer Token
Access token end point

Grant type

Client ID

Client secret

Scope

Note

The credentials are stored using the procedure `DBMS_CLOUD.CREATE_CREDENTIAL` in the backend.

For the sample example provided above, the procedure would be as follows:

```
BEGIN
  dbms_cloud.create_credential(
    credential_name => 'POLARIS_CRED',
    username       => 'BEARER_TOKEN',
    password       => '<BEARER_TOKEN_VALUE This is
generated using token endpoint, client id, client secret>'
  );
END;
```

Add Catalog

1
2
3

Catalog source
Iceberg catalog
Select namespace

Catalog name *

Iceberg catalog type *

Iceberg catalog endpoint *

Iceberg catalog credential *

Bucket credentials

This credential allows the Oracle Autonomous Database to access the Iceberg table.

- **Bucket credentials:** This credential stores a cloud storage volume to store your Iceberg tables in Parquet format. This field is mandatory for Polaris catalogs. Click **Next** to proceed to the **Iceberg catalog** tab.
4. On the **Iceberg catalog** tab, you can specify the following field values:
- **Iceberg catalog namespaces:** This field displays the hierarchical organization of tables. The **Iceberg catalog endpoint** consists of the root namespace. Select the namespace on which you want to have the catalog mounted.

- **Namespace separator style:** Select **Polaris style**. This is the default value. Select any value based on the way your catalog is configured.

Note

It is preferred that you retain the default value.

- **Bucket region:** Enter the bucket region. For example, *us-west-2*. Iceberg catalog endpoints, serve as interfaces for managing Iceberg tables stored in cloud storage like S3. The bucket region refers to the geographical location where the S3 bucket is hosted.

This is mandatory field for Polaris catalogs.

Add Catalog

1 **Catalog source** 2 **Iceberg catalog** 3 **Select namespace**

Catalog name: POLARIS_CATALOG

Iceberg catalog type: Polaris

Iceberg catalog endpoint: https://e [] [] [] -catalog

Iceberg catalog credential: POLARIS_CRED

Iceberg catalog namespaces *

- SPAPADOM_ICEBERG
 - PUBLIC**

Advanced

Namespace separator style

Databricks style (.)

Polaris style (%1F)

Other

Bucket region

Back Next **Add** Cancel

Click **Add**.

After you have successfully mounted a new catalog over an Iceberg Catalog, you can view the Polaris Catalog on the **Manage Catalogs** page. You can disable and delete the catalog from this page.

Manage Catalogs

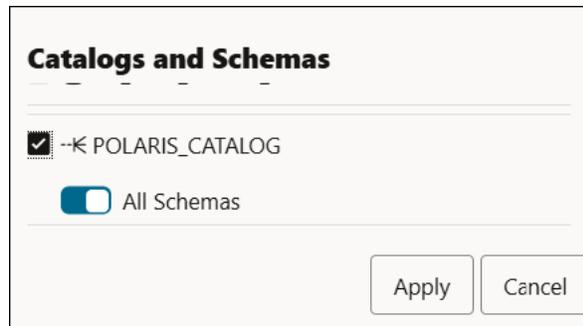
← POLARIS_CATALOG ⋮

Close

View the mounted Polaris Catalog on the Catalog Explorer

Once you have mounted the new catalog, you can load tables from the mounted catalog. To search for the catalog:

1. On the Catalog tool, click **Select Catalogs...**
2. Click **POLARIS_CATALOG** filter on **Catalogs and Schemas** wizard to display the **POLARIS_CATALOG** catalogs.
3. Click **Apply** to view the results on the Catalog page.



To mount a Unity Catalog over an Iceberg Data Catalog:

1. On the Catalog page, select **Manage Catalog**. Click **+Add**.
2. On the Add Catalog dialog, select **Iceberg Catalog**.

Add Catalog

1 Catalog source 2 Iceberg catalog 3 Select namespace

Database Link
Add a catalog over another database to which you have a Database Link already configured.

Data Share
Add a catalog over a set of data that has been shared with you, using either Live Sharing (from another Autonomous Database), or Delta Sharing.

Amazon Glue
Add a catalog over an Amazon Web Services Glue data catalog region.

OCI Data Catalog
Add a catalog over an OCI Data Catalog.

Iceberg catalog
Add a catalog over an Iceberg data catalog. Unity and Polaris catalogs are supported.

Back Next Add Cancel

3. Click **Next**.

4. Specify the following field values on the Catalog source tab of the Add Catalog dialog:

- **Catalog name:** Enter the name of the Catalog you want to mount the Iceberg Catalog as.
For example, *Unity*.
- **Iceberg catalog type:** Select *Unity* from the drop-down field.
- **Iceberg catalog endpoint:**
For Unity Catalog, you typically use the Databricks REST API. The endpoint for accessing Iceberg tables via Unity Catalog is usually in the format of `https://<databricks-instance-url>/api/2.1/unity-catalog/iceberg`.

Note

See <https://docs.databricks.com/aws/en/external-access/iceberg> for more information.

- **Iceberg catalog credential:** Select a credential from the list of available credentials in the drop-down list. If you do not have a credential, you can create one. Select **Create Credential**.
On the Create Credential dialog, specify the following values:
 - **Credential Name:** Enter the name of your credential. For example, *UNITY_CRED*.
 - **Bearer Token:** Paste the Bearer Token value.
Select **Create Credential**.
UNITY_CRED is displayed as **Iceberg catalog credential**.

Add Catalog

1
2
3

Catalog source
Iceberg catalog
Select namespace

Catalog name *

Iceberg catalog type *

Iceberg catalog endpoint *

Iceberg catalog credential *
 Create Credential

Bucket credentials
 Create Credential

Back
Next
Add
Cancel

Click **Next** to proceed to **Iceberg catalog** tab.

On the **Iceberg catalog** tab, you can specify the following field values:

- **Iceberg catalog namespaces:** Select any one of the schemas you need access to.
- **Namespace separator style:** The namespace separator style typically refers to how namespaces are structured and separated within the catalog. By default, **Databricks style** is selected.
- **Bucket region:** This field is not mandatory.

Add Catalog

1
2
3

Catalog source
Iceberg catalog
Select namespace

Catalog name UNITY_CATALOG

Iceberg catalog type Unity

Iceberg catalog endpoint https://[redacted]rg/v1

Iceberg catalog credential [redacted]_UNITY_CRED

Iceberg catalog namespaces *

bigdata_qa

main

samples

system

Advanced

Namespace separator style

Databricks style (.)

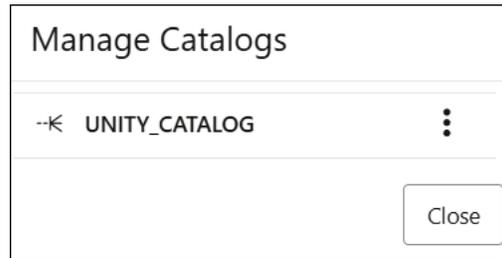
Polaris style (%1F)

Other

Bucket region

Back
Next
Add
Cancel

Click **Add**.

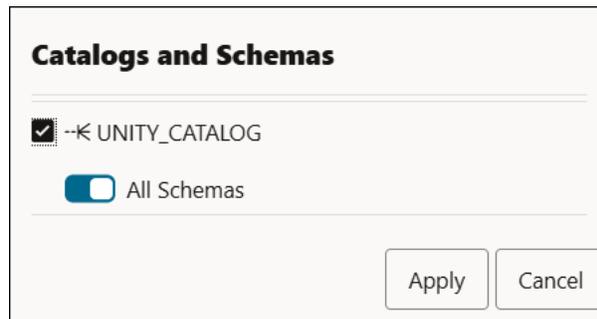


After you have successfully mounted a new catalog over an Iceberg Catalog, you can view the Unity Catalog on the **Manage Catalogs** page. You can disable and delete the catalog from this page.

View the mounted Unity Catalog on the Catalog tool

Once you have mounted the new catalog, you can load tables from the mounted catalog. To search for the catalog:

1. On the Catalog tool, click **Select Catalogs...**
2. Click **UNITY_CATALOG** filter on the **Catalogs and Schemas** wizard to display the **UNITY_CATALOG** catalogs.



3. Click **Apply** to view the results on the Catalog page.



- [Inline Data Load of Delta Share from the Catalog Tool](#)
You can load a Delta Share from the Catalog tool without leaving the Catalog tool for loading.
- [Query Remote Catalogs and Databases](#)
You can access and query external data by mounting catalogs. This feature uses a simple SQL syntax that is similar to how you query data using database links.

Inline Data Load of Delta Share from the Catalog Tool

You can load a Delta Share from the Catalog tool without leaving the Catalog tool for loading.

You need to first download your share profile to receive shares.

1. Download your share profile:

For the data share you want to receive, download your profile sent from the share provider.

- Open the invitation email you received from the data share provider, when the data share was published.
- Copy the profile URL from the email.
- Paste the URL into your browser and click **Get Profile Information**.
- Your profile has been downloaded to your local machine. The Profile information downloaded message page is displayed.

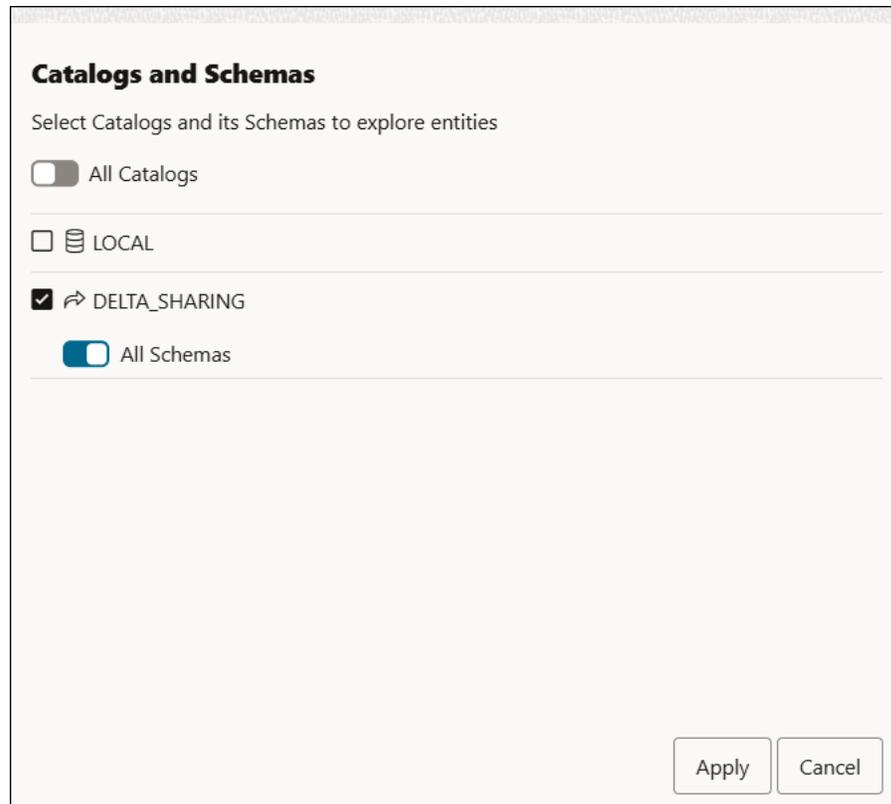
You can subscribe to the data share provider using the Catalog tool.

To subscribe, you need to use the information contained in the uploaded JSON profile you received from the share provider.

From the **Manage Catalogs** feature of the Catalogs tool, you can upload the JSON profile and follow the Add Catalog wizard.

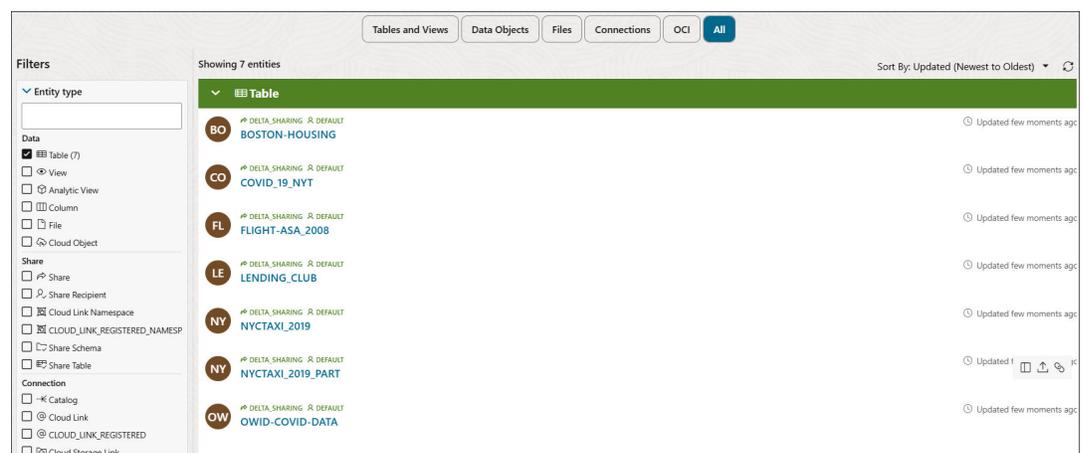
1. On the Data Studio menu of Database Actions, select the Catalog tool.
2. Select **Manage Catalogs**.
3. On the **Manage Catalogs** dialog, select **+Add**.
4. Mount Delta Share as catalog:
Select **Data Share** as a Catalog Source on the Add Catalog dialog and click **Next**.
5. Select **Delta Share Provider JSON**, accept the **From File** default option, and click the **Delta Share Profile JSON** field.
 - Navigate to and select the **JSON profile** you downloaded in a previous step. Click **OPEN**.
 - Enter the **Provider Name**, under Share Provider Details and click **Next**.
6. Register shares made available to you.
Move shares from **Available Shares** to **Selected Shares** and click **Next**.
7. Specify details of the Catalog:
Enter the name of the Catalog and click **Create**.

The Catalog page is displayed with a success message confirming that the Mounting Catalog for Shares was successful and that it has created the share provider.
8. View the newly created Delta share by filtering the results using the Select Catalogs... icon:
 - Click **Select Catalogs...**
 - Select **Delta Sharing** from the **Catalogs and Schemas** wizard.

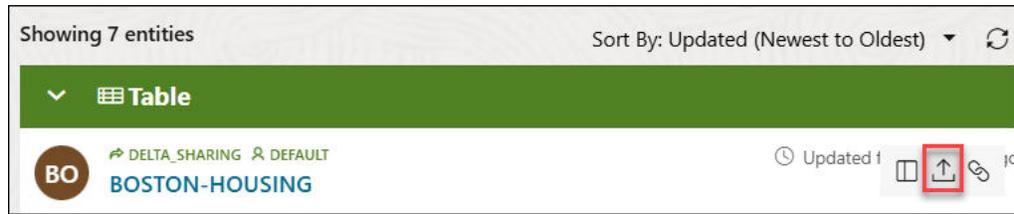


- Click **Apply** to apply the filter criteria in the search results.

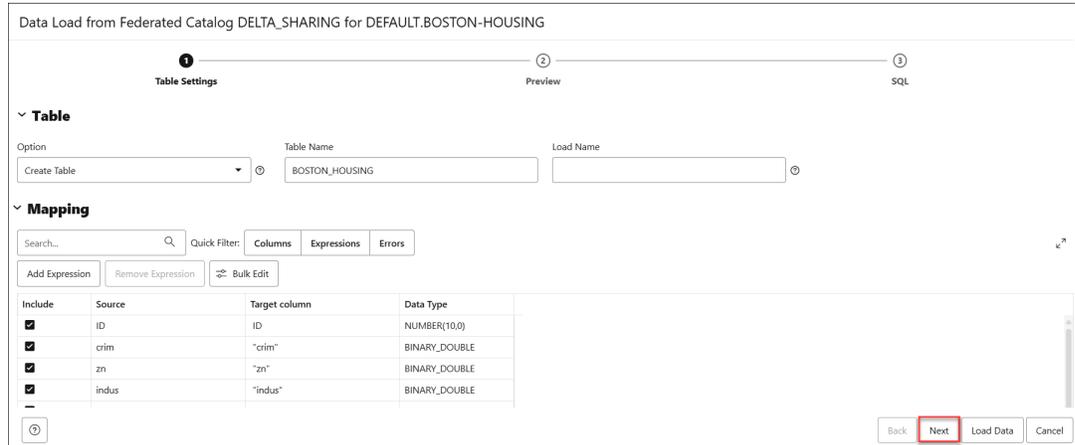
You will view all the tables listed from the Delta Share.



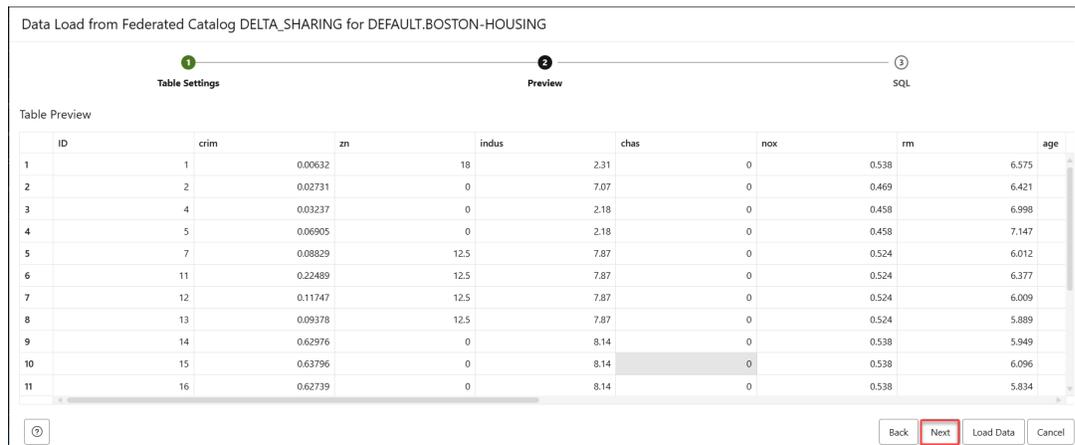
9. Select the table you want to load data from, then click **Load to table**.



- On the **Table Settings** of the Data Load dialog, you can specify processing options to specify settings for the data load job. You can also rename the newly created table from the catalog.



- Click **Next** to view the data in the table.



- Click **Next** to view the **Preview** tab to display the equivalent SQL code of the creating table job.



13. Select **Load Data**.

A confirmation message is displayed that the data load job has started.

14. Navigate to the Data Load home page to view the progress of the data load job.



You have loaded data from Delta Share from the Catalog tool.

Query Remote Catalogs and Databases

You can access and query external data by mounting catalogs. This feature uses a simple SQL syntax that is similar to how you query data using database links.

The simplified SQL syntax makes it easy to discover and query external data without having to choose which external data to access, and without having to construct external tables or perform any other data integration tasks.

Refer to the [Query External Data](#) chapter for more information.

By querying remote catalogs and databases, you are accessing data directly from the external catalog in place, without having to move data into the Autonomous AI Database. This feature enables you to export data from the catalogs that the Autonomous AI Database can access without establishing multiple connections. You can query the Autonomous AI Database to see which external catalogs are available, what data is available for any given catalog, and you can query the data directly, from any SQL client. This involves providing a standard and simplified way to discover and query data sets connected to the Autonomous AI Database, leveraging the catalog's ability to present external data as standard Oracle schemas and tables.

Refer to the [Manage Catalogs](#) chapter for information on catalogs supported across various categories, including Databases, External Data Catalogs, Shares, and more.

Some of the key features are:

- **Discover External Data:** Using SQL, you can identify external catalogs along with the tables they contain. This will help you in understanding the available data structure.
- **Simplify Table Discovery:** Use existing query syntax and familiar views to easily list tables within an external catalog.
- **Integrate with data tools:** You can integrate the query data with Data Studio suite of tools to make informed decisions and gain deeper insights.

Topics:

The following topics explain the Query Remote Catalogs and Databases feature:

- [Conceptual Diagram of Querying Remote Catalogs](#)
- [Concepts related to querying remote catalogs and databases](#)
- [About Querying Remote Catalogs and Databases](#)
- [Examples: List the Mounted Catalogs](#)
- [Examples: List the Tables in a Catalog, Including Their Owning Schemas](#)
- [Examples: Query a Table from a Delta Share and Analyze the View](#)

- [Conceptual Diagram of Querying Remote Catalogs](#)

The following diagram displays the conceptual network diagram of querying remote catalogs using Autonomous Database Serverless:

- [Concepts related to querying remote catalogs and databases](#)

An understanding of the following concepts is necessary for querying with Remote Catalogs:

- [About Querying Remote Catalogs and Databases](#)

In this chapter, you will learn to query external data using the catalog metadata using an SQL syntax of the form: `[schema].object@catalog_name`.

- [Examples: List the Mounted Catalogs](#)

You will use this syntax in a SQL query.

- [Examples: List the Tables in a Catalog, Including Their Owning Schemas](#)

To list all tables within a specific schema (catalog) from the enabled catalogs, you can use:

- [Examples: Query Data from a Table in a Catalog](#)

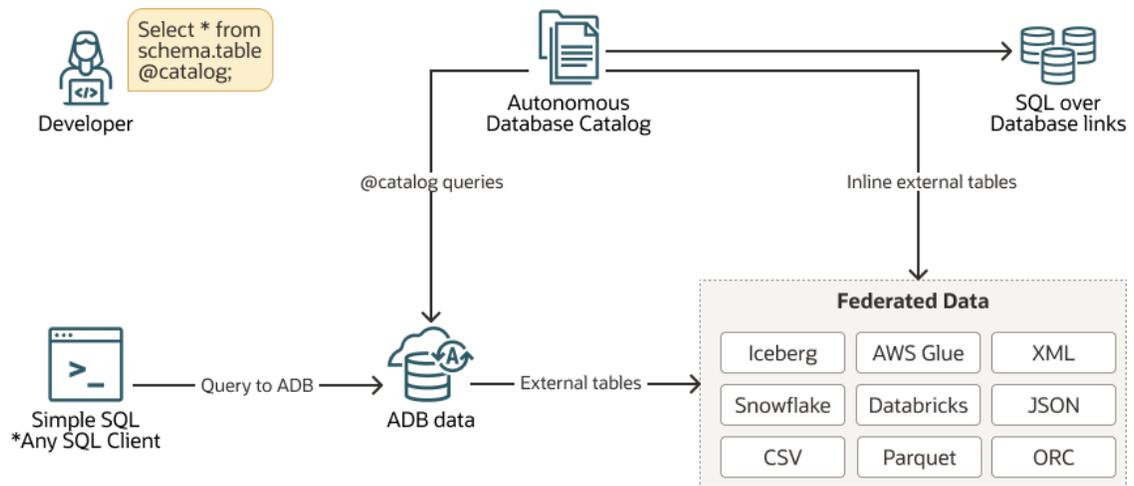
To query data from a table in a specific catalog (schema), you must prepend the schema name to the table name.

- [Examples: Query a Table from a Delta Share and Analyze the View](#)

Let's assume that we are working on an inventory database where we have data about the information collected by the U.S. Census Service concerning housing in the Boston area, and we need a query that returns the maximum value of the full-value property tax.

Conceptual Diagram of Querying Remote Catalogs

The following diagram displays the conceptual network diagram of querying remote catalogs using Autonomous Database Serverless:



Data from all types of external catalog can be queried this way including:

- Database Links:** Data in other databases where a catalog is mounted over a database link.

These catalogs are built using traditional database links, which are set up in advance by system administrators. The catalog tool autonomous database uses standard SQL queries on these links to discover tables and other objects.

There are two types of database links: Oracle and Heterogeneous.

For Oracle links, you will use queries like "SELECT table_name FROM sys.all_tables@link".

For Heterogeneous links, similar queries are run without the SYS term, such as "SELECT table_name FROM all_tables@link".
- Glue Catalogs:** Data made available via AWS Glue, where a catalog is mounted over an AWS Glue region.
- OCI Data Catalogs:** Data made available via OCI Data Catalogs. These catalogs use customer-defined credential objects to securely connect to OCI endpoints protected by ACLs.
- Delta Share Catalogs:** Data that has been shared using the Delta Sharing standard from any system that can use this standard to share data, where a catalog is mounted over the share. These catalogs leverage External Table Delta Sharing support, which employs customer-defined credential objects to access delta sharing endpoints. The credential includes a bearer token for authentication.
- Iceberg Catalogs:** Data made available via an Iceberg Data Catalog, such as Databricks Unity Iceberg Catalog, or Snowflake Polaris.

Concepts related to querying remote catalogs and databases

An understanding of the following concepts is necessary for querying with Remote Catalogs:

- Catalog:** A catalog is a metadata layer that organizes and describes data assets such as schemas and tables. It acts as a bridge between the database and external data sources. A catalog contains one or more schemas, and each schema can have multiple tables. These schemas and tables represent the logical organization of data.

- **External Catalogs:** they are catalogs that stores metadata for systems other than the locally connected ADB, including other ADBs, remote databases, shares, external data catalogs.
- **DB Links:** DB Links are a mechanism to connect to a remote database from an Oracle database. They allow you to access tables, views, and other objects in a remote database as if they were local objects. They enable querying data in remote databases. This feature aims to treat DB Links as a type of "catalog." The data dictionary of a remote database connected via a DB Link can be considered a catalog.
- **Mounting a Catalog:** A **mounted catalog** is one where the Autonomous Database has an active connection to the catalog, allowing it to access and query its datasets. Oracle Autonomous Database supports mounting catalogs from external data sources like AWS Glue or Databricks.

About Querying Remote Catalogs and Databases

In this chapter, you will learn to query external data using the catalog metadata using an SQL syntax of the form: `[schema]. object@catalog_name`.

Here the syntax `[schema]. object@catalog_name` refers to a representation of accessing database objects

The table describes the parameters and their descriptions used in the syntax:

Table 13-2 Parameters and Descriptions used in the syntax

Parameters	Description
Schema	This field is optional, and it defaults to the current user. A schema is a namespace within a catalog that organizes related database objects such as tables, views, procedures, or functions. It acts as a logical container for these objects.
Object	This field refers to any database object contained within a schema, such as tables, views, procedures, or functions.
Catalog	The catalog field displays the name of the catalog with the referenced table. You can create and manage catalogs using the <code>DBMS_CATALOG</code> package or the Catalog Tool of the Data Studio suite of tools.

Note

You can replace catalogs with **dbLinks** as well.

Accessing Remote Tables

To access a table from a remote schema using a catalog, you must follow these steps:

1. **Mount a Catalog:** Begin by mounting a catalog over any supported type of data store (e.g., Iceberg, Delta Lake, etc.). Mounting a catalog standardizes the metadata, enabling SQL access to external data as if it were tables within a schema in the catalog. Refer to [Manage Catalogs](#) for detailed steps on mounting catalogs. Once the catalog is mounted and metadata is standardized, you can directly query external data using SQL. Even if the

external data does not physically exist as traditional tables, the catalog abstracts it into a tabular format for easy querying.

- 2. Reference the table:** Once mounted, you can reference the remote table by specifying the schema and catalog (if applicable) followed by the table name using the following syntax:
`[remote_schema].table@catalog.`

Here the `schema` in this identifier is the remote schema that contains the remote table. `Table`, in this case, is the table name in the remote catalog.

Note

The user must have the appropriate credentials to access the specific table and catalog.

Key Points to Consider

- The feature is designed using the `DBMS_CATALOG` package installed in an ADB-S environment.
- DDL operations (e.g., `CREATE`, `ALTER`, `DROP`) are not allowed on remote databases via catalogs. If you do so, you may encounter an `"ORA-02021: DDL operations are not allowed on a remote database"` error.
- Same name Restriction:** This feature enforces that for a given user, there cannot exist a DB link and a catalog with the same name.

In the subsequent chapters, you will learn how to:

- [Display the list of Mounted Catalogs](#)
- [Display the list of tables in a Catalog, Including Their Owing Schemas](#)
- [Query data from a Table in a Catalog](#)
- [Query data using DBLinks](#)

You will use SQL Worksheet Editor of Database Actions to run SQL queries. You must first connect to your Database Actions instance and select **SQL** under **Development** menu.

Prerequisites:

You will require an OCI account.

How are DBlinks different from Catalogs?

DBLinks: They are primarily used to connect and query another database directly using a syntax.

For example:

```
SELECT *FROM <schema_name>.<table_name>@<dblink_name>;
```

When a catalog is mounted over another database via a DBLink, there is no need to separately define metadata in the catalog because the DBLink syntax directly accesses the remote database objects. Essentially, the DBLink acts as a transparent connection to the remote database, allowing queries to run as if the data were local.

Catalogs: They serve as a metadata layer that can be mounted over different types of data stores, not only databases connected via DBLinks but also external data catalogs or data shares like Delta Share or Live Share providers. When a catalog is mounted over a non-database store, the `@catalog` syntax adapts queries to generate inline external tables, enabling access to external data without requiring a DBLink. This means catalogs abstract the metadata management and provide a unified interface for querying diverse data sources, including external ones.

Examples: List the Mounted Catalogs

You will use this syntax in a SQL query.

For example, to list the catalogs you mount, you could run the following command:

```
SELECT CATALOG_NAME, CATALOG_TYPE, IS_ENABLED FROM USER_MOUNTED_CATALOGS;
```

The output of the query lists the catalogs that you connect with.

After the connection, you can query the catalogs using the syntax mentioned above:

```
SELECT * FROM "DEFAULT"."BOSTON-HOUSING"@CATALOGNAME;
```

In the above example, *Catalogname* is a catalog you have mounted, and *BOSTON-HOUSING* is a table in that catalog.

Examples: List the Tables in a Catalog, Including Their Owning Schemas

To list all tables within a specific schema (catalog) from the enabled catalogs, you can use:

```
SELECT OWNER, TABLE_NAME FROM ALL_TABLES@CATALOGNAME;  
WHERE OWNER = '<SCHEMA_NAME>';
```

Here, *ALL_TABLES* is the table name.

For example, to list tables in the *SH* schema (a sample schema available in Oracle Autonomous Database):

```
SELECT OWNER, TABLE_NAME FROM ALL_TABLES@CATALOGNAME;  
WHERE OWNER = 'SH';
```

This query returns all tables owned by the specified schema.

Note

In this example, you are listing tables in the *SH* schema of an external database.

Examples: Query Data from a Table in a Catalog

To query data from a table in a specific catalog (schema), you must prepend the schema name to the table name.

For example, querying data from the `SALES` table in the `SH` schema:

```
SELECT * FROM
  "SH"."SALES"@catalogname;
```

Note

In this example, you are querying data in the `SH` schema of an external database.

If you are working with other schemas like `SSB`, you can similarly query their tables by specifying the schema name:

```
SELECT * FROM
  "SSB"."SALES"@catalogname;
```

Note

In this example, you are querying data in the `SSB` schema of an external database.

Examples: Query a Table from a Delta Share and Analyze the View

Let's assume that we are working on an inventory database where we have data about the information collected by the U.S. Census Service concerning housing in the Boston area, and we need a query that returns the maximum value of the full-value property tax.

1. Create an Oracle Delta Share provider and credential:

For creating a Delta Share provider, you typically need to interact with the Delta Sharing platform, which involves creating a credential suitable for use with delta share providers. In this example, you can create `DATABRICKS_PROVIDER`.

```
BEGIN
  dbms_share.create_or_replace_share_provider(
    provider_name=> 'DATABRICKS_PROVIDER',
    endpoint=>      'https://sharing.delta.io/delta-sharing');
  dbms_cloud.create_credential(
    credential_name => 'CREDENTIAL_NAME',
    username => 'BEARER_TOKEN',
    password => '*****');
  dbms_share.set_share_provider_credential(
    'DATABRICKS_PROVIDER', 'CREDENTIAL_NAME');
END;
//
```

The output of this procedure successfully creates a delta share provider.

2. Mount a DB Share Catalog:

After creating credentials, you can mount the registered share provider from step 1 as a catalog to enable the catalog users to search and discover data from the share, and to query that data using SQL.

```
BEGIN
  dbms_catalog.mount_share(
    catalog_name=>'databricks',
    share_provider=>'databricks_provider',
    share_name=>'delta_sharing');
END;
/
```

The output of this procedure successfully creates a catalog integration that enables the Catalog tool to query tables stored in Databricks.

3. Run a simple query:

You can now run a simple query on the table:

```
select max(tax) from "DEFAULT"."BOSTON-HOUSING"@catalogname;
```

The output returns the maximum value of the full value of the property tax rate present in the table.

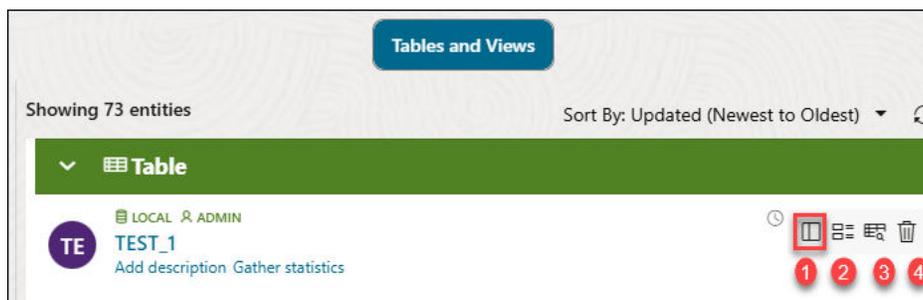
```
MAX(TAX)
-----
711
```

Viewing Entity Details

When you have found an entity you are interested in, you can click the entity to view its details, or click one of the other available actions from the menu that appears on the right hand side when you hover over that row.

The way you select the entity details can be set using the **Customize Themes** tab in the [Catalog Settings](#) menu.

To view details about an entity, click the **Actions** icon at the top right of the Table and View Loads header entry, then click **Table** and select **View Details**.



For all entities, the details include Lineage and Impact sections. The inclusion of other details, such as Preview and Statistics, varies by entity type.

Preview

Preview displays the data of the entity. For a table, the Preview displays the columns of the table and the data in those columns. You can sort the data in the column into ascending or descending order by clicking the up or down arrow to the right of the column name.

Describe

For an analytic view, the Describe tab has information about the entity, and displays the hierarchies, levels, level depth, dimension tables, level columns, and number of distinct values for a level.

Lineage

Lineage displays all known information about the upstream dependencies of the entity, and therefore how the entity was created and how it is linked to other entities.

For example, for a table that you created in your database, the lineage is just the table. For a table that you created by loading a CSV file from cloud storage, the lineage includes the ingest directive for the data load and the CSV file that is the source of the data.

Pointing to the name of an item in the lineage displays the table name, the application that created it, the type of entity, the path to it, and the schema it is in.

Arrows point from an entity to the entity that it derives from. For example, for a table created in a data load job, an arrow points from the table to the ingest job and an arrow points from the ingest job to the CSV file. If you point to an arrow, then a Links Information box appears that shows information about the relationship between the two entities.

To view more details about an item, click the Actions icon for the item, then click **Expand**. For a table, the columns of the table are displayed. Pointing to the name of the table or of a column displays the name, application, type, path, and schema of the table or column. To collapse the display, click the Actions icon, then click **Collapse**.

You can increase or decrease the size of the displayed objects by using the + (plus) and - (minus) keys. You can reposition the objects by grabbing a blank spot in the display and dragging vertically or horizontally.

The lineage for some entities, such as analytic views, is more complex. An analytic view entity deployed for a business model has links to columns in a fact table and to hierarchies. The fact table has links to attribute dimensions and, for a data load job, to an ingest directive for the job. The ingest directive has a link to the source file. The attribute dimensions have links to tables for the dimensions. Those tables have links to ingest directives that have link to source files.

Impact

Impact shows all known information about the downstream use of an entity, and therefore how a change in the definition of an entity may affect other entities that depend on it. For example, if a table is used in an analytic view, a change to one of the column definitions in the table may affect the mapping from that column to the analytic view.

Classifications

For an analytic view and its attribute dimensions, hierarchies, and measures, the Classifications tab displays classifications and their values. Classifications are metadata that applications can use to present information about analytic views. When creating a business model, you may specify the values for the Caption and Description classifications.

Optimize

For an analytic view, the Optimize tab has information about caches created for the analytic view. A cache may exist if the advanced option Enable Autonomous Aggregate Cache was selected for the business model for which the analytic view is deployed.

Statistics

Statistics display information about the entity. For example, the statistics for a table include the size of the table and the numbers of rows and columns. They also include the names of the columns, their data types, the number of distinct values and null values, the maximum and minimum values, and other information.

The data is represented in the form of histogram which is column statistic which provides more detailed information about data distribution in a table's columns.

The histograms in the statistics pane can be representative of the following types:

- **Frequency:** In a frequency histogram, each distinct column value corresponds to a single bucket of the histogram. Since each value has its own dedicated bucket, some buckets may have many values, whereas others have few.
- **Top-frequency:** A top frequency histogram is a variation on a frequency histogram that ignores non-popular values that are statistically insignificant
- **Height-Balanced:** In this histogram, column values are divided into buckets so that each bucket contains approximately the same number of rows.
- **Hybrid:** A hybrid histogram combines characteristics of both height-based histograms and frequency histograms. This approach enables the optimizer to obtain better selectivity estimates in some situations.

Refer to [Gathering Statistics for Tables](#) section for more details.

Job Report

Displays a report of the total rows loaded and failed for a specific table.

The report displays of the total rows loaded and failed.

You can view the name of the table, the time the table was loaded and the time taken to process the load.

Data Definition

Data Definition displays the Oracle Autonomous Database DDL that created the entity.

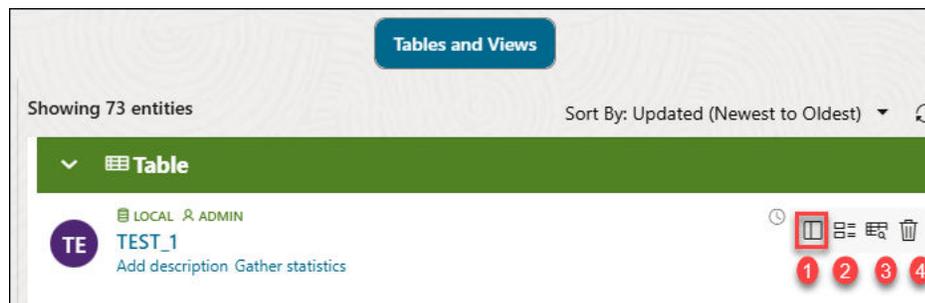
You can view the following **Entity Types** from the drop-down:

- All
- Table
- View
- Analytic View
- Table,View,Analytic View
- Analytic Dashboard
- Attribute (Attribute Dimension)
- Attribute (Hierarchy)

- Attribute Dimension
- Cloud Link (Cloud Link Namespace)
- Cloud Link Namespace
- Cloud Object (Cloud Storage Link)
- Cloud Object (Data Catalog Asset)
- Cloud Object (Share Link)
- Cloud Storage Link
- Cloud Virtual Object (Cloud Storage Link)
- Column (Table)
- Column (View)
- Data Catalog Asset
- Data Catalog Link
- Database Link
- Directory
- External Location
- File (Directory)
- Function
- Hierarchy
- Ingest Job
- Level (Attribute Dimension)
- Level (Hierarchy)
- Live Table Feed
- Measure (Analytic View)
- Mining Model
- Package
- Procedure
- Schema
- Share
- Share Link
- Share Provider
- Share Recipient
- Share Schema (Share)
- Share Table (Share Schema)

Table Entity Details

1. Click the **View Details** icon next to **Table** entity to perform the following operations:



- Select the **Preview** tab to view details about the table.
From the **Actions** drop-down list, you can select any of the following operations.
 - Select the **Gather Statistics** tab to display new statistics after you modify the table's structure. Refer to the [Gathering Statistics for Tables](#) for more detailed information. You can also view statistics from the **Statistics** tab while viewing the table details in the Table Entity wizard.
 - Select **Add to Share** to share the table. Refer to [Provide Share](#) for detailed information. You can also share the selected table using the **Actions** drop-down field while viewing the table details in the Table Entity wizard.
 - Select **Register to Cloud Link** to register the table for remote access for a selected audience you define. See [Registering Cloud Links to Access Data](#) section for more information.
 - Select **Create Analytic View** to create an Analytic View from the selected associated table. See [Creating Analytic Views](#).
 - Select [Export Data to Cloud](#) to export data to a cloud object store.
 - Select **Edit** to edit the properties of the table. Refer to [Editing Tables](#) for more information.
- 2. Select **AI Assist** to enhance this data table with the power of AI.
- 3. Select **Query** to analyze the selected table in the Data Analysis tool. Refer to [The Data Analysis Tool](#) for more information.
- 4. Select **Drop** to delete the table.

View Entity Details

1. Click the **View Details** icon next to **View** entity to perform the following operations:
 - Select the **Preview** tab to view details about the view.
 - From the **Actions** drop-down list, you can select any of the following operations.
 - Select **Add to Share** to share the table. Refer to [Provide Share](#) for detailed information.
 - Select **Create Analytic View** to create an Analytic View from the selected associated view. See [Creating Analytic Views](#).
 - Select [Export Data to Cloud](#) to export data to a cloud object store.
 - Select **Query** to analyze the selected view in the Data Analysis tool. Refer to [The Data Analysis Tool](#) for more information.
2. Select **Drop** to delete the view.

Share Entity Details

1. Click the **View Details** icon next to the **Share** entity to perform the following operations
 - Select the **Preview** tab to view details about the share.
 - From the **Actions** drop-down list, you can select any of the following operations
 - Select **Manage Versions** to manage the different versions you create.
 - Select **Jobs** to view the **Jobs** table and see information related to it. A job is created whenever something needs to happen to a share. This includes the following:
 - * PUBLISH VERSIONED SHARE
 - * PUBLISH LIVE SHARE
 - * DROP SHARE VERSIONS
 - Select **Scheduling** to enable scheduling for the share entity.
 - Select **Publish** to publish the share with the updates you made.
 - Select **Unpublish** to unpublish the changes you did to the share.
 - Select **Objects** to view the **Select Tables** dialog of the specific share where you can remove tables from the list of Shared Tables list or add tables from the list of Available Tables list to the list of Shared Tables.
 - Select **Recipients and Profiles** to view the **Recipients and Profiles** dialog which enables you to create a new share recipient. You can also revoke the recipient right from the share by clicking X next to the recipient name. The Recipients and Profiles dialog enables you to update the list of recipients, copy profile activation link, send an activation mail to the selected recipient.
2. Select **Delete** to delete the share entity.

Share Recipient Entity Details

1. Click the **View Details** icon next to the **Share Recipient** entity to view the [share recipient entity details](#) perform the following operations
 - Select the **Overview** tab to view details about the Share Recipient entity.
 - From the **Actions** drop-down list, you can select any of the following operations:
 - Select **Get Bearer Token** to open an Auth Token dialog to authenticate the recipient. Use this option when your token expires. This enables you to generate a new token. A Bearer Token is part of the JSON file.
 - Select **Copy Profile Activation Link** to copy the JSON profile activation link to clipboard.
 - Select **Send Activation Mail** to send an activation mail to the Share Recipient.
 - Select **Rename** to rename the Recipient.
 - Select **Granted Shares** to open **Manage Granted Shares** dialog where you can grant access to a share from list of available shares or you can revoke access from the current selection of share.
2. Select **Delete** to delete the recipient you created entirely. By doing so, access to all shares will end instantly and any existing delta credentials will become invalid.

Cloud Storage Link Entity Details

1. Click the **View Details** icon next to the **Cloud Storage Link** entity to view the following operations
 - Select the **General** tab to view details about the cloud storage entity.
 - From the **Actions** drop-down list, you can select any of the following operations:
 - Select **Objects** to view objects available in the selected storage link. You can click a file on the navigator pane to view it on the display area.
 - Selecting **Create Live Table Feed** opens the Create Live Table feed wizard with the selected cloud storage link on the Cloud Location URL field. See [Feeding Data](#) to view more details.
 - Select **Edit** to update any details on the cloud storage location. See [Managing Connections](#) to view details on creation of cloud storage location.
 - Select **Rename** to rename the cloud store location to a different name.
 - Selecting **Link Tables** opens the Link Data page on the Data Load tool with the selected cloud storage link on the Cloud Location URL field. You can link data present in the cloud storage to the Autonomous Database. See [Linking to Objects in Cloud Storage](#).
 - Selecting **Load Tables** opens the Load Data page on the Data Load tool with the selected cloud storage link on the Cloud Location URL field. You can load data present in the cloud storage to the Autonomous Database. See [Loading Data from Cloud Storage](#).
2. Select **Delete** to delete the cloud store location.

Catalog Entity Details

1. Click the **View Details** icon next to the **Catalog** entity to view the following operations
 - Select the **General** tab to view details about the cloud storage entity.
 - From the **Actions** drop-down list, you can select any of the following operations:
 - Select **Disable** to disable the catalog.
2. Select **Delete** to delete the catalog.

Data Catalog Link Entity Details

1. Click the **View Details** icon next to the **Data Catalog Link** entity to view the following operations.
2. Select **Edit** to edit the data catalog.
3. Select **Rename** to rename the data catalog.
4. Select **Delete** to delete the data catalog.

Database Link Entity Details

1. Click the **View Details** icon next to the **Catalog** entity to view the following operations
 - From the **Actions** drop-down list, you can select any of the following operations:
 - Select **Load to table** to load the database link to a table.
 - Select **Link to table** to link the database link to a table.

- Select **Mount as catalog** to mount the database link as catalog.
2. Select **Delete** to delete the database link entity.

Live Table Feed Entity Details

1. Click the **View Details** icon next to the **Live Table Feed** entity to view the following operations:
 - Select the **Preview** tab to view details about the Live Table Feed.
 - Select **Edit Live Table Feed** to view the Edit Live Table Feed wizard where you can edit the live table.
 - Select **Delete Live Table Feed** to delete the Live Table Feed.
 - Select **Run Live Table Feed Immediately(Once)** to run the live table feed.
 - Select **Manage Notifications** to [Send Email using Live Feed Tool](#).

Share Provider Entity Details

1. Click the **View Details** icon next to the **Share Provider** entity to [view the share provider entity details](#):
 - From the **Actions** drop-down list, you can select any of the following operations:
 - Select **Load to table** to view the Load Data page with the Share tab selected.
 - Select **Link to table** to view the Link Data page with the Share tab selected to view and run the related link object. Drag and drop the shared data to add it to the data link job.
 - Select **Rename** to renames the Provider Name.
 - Select **Copy Endpoint** to copy the share endpoint.
2. Select **Manage Shares** for Share provider dialog box. This lists the shares you select to share with the recipient. You can edit the list of shares you wish to share with the recipient.
3. Select **Delete** to remove the Share Provider Entity.

Note

Some catalog entity types do not have details screens

There are certain items in the catalog that do not have detail screens yet, thus clicking on them might not provide more details.

- [Autonomous Database Entity Details](#)
You can mount additional catalogs from other Autonomous Databases in the same tenancy.

Autonomous Database Entity Details

You can mount additional catalogs from other Autonomous Databases in the same tenancy.

Before selecting whether or not to mount a catalog, you can browse the databases using the catalog to get additional information about each one.

Click **All** filter from the top and select **Autonomous Database** from OCI faceted filter panel in the left.

Selecting the above options lists all the autonomous databases in your tenancy.

You can view details about the database and launch the following applications from the **Database actions** drop-down field in the **Autonomous Database Entity** dialog box:

- Database Transforms
- Intelligent Data Lake
- Graph Studio
- APEX
- Machine learning notebook
- Machine learning user management
- Mongo Database
- ORDS
- SQL Developer

You can hover over the autonomous database whose details you want to view.

Click **Actions** next to the Autonomous Database entity and select **View Details** to view the following details:

General

The General tab displays the tenancy details, backup and recovery information, and network and maintenance details of the selected autonomous database.

Under **General Information**, you can view:

- **Database Name:** You set the database name when you provision a database or when you rename a database.
- **Workload Type:** The Autonomous Database supports different workload types, including: Data Warehouse, Transaction Processing, JSON Database, and APEX Service.
- **Compartment:** When creating a compartment, you must provide a name for it (maximum 100 characters, including letters, numbers, periods, hyphens, and underscores) that is unique within its parent compartment. Oracle will also assign the compartment a unique ID called an Oracle Cloud ID.
- **OCID:** It displays the Oracle Cloud ID. See [Resource Identifiers](#) for more information.
- **Created:** Displays the date when the autonomous database was created.
- **License type:** The License type field shows on the Autonomous Database Information tab when you enable the Bring Your Own License (BYOL) option. This field shows your license and Oracle Database Edition.
- **Database Version:** Specifies the version of the Database.
- **Lifecycle State:** You can view availability information for an autonomous database instance in this field.
- **Instance type:** You can view the instance details of the autonomous database.
- **Character Set:** The Autonomous Database default database character set is Unicode `AL32UTF8` and the default national character set is `AL16UTF16`. When you provision a database, depending on the workload type, you can select a database character set and a national character set.

- **Mode:** You can select any of the following autonomous database operation modes:
 - Read/Write
 - Read-Only
 - Restricted

The default mode is Read/Write.

Under **Resources** you can view:

- **Compute:** This field displays the autonomous database instance ECPU count (OCPU count if your database uses OCPUs).

Under **APEX** you can view, **instance name** that displays the autonomous database instance you use to create an APEX Service instance using an OCI.

Under **Data Safe** you can view, **Status** field that displays if you have registered the autonomous database with Oracle Data Safe. Oracle Data Safe, which is included with Autonomous Database, provides a unified control center that helps you manage the day-to-day security and compliance requirements of Oracle Databases.

Under **Disaster Recovery**, you can view:

- **Role:** Each autonomous database has a designated role: primary, standby, or snapshot standby.
- **Local:** If there is a local peer database, Autonomous Data Guard standby, this field shows the local standby's availability domain. See [Use Standby Databases with Autonomous Data Guard for Disaster Recovery](#) for more information.
- **Cross-region:** Displays if cross-region Autonomous Data Guard is enabled by provisioning a remote standby instance.

Under **Backup**, you can view:

- **Automatic backup retention period:** The backup retention period specifies how many days automatic backups are retained and the period for which you are billed for the automatic backups that are retained. If you change the backup retention period to a shorter period and automatic backups exist that are older, you are no longer billed for the automatic backups with timestamps beyond the specified retention period.
- **Total backup storage:** Displays the total storage being billed, including for automatic backups and if there are long-term backups, this also includes the long term backup storage.
- **Last Automatic backup:** Shows the timestamp of the last automatic backup.
- **Next long term backup:** When you configure scheduled long-term backups, this field displays the timestamp for the next long-term backup.
- **Long term backup schedule:** Displays the long-term backup schedule, with links to edit or delete a long-term backup schedule.

Under **Network**, you can view:

- **Access type:** Displays the specified access type for the Autonomous Database instance.
- **Access control list:** This field displays if Access Control Lists (ACLs) of an Autonomous Database Instance are defined. It can display the following two values:
 - **Enabled:** The field displays Enabled when ACLs of the Autonomous Database Instance are defined.

- **Disabled:** The field displays Disabled when ACLs of the Autonomous Database Instance are not defined.
- **Mutual TLS(mTLS) authentication:** Displays if your Autonomous Database instance is configured to only allow mTLS connections.

Under **Maintenance**, you can view:

- **Patch level:** You can view Autonomous Database patch information.
- **Next maintenance:** This field in the Oracle Cloud Infrastructure Console reflects a maintenance window date and time based on the patch level.
- **Customer contacts:** When customer contacts are set, Oracle sends notifications to the specified email addresses for Autonomous Database service-related issues.

Under **Encryption**, you can view the **Encryption key** field that displays the master encryption key the Oracle Autonomous Database creates.

Lineage

Lineage displays all known information about the upstream dependencies of the entity, and therefore how the entity was created and how it is linked to other entities.

For example, for an autonomous database that you create, the lineage is just the cloud. For an autonomous database you create from cloud storage, the lineage includes the ingest directive for the load and the cloud object store path that is the source of the database.

Pointing to the name of an item in the lineage displays the database name, the application that created it, the type of entity, the path to it, and the schema it is in.

Arrows point from an entity to the entity that it derives from.

You can increase or decrease the size of the displayed objects by using the + (plus) and - (minus) keys. You can reposition the objects by grabbing a blank spot in the display and dragging vertically or horizontally.

Impact

Impact shows all known information about the downstream use of an entity, and therefore how a change in the definition of an entity may affect other entities that depend on it.

Click **Actions** next to the Autonomous Database entity and select **+ Create Database Link** to create database links from the Autonomous Database. See [Load Data from Oracle and Non-Oracle Databases using Database Links](#).

Click **Actions** next to the Autonomous Database entity and select **Create Share Recipient** to create share recipient from the Autonomous Database. See [Create Share Recipient](#) to create a share recipient from the Autonomous Database.

Registering Cloud Links to Access Data

Cloud Links enable you to remotely access read only data on an Autonomous AI Database instance.

You can register a table for remote access for a selected audience. Scope indicates who can remotely access the data. Scope can be set to various levels, including to the region where the database resides, to individual tenancies, or to compartments. You can provide a namespace and a name other than the original schema and object names. For example, you can register a

table under the namespace FOREST and for security purposes or for naming convenience, you can provide a namespace and a name other than the original schema and object names.

To register a table to cloud link, click **Actions** next to the Table entity and select **Register to Cloud Link**.

Specify the following field values:

- **Cloud Link Namespace:** Enter a namespace to register the table. For example, ADMIN.
- **Cloud Link Name:** Enter a name for the cloud link. For example, Sales.
- **Description:** Specify the description of your cloud link.
- **Scope:** Specifies who and from where a user is allowed to access the registered table. You can choose from any of the following available options:
 - **MY\$REGION:** You can grant remote data access to other tenancies in the region of the Autonomous AI Database instance that is registering the data set. This is the least restrictive scope.
 - **MY\$TENANCY:** You can grant remote data access to any resource, tenancy, compartment, or database in the tenancy of the Autonomous AI Database instance that is registering the data set. This scope is more restrictive than MY\$REGION scope.
 - **MY\$COMPARTMENT:** You can grant remote data access to any resource, compartment, or database in the compartment of the Autonomous AI Database instance that is registering the data set.
 - **OCID:** Access to the data set is allowed for the specific Autonomous AI Database instances identified by OCID.

Click **OK** to finish the registration of the selected table with the cloud link.

Export Data to Cloud

Use the **Export Data to Cloud** menu in the table actions of the Browse Catalog page to export data as text from an Autonomous Database to a cloud Object Store. The text format export options are CSV, JSON, Parquet, or XML.

To export data to a cloud storage location, click **Actions** next to the Table entity on the Browse Catalog page and select **Export Data to Cloud**.

Export Data to Cloud Storage Location

Table Name
MEDICARE_XT

Export As
JSON Show Options

Cloud storage location *
AMS_SHARE (https://e...1.oraclecloud.com/n/adwcdemo/b/AMS_share2/o/) Create Cloud location

✔ **Cloud Location Access**
You have access to Read, Write and Delete

File prefix
MEDICARE_XT

Show code Export Cancel

The Export Data to Cloud Storage Location displays the following fields:

- **Table Name:** This field displays the table name.
- **Export As:** This drop-down displays the list of format options you wish to export the data. You can select CSV, JSON, Parquet, or XML. When you export the data as **CSV**, and click **Show Options**, you can:
 - Set the maximum file size value using the slider. The maximum file size you can export is from 10MB to 1GB.
 - Select the **header** option to write column names as the first line in output files.
 - Select the **Escape** option to specify the occurrence of the quote character in the field value using the “\” character.
 - **Text Enclosure and Field delimiter:** These options are visible only when the selected file is in CSV format. Select or enter the character used in the source file for text enclosure and field delimiter.
 - **Compression:** Select **No Compression** to disable compression or **GZIP** to enable GZIP compression for selected file.
- **Cloud storage Location:** Select the uri value to the URL for an existing bucket on your Cloud Object Storage from the drop-down. If you do not have a cloud storage location, select **Create Cloud Location** to create a cloud storage location to export your data. For more details, see [Create Credentials](#).

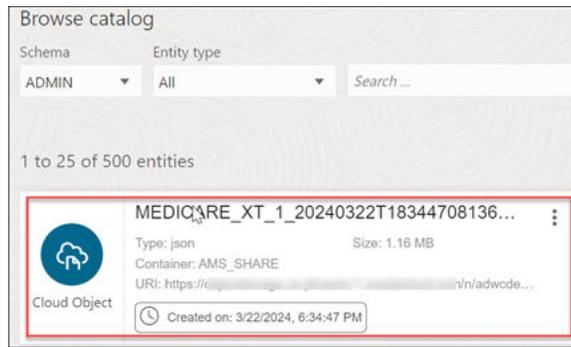
i **Note**

The Cloud storage location is highlighted in red if the tool doesn't accept the selected cloud location. In this case, you will receive a Cloud Location Access Error that says “You do not have access to Read, Write and Delete”.

- **File prefix:** Specify the File prefix. This will be the prefix of the file you export.

Click **Export**.

After the file export is complete, you will view the newly created cloud object in the entity list of the Browse Catalog page.



Gathering Statistics for Tables

You can generate statistics that measure the data distribution and storage characteristics of tables.

You must gather statistics periodically for tables where the statistics become stale over time because of changing data volumes or changes in column values. The Gather Statistics icon gathers new statistics after the table's structure are modified in ways that make the previous statistics inaccurate. For example, after loading a significant number of rows into a table, collect new statistics on the number of rows. After updating data in a table, you do not need to collect new statistics on the number of rows, but you might need new statistics on the average row length.

Table Statistics also include column statistics. The statistics you generate include the following:

Table statistics:

- **Table Size:** Specifies the size of the table in bytes.
- **Number of rows:** Displays the number of rows in the table.
- **Number of columns:** Displays the number of columns in the table
- **Compressed size:** Displays the size of compressed table in bytes.
- **Statistics gathered on:** Displays date and time of last statistics gathered.

Column Statistics

- Number of distinct values (NDV) in column
- Number of nulls in column
- Data distribution (histogram)

The above information is displayed in the statistics details of tables.

Editing Tables

You can create and edit objects using Edit Table wizard available from the **Edit** menu in **Actions** (three vertical dots) besides the table entity.

Clicking **Edit** from the Actions menu opens the **Edit Table** wizard. You can visit the panes in any order to edit a table. The table properties are grouped in several panes.

- **Schema:** Database schema in which the table exists.

- **Name:** Name of the table.

The different panes in the dialog are described in the following sections:

- Columns Pane
- Primary Key Pane
- Unique Keys Pane
- Indexes Pane
- Foreign Keys Pane
- Table Constraints Pane
- Comments Pane
- Storage Pane
- External Table Properties Pane
- Materialized View Pane
- DDL Pane
- Output Pane
- [Columns Pane](#)
Specifies properties for each column in the table.
- [Primary Key Pane](#)
Specifies the primary key for the table.
- [Unique Keys Pane](#)
Specifies one or more unique constraints for the table.
- [Indexes Pane](#)
Lists the indexes defined for the table.
- [Foreign Keys Pane](#)
Specifies one or more foreign keys for the table.
- [Table Constraints Pane](#)
Specifies one or more check constraints for the table.
- [Comments Pane](#)
Enter descriptive comments in this pane.
- [Storage Pane](#)
Enables you to specify storage options for the table.
- [External Table Properties Pane](#)
Specifies options for an external table.
- [Materialized View Pane](#)
Specifies options for a materialized view.
- [DDL Pane](#)
You can review and save the SQL statements that are generated when creating or editing the object. If you want to make any changes, go back to the relevant panes and make the changes there.
- [Output Pane](#)
Displays the results of the DDL commands.

Columns Pane

Specifies properties for each column in the table.

General tab

Lists the columns available in the table. To add a column, click **Add Column (+)**. A new row is added to the table below. Select the row and enter the details for the column. To delete a column, select the row and click **Remove Column (-)**. To move a column up or down in the table, select it and use the up-arrow and down-arrow icons.

- **Name:** Name for the column.
- **Datatype:** Data type for the column.
- **Default:** If no value is specified, the default value is null.
- **Default on NULL:** Applicable for Oracle Database 12c and later releases. If this option is selected, when a row is inserted into the table and the value specified for the column is NULL, the default value is inserted into the column.
- **Expression:** Expression for computing the value in the column.
- **Comments:** Optional descriptive comments about the column. Use this field to provide descriptions for the attributes.

In the table:

- **PK:** If this option is selected, the column becomes the primary key.
- **Identity Column :** If this option is selected, the column becomes an identity column. This is applicable only for Oracle Database 12c and later releases. For more details, see the Identity Column tab.

Constraints tab

Displays the Not Null and Check Constraints for a column. A check constraint requires values in a column to comply with a specified condition.

- **Not Null Constraint: Name:** Name for the Not Null constraint.
- **Not Null Constraint: Not Null:** If this option is selected, the column must contain data. You cannot specify no value or an explicit null value for this column when you insert a row. If this option is not checked, the column can contain either data or no data. A primary key column cannot be null.
- **Check Constraint: Name:** Name for the check constraint definition.
- **Check Constraint: Constraint:** Condition that must be met for a column to fulfill the check constraint. You can use any valid CHECK clause (without the CHECK keyword). For example, to indicate that the value in a numeric column named RATING must be from 1 to 10, you can specify: rating >=1 and rating <= 10.
- **Enabled:** If this option is selected, the constraint is checked when data is entered or updated in the column.
- **Deferrable:** If this option is selected, you can defer checking the validity of the constraint until the end of a transaction.
- **Initially Immediate:** If this option is selected, the constraint is checked whenever you add, update, or delete data from the column.
- **Validate:** If this option is selected, the existing data is checked to see if it conforms to the constraint.

Primary Key Pane

Specifies the primary key for the table.

The primary key is the column, or set of columns, that uniquely identifies each row in the table. If the Primary Key checkbox is selected for a column in the General tab, the corresponding fields are automatically populated in the Primary Key pane. You can make changes to the properties as required.

An index is automatically created on the primary key.

- **Name:** Name of the constraint to be associated with the primary key definition.
- **Enabled:** If this option is checked, the primary key constraint is enforced: that is, the data in the primary key column (or set of columns) must be unique and not null.
- **Index:** Name of the index to which the primary key refers. Tablespace: Name of the tablespace associated with the index.
- **Available Columns:** Lists the columns that are available to be added to the primary key definition. You can select multiple attributes, if required, for the primary key.
- **Selected Columns:** Lists the columns that are included in the primary key definition. To add a column to the primary key definition, select it in Available Columns and click the Add (>) icon; to remove a column from the primary key definition, select it in Selected Columns and click the Remove (<) icon. To move all columns from available to selected (or the reverse), use the Add All (>>) or Remove All (<<) icon. To move a column up or down in the primary key definition, select it in Selected Columns and use the arrow buttons.

Unique Keys Pane

Specifies one or more unique constraints for the table.

A unique constraint specifies a column, or set of columns, whose data values must be unique: each data value must not be null, and it must not be the same as any other value in the column.

To add a unique constraint, click the Add button; to delete a unique constraint, select it and click the Remove button.

- **Name:** Name of the unique constraint.
- **Enabled:** If this option is selected, the unique constraint is enforced.
- **Rely:** If this option is selected, the constraint in NOVALIDATE mode is taken into account during query rewrite.
- **Deferrable:** If this option is selected, in subsequent transactions, constraint checking can be deferred until the end of the transaction using the SET CONSTRAINT(S) statement.
- **Initially Immediate:** If this option is selected, the constraint is checked at the end of each subsequent SQL statement.
- **Validate:** If the option is selected, the existing data is checked to see if it conforms to the constraint.
- **Index:** Name of the index to which the unique key refers.
- **Tablespace:** Name of the tablespace associated with the index.
- **Available Columns:** Lists the columns that are available to be added to the unique constraint definition.

- **Selected Columns:** Lists the columns that are included in the unique constraint definition. To add a column to the unique constraint definition, select it in Available Columns and click the Add (>) icon; to remove a column from the unique constraint definition, select it in Selected Columns and click the Remove (<) icon. To move all columns from available to selected (or the reverse), use the Add All (>>) or Remove All (<<) icon. To move a column up or down in the unique constraint definition, select it in Selected Columns and use the arrow buttons.

Indexes Pane

Lists the indexes defined for the table.

To add an index, click Add Index (+); to delete an index, select it and click Remove Index (-).

- **Name:** Name of the index.
- **Type:** The type of Oracle index. Non-unique means that the index can contain multiple identical values; Unique means that no duplicate values are permitted; Bitmap stores rowids associated with a key value as a bitmap.
- **Tablespace:** Name of the tablespace for the index.
- **Expression:** A column expression is an expression built from columns, constants, SQL functions, and user-defined functions. When you specify a column expression, you create a function-based index.
- **Available Columns and Selected Columns:** Columns selected for the index. To select a column, click the column in the Available Columns box, and then click the Add Selected Columns icon to move it to the Selected Columns box.

Foreign Keys Pane

Specifies one or more foreign keys for the table.

A foreign key specifies a column ("local column"), whose data values match values in the primary key or unique constraint of another table.

- **Name:** Name of the foreign key definition.
- **Enabled:** If this option is checked, the foreign key is enforced.
- **Rely, Deferrable, Initially Immediate, Validate:** See the description of these fields in the Unique Keys pane.
- **Referenced Constraint: Schema:** Name of the schema containing the table with the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: Table:** Name of the table with the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: Constraint:** Name of the primary key or unique constraint to which this foreign key refers.
- **Referenced Constraint: On Delete:** Action to take automatically when a row in the referenced table is deleted and rows with that value exist in the table containing this foreign key: NO ACTION (shown by a crossing line in diagrams) performs no action on these rows; CASCADE (shown by an "X") deletes these rows; SET NULL (shown by a small circle) sets null all columns in those rows that can be set to a null value.
- **Associations: Local Column:** Lists the column in the currently selected (local) table that is included in the foreign key definition. For each referenced column in the foreign key definition, select the name of a column in the edited table.

- **Associations: Referenced Column:** For each local column, identifies the column in the other (foreign) table that must have a value matching the value in the local column.

Table Constraints Pane

Specifies one or more check constraints for the table.

A check constraint specifies a condition that must be met when a row is inserted into the table or when an existing row is modified.

- **Name:** Name of the check constraint definition.
- **Check Condition:** Condition that must be met for a row to fulfil the check constraint. You can use any valid CHECK clause (without the CHECK keyword). For example, to indicate that the value in a numeric column named RATING must be from 1 to 10, you can specify rating >=1 and rating <= 10.
- **Enabled:** If this option is checked, the check constraint is enforced.

Comments Pane

Enter descriptive comments in this pane.

This is optional.

Storage Pane

Enables you to specify storage options for the table.

When you create or edit a table or an index, you can override the default storage options.

- **Organization:** Specifies that the table is stored and organized with (Index) or without an index (Heap) or as an external table (External).
- **Tablespace:** Name of the tablespace for the table or index.
- **Logging:** ON means that the table creation and any subsequent INSERT operations against the table are logged in the redo log file. OFF means that these operations are not logged in the redo log file.
- **Row Archival:** YES enables in-database archiving, which allows you to archive rows within the table by marking them as invisible.

External Table Properties Pane

Specifies options for an external table.

An external table is a read-only table whose metadata is stored in the database but whose data is stored outside the database.

External Table

- **Access Driver Type:** Specifies the type of external table.
 - **ORACLE_LOADER:** Extracts data from text data files. This is the default access driver, which loads data from external tables to internal tables.
 - **ORACLE_DATAPUMP:** Extracts data from binary dump files. This access driver can perform both loads and unloads.
 - **ORACLE_BIGDATA:** Extracts data from Oracle Big Data Appliance.

- **ORACLE_HDFS**: Extracts data stored in a Hadoop Distributed File System (HDFS).
- **ORACLE_HIVE**: Extracts data stored in Apache HIVE.
- **Default Directory**: Specifies the default directory to use for all input and output files that do not explicitly name a directory object. The location is specified with a directory object, not a directory path.
- **Access Params**: Assigns values to the parameters of the specific access driver for the external table. Access parameters are optional.
 - **OPAQUE_FORMAT_SPEC**: The `opaque_format_spec` specifies all access parameters for the `ORACLE_LOADER`, `ORACLE_DATAPUMP`, `ORACLE_HDFS`, and `ORACLE_HIVE` access drivers. For descriptions of the access parameters, see *Oracle Database Utilities*. Field names specified in the `opaque_format_spec` must match columns in the table definition, else Oracle Database ignores them.
 - **USING CLOB**: Enables you to derive the parameters and their values through a subquery. The subquery cannot contain any set operators or an `ORDER BY` clause. It must return one row containing a single item of data type `CLOB`.
- **Reject Limit**: The number of conversion errors that can occur during a query of the external data before an Oracle Database error is returned and the query is aborted.
- **Project Column**: Determines how the access driver validates the rows of an external table in subsequent queries.
 - **ALL**: Processes all column values, regardless of which columns are selected, and validates only those rows with fully valid column entries. If any column value raises an error, such as a data type conversion error, the row is rejected even if that column was not referenced in the select list of the query.
 - **REFERENCED**: Processes only those columns in the select list of the query. The `ALL` setting guarantees consistent result sets. The `REFERENCED` setting can result in different numbers of rows returned, depending on the columns referenced in subsequent queries, but is faster than the `ALL` setting. If a subsequent query selects all columns of the external table, then the settings behave identically.
- **Location**: Specifies the data files for the external table. Use the Add (+) icon to add each location specification.
 - `ORACLE_LOADER` and `ORACLE_DATAPUMP`, the files are named in the form `directory:file`. The directory portion is optional. If it is missing, then the default directory is used as the directory for the file. If you are using the `ORACLE_LOADER` access driver, then you can use wildcards in the file name. An asterisk (*) signifies multiple characters and a question mark (?) signifies a single character.
 - For `ORACLE_HDFS`, `LOCATION` is a list of Uniform Resource Identifiers (URIs) for a directory or for a file. There is no directory object associated with a URI.
 - For `ORACLE_HIVE`, `LOCATION` is not used. Instead, the Hadoop HCatalog table is read to obtain information about the location of the data source (which could be a file or another database).

Opaque Format Spec

Specifies all access parameters for the `ORACLE_LOADER`, `ORACLE_DATAPUMP`, `ORACLE_HDFS`, and `ORACLE_HIVE` access drivers.

CLOB Subquery

Type or copy and paste the query.

Materialized View Pane

Specifies options for a materialized view.

Query: Contains the SQL code for the query part of the view definition. Type or copy and paste the query.

General

- **On Pre-built Table:** If **Yes**, an existing table is registered as a preinitialized materialized view. This option is particularly useful for registering large materialized views in a data warehousing environment. The table must have the same name and be in the same schema as the resulting materialized view, and the table should reflect the materialization of a subquery.
- **Reduced Precision:** **Yes** authorizes the loss of precision that will result if the precision of the table or materialized view columns do not exactly match the precision returned by the subquery. If **No**, the precision of the table or materialized view columns must exactly match the precision returned by the subquery, or the create operation will fail.
- **For Update:** Select **Yes** to allow a subquery, primary key, object, or rowid materialized view to be updated. When used in conjunction with Advanced Replication, these updates will be propagated to the master.
- **Real Time MV:** Select **Yes** to create a real-time materialized view or a regular view. A real-time materialized view provides fresh data to user queries even when the materialized view is not in sync with its base tables due to data changes. Instead of modifying the materialized view, the optimizer writes a query that combines the existing rows in the materialized view with changes recorded in log files (either materialized view logs or the direct loader logs). This is called on-query computation.
- **Query Rewrite:** If **Enable**, the materialized view is enabled for query rewrite, which transforms a user request written in terms of master tables into a semantically equivalent request that includes one or more materialized views.
- **Build:** Specifies when to populate the materialized view. **Immediate** indicates that the materialized view is to be populated immediately. **Deferred** indicates that the materialized view is to be populated by the next refresh operation. If you specify **Deferred**, the first (deferred) refresh must always be a complete refresh; until then, the materialized view has a staleness value of unusable, so it cannot be used for query rewrite.
- **Use Index:** If **Yes**, a default index is created and used to speed up incremental (fast) refresh of the materialized view. If **No**, this default index is not created. (For example, you might choose to suppress the index creation now and to create such an index explicitly later.)
- **Index Tablespace:** Specifies the tablespace in which the materialized view is to be created. If a tablespace is not selected, the materialized view is created in the default tablespace of the schema containing the materialized view.
- **Cache:** If **Yes**, the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed. This setting is useful for small lookup tables. If **No**, the blocks are placed at the least recently used end of the LRU list.

Refresh Clause

- **Refresh:** Select **Yes** to enable refresh operations.
- **Refresh Type:** The method of refresh operation to be performed:

- Complete Refresh: Executes the defining query of the materialized view, even if a fast refresh is possible.
- Fast Refresh: Uses the incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables. The changes for conventional DML changes are stored in the materialized view log associated with the master table. The changes for direct-path INSERT operations are stored in the direct loader log.
- Force Refresh: Performs a fast refresh if one is possible; otherwise, performs a complete refresh.
- **Action:** The type of refresh operation to be performed:
 - On Demand: Performs a refresh when one of the DBMS_MVIEW refresh procedures are called.
 - On Commit: Performs a fast refresh whenever the database commits a transaction that operates on a master table of the materialized view. This may increase the time taken to complete the commit, because the database performs the refresh operation as part of the commit process.
 - Specify: Performs refresh operations according to what you specify in the Start on and Next fields.
- **Start Date:** Starting date and time for the first automatic refresh operation. Must be in the future.
- **Next Date:** Time for the next automatic refresh operation. The interval between the Start on and Next times establishes the interval for subsequent automatic refresh operations. If you do not specify a value, the refresh operation is performed only once at the time specified for Start on.
- **With:** Refresh type, which determines the type of materialized view:
 - Primary Key: Creates a primary key materialized view, which allows materialized view master tables to be reorganized without affecting the eligibility of the materialized view for fast refresh.
 - Row ID: Creates a rowid materialized view, which is useful if the materialized view does not include all primary key columns of the master tables.
- **Default Storage:** If Yes, DEFAULT specifies that Oracle Database will choose automatically which rollback segment to use. If you specify DEFAULT, you cannot specify the `rollback_segment`. DEFAULT is most useful when modifying, rather than creating, a materialized view.
- **Storage Type:** MASTER specifies the remote rollback segment to be used at the remote master site for the individual materialized view. LOCAL specifies the remote rollback segment to be used for the local refresh group that contains the materialized view. This is the default.
- **Rollback Segment:** Enter the name of the rollback segment.
- **Using Constraint:** If this option is checked, more rewrite alternatives can be used during the refresh operation, resulting in more efficient refresh execution. The behavior of this option is affected by whether you select Enforced or Trusted.
 - Enforced: Causes only enforced constraints to be used during the refresh operation.
 - Trusted: Enables the use of dimension and constraint information that has been declared trustworthy by the database administrator but that has not been validated by the database. If the dimension and constraint information is valid, performance may improve. However, if this information is invalid, then the refresh procedure may corrupt the materialized view even though it returns a success status.

DDL Pane

You can review and save the SQL statements that are generated when creating or editing the object. If you want to make any changes, go back to the relevant panes and make the changes there.

For a new table, click **CREATE** to view the generated DDL statements.

When you edit table properties, click **UPDATE** to view the generated ALTER statements.

For a new table, the UPDATE tab will not be available. When you are finished, click **Apply**.

Output Pane

Displays the results of the DDL commands.

If there are any errors, go to the appropriate pane, fix the errors, and run the commands again. You can save to a text file or clear the output.

Manage Catalogs with DBMS_CATALOG

The `DBMS_CATALOG` package provides a comprehensive set of procedures, functions, and types for managing database catalogs for Oracle Autonomous AI Database.

What is a Catalog?

A *catalog* is defined to be a collection of named *schemas*, each of which contains a collection of named objects, such as `TABLES` or `VIEWS`. Every Oracle database has a single *local* catalog – the data dictionary. This catalog is always present and cannot be removed. But there are other examples of catalogs, defined outside the database.

- The set of objects available through a database link
- A set of shared objects available through `DBMS_SHARE` (e.g. Delta Sharing);
- A set of Iceberg tables managed by an Iceberg REST Catalog;
- A set of objects defined by a third party product, such as Amazon Glue, Databricks Unity Catalog, or Snowflakes Polaris;
- A set of objects defined by the Oracle OCI Data Catalog Service.
- A *Catalog* can also be thought of as a "Domain", or "Data Product", which is a set of objects grouped together for a specific business purpose.

In an operating system you can access external data files by [mounting a file system](#). By analogy, you can access external data in an Autonomous AI Database by *mounting a catalog* using the `DBMS_CATALOG` package.

The following example shows how to mount an external Iceberg REST Catalog. To use it, you would need three pieces of information.

- The Iceberg REST Catalog endpoint;
- A credential (e.g. a bearer token) used to call this endpoint;

- A credential (e.g. a username/password) used to access the bucket where the Iceberg data files are stored.

```

BEGIN
  -- Create a credential capable of accessing an external Iceberg REST catalog
  dbms_cloud.create_credential('ICEBERG_CATALOG_CRED', ...);

  -- Create a credential capable of accessing the bucket where the
  -- Iceberg data files are stored.
  dbms_cloud.create_credential('ICEBERG_STORAGE_CRED', ...);

  -- Mount the iceberg catalog
  dbms_catalog.mount_iceberg(
    catalog_name          => 'ICEBERG_CAT',
    endpoint              => 'https://...',
    catalog_credential    => 'ICEBERG_CATALOG_CRED',
    data_storage_credential => 'ICEBERG_STORAGE_CRED',
    catalog_type          => 'ICEBERG_UNITY');
END;
/

-- List tables in the iceberg catalog
SELECT owner, table_name
FROM all_tables@iceberg_cat;

-- Read data from an iceberg table
SELECT *
FROM a_schema.a_table@iceberg_cat;

```

The DBMS_CATALOG is used to define and handle operations related to the catalog.

It provides a comprehensive set of procedures, functions, and types for managing catalogs within Oracle Autonomous database. It supports operations such as mounting and unmounting catalogs, managing catalog properties, handling credentials, and working with catalog entities like tables, schemas, and objects. This package is essential for integrating external data sources and managing metadata in a secure and efficient manner.

Security Model

The DBMS_CATALOG package operates under the AUTHID CURRENT_USER model, meaning it runs with the privileges of the current user. You must have appropriate permissions to perform operations such as mounting catalogs, updating properties, or managing credentials.

Note

Few catalogs require Autonomous AI Database to connect to external internet resources. Therefore, you must add the relevant external addresses to the access control lists for the associated database user. For example, when enabling access to a Databricks Unity Iceberg catalog on Azure, you may need to whitelist two addresses: one for accessing the Iceberg REST APIs and another if the user requires access to the underlying data.

```
BEGIN
  dbms_network_acl_admin.append_host_ace(
    host => '*.azuredatabricks.net',
    lower_port => 443,
    upper_port => 443,
    ace => xs$ace_type(
      privilege_list => xs$name_list('http', 'http_proxy'),
      principal_name => 'DBUSER',
      principal_type => xs_acl.p_type_db));

  dbms_network_acl_admin.append_host_ace(
    host => '*.blob.core.windows.net',
    ace => xs$ace_type(
      privilege_list => xs$name_list('http', 'http_proxy'),
      principal_name => 'DBUSER',
      principal_type => xs_acl.p_type_db));
END;
/
```

You must have the DWROLE to run the DBMS_CATALOG methods.

Run the following command to grant DWROLE to the user:

```
GRANT DWROLE TO MY_USER;
```

Catalog Types and Constants

This section provides an overview of the catalog types and key constants defined in the package. It explains the different catalog categories used to organize various catalog sources and outlines essential constants for configuring and managing catalog properties and behaviors.

Catalog Types

The package specifies multiple catalog types to classify and handle different categories of catalogs. They are:

- **CATALOG_TYPE_SHARE:** Represents a share-based catalog, which is designed primarily for sharing data between systems or users. For example, a data sharing service like Data Studio's Data Sharing uses share-based catalogs to distribute data securely.
- **CATALOG_TYPE_DATA_CATALOG:** Represents external data catalogs such as AWS Glue or OCI Data Catalog. These catalogs serve as centralized repositories for metadata about datasets. For instance, the AWS Glue Data Catalog automatically discovers and manages

metadata of data sources across AWS services like S3 and Redshift, allowing seamless data integration and querying.

- `CATALOG_TYPE_DB_LINK`: Represents external data catalogs such as AWS Glue or OCI Data Catalog. These catalogs serve as centralized repositories for metadata about datasets. For instance, the AWS Glue Data Catalog automatically discovers and manages metadata of data sources across AWS services like S3 and Redshift, allowing seamless data integration and querying. See [Load Data from Oracle and Non-Oracle Databases using Database Links](#) for more information.
- `CATALOG_TYPE_VIRTUAL`: Represents virtual catalogs that provide an abstraction over physical data sources. Virtual catalogs do not store data themselves but present a unified interface to query disparate data sources — for example, virtual views created in a data virtualization platform.
- `CATALOG_TYPE_ICEBERG`: Represents an Iceberg catalog, which manages metadata for tables stored in the Apache Iceberg format. Iceberg is a table format designed for huge analytic datasets, supporting features like schema evolution and time travel. See [Manage Catalogs](#) for more information.

Constants

The following section highlights the essential constants that define catalog properties and settings. The constants include:

- `TYPE_CATALOG` and `NS_CATALOG`: This defines the catalog type and namespace as `CATALOG`.
- `DEFAULT_CATALOG`: The default catalog name set to `LOCAL`.
- Property constants such as `PROP_IS_ENABLED`, `PROP_IS_SYNCHRONIZED`, `PROP_CACHE_ENABLED`, `PROP_CACHE_DURATION`, and others for managing catalog behavior and metadata caching.

Constant Name	Value	Description
<code>PROP_IS_ENABLED</code>	<code>IS_ENABLED</code>	<p>This property determines whether a catalog is currently enabled for query and search.</p> <p>Valid Values:</p> <ul style="list-style-type: none"> – YES: The catalog will appear in the Data Studio UI search dialog and can be used in SQL queries using the database link syntax (e.g. <code>select username from all_users@catalog</code>). – NO: The catalog will be listed in <code>ALL_MOUNTED_CATALOGS</code> view, but will not be included in Data Studio UI searches and cannot be used in database link syntax.

Constant Name	Value	Description
PROP_CACHE_ENABLED	CACHE_ENABLED	<p>This property determines whether metadata from a remote schema should be cached in the local database to improve performance.</p> <p>Valid Values</p> <ul style="list-style-type: none"> – YES The metadata will be cached <i>on first access</i> and will remain in the cache for a period specified by PROP_CACHE_DURATION. – NO: The metadata will not be cached. All attempts to access objects within the catalog will reach out, at query time, to the remote metadata source.
PROP_CACHE_DURATION	CACHE_DURATION	<p>The number of seconds that metadata should be held in the cache before it becomes stale. The default value is 3600 seconds (one hour).</p> <p>Note that the user may flush the cache manually at any point by calling DBMS_CATALOG.FLUSH_CATALOG_CACHE.</p> <p>The user may refresh the cache at any point by calling DBMS_CATALOG.PREFILL_CATALOG_CACHE.</p>
PROP_CACHE_ASYNC	'CACHE_ASYNC'	<p>This field displays if the cache be populated using asynchronous jobs.</p> <p>Valid Values</p> <ul style="list-style-type: none"> – YES: The metadata will be cached using a DBMS_SCHEDULER job that is created the first time a user requests the metadata. – NO: The metadata will be cached using the database session of the user who requests the metadata.

Constant Name	Value	Description
PROP_DEFAULT_SCHEMA	DEFAULT_SCHEMA	<p>The name of a remote schema that will be chosen by default if the user runs a query of the form:</p> <pre>select * from table@catalog</pre> <p>The default schema plays the same role as the user's own schema in the local database.</p>
PROP_DCAT_TYPE	DATA_CATALOG_TYPE	<p>The type of a catalog mounted by DBMS_CATALOG.MOUNT_DATA_CATALOG</p> <p>Valid Values</p> <ul style="list-style-type: none"> – AWS_GLUE: The catalog is defined on top of a remote AWS Glue repository. – OCI_DCAT: The catalog is defined on top of an OCI Data Catalog instance.

 **Note**

This is a *read only* property.

Constant Name	Value	Description
PROP_CUSTOM	CUSTOM	<p>A custom property is a name/value pair associated, by the user, with a catalog, schema, table, or other object within the catalog.</p> <p>There are two ways to specify a custom property:</p> <ul style="list-style-type: none"> – You can set a single custom property, "MY_PROP" say, by specifying the composite property name 'CUSTOM:MY_PROP' along with an arbitrary string value. <pre>DBMS_CATALOG.UPDATE_CATALOG_PROPERTY(catalog_name => 'some_catalog', catalog_property => 'CUSTOM:MY_PROP', new_value => 'Property Value');</pre> <ul style="list-style-type: none"> – You can specify a group of customer properties by specify the simple property name 'CUSTOM' along with a JSON object that contains name value pairs. <pre>DBMS_CATALOG.UPDATE_CATALOG_PROPERTY(catalog_name => 'some_catalog', catalog_property => 'CUSTOM', new_value => '{"Property1" : "Value 1", "Property2" : "Value 2", ...}');</pre>

Note

Custom property names are case sensitive, so CUSTOM:MY_PR

Constant Name	Value	Description
		<p>OP is distinct from CUSTOM:My_Pr op.</p> <p>To remove a custom property, set the value to NULL.</p> <p>Custom properties can be used as search terms in the Data Studio UI. For example, you can find all tables with the property MY_PROP by specifying the conditions parameter in DBMS_CATALOG.GET_TABLES.</p> <pre>SELECT table_name FROM DBMS_CATALOG.GET_TABLES(catalog_name => 'some_catalog', conditions => '#MY_PROP');</pre> <p>You can also search for specific values.</p> <pre>SELECT table_name FROM DBMS_CATALOG.GET_TABLES(catalog_name => 'some_catalog', conditions => '#MY_PROP="Property Value"');</pre>

Constant Name	Value	Description
PROP_METADATA	METADATA	<p>Custom metadata is similar to custom properties, but it can contain free-form JSON. The metadata is stored with the object and can be retrieved, but it is not used as a search term.</p> <p>There are two ways to specify custom metadata:</p> <ul style="list-style-type: none"> – You can set the entire metadata, as JSON, by using the property name 'METADATA'. <pre>DBMS_CATALOG.UPDATE_CATALOG_PROPERTY(catalog_name => 'some_catalog', catalog_property => 'METADATA', new_value => '{"subObject": {"propName": "abc"}}')</pre> <ul style="list-style-type: none"> – You can update a sub-component of the metadata by using the composite property name 'METADATA:path', where path is some relative JSON path. <pre>DBMS_CATALOG.UPDATE_CATALOG_PROPERTY(catalog_name => 'some_catalog', catalog_property => 'METADATA:subObject. propName', new_value => 'xyz');</pre>

Constant Name	Value	Description
PROP_CONFIGURATION	CONFIGURATION	<p>The configuration property is used to update specific configuration properties for Iceberg catalogs. It works in a similar way to the CUSTOM property.</p> <pre> BEGIN dbms_catalog.update_cat alog_property('iceberg_cat', 'CONFIGURATION:IS_CASE_ SENSITIVE', 'YES'); END; / </pre>

Data Types

The DBMS_CATALOG package introduces various custom data types to manage catalog-related information. They are:

- `credential_info` and `credential_info_map`: These are structures to store and map credential information.
- `catalog_table` and `catalog_tables`: These are records and tables for storing detailed table metadata (e.g., owner, name, description, status).
- `catalog_schema` and `catalog_schemas`: These are records and tables for schema metadata.
- `catalog_object` and `catalog_objects`: These are records and tables for object metadata within a catalog.
- [DBMS_CATALOG Package](#)
The DBMS_CATALOG package includes a wide range of subprograms for catalog management, grouped by functionality.
- [Summary of DBMS_CATALOG](#)
This section covers the DBMS_CATALOG subprograms provided with Autonomous AI Database.

DBMS_CATALOG Package

The DBMS_CATALOG package includes a wide range of subprograms for catalog management, grouped by functionality.

- [Catalog Mounting and Unmounting](#):
You can add a new catalog to your Autonomous AI Database, either by mounting an existing connection (for example an existing database link to a database), or by adding a new connection and registering the external system as a new catalog.

You can enable or disable a catalog.

Note

Disabled catalogs keep connection details and any current metadata but are not updated and are not accessible either from UIs or APIs.

Remove or detach a catalog from your Autonomous AI Database.

- [Catalog Property Management:](#)

The subprograms mentioned here allows you to retrieve and update various properties of catalogs, schemas, and tables within your system. Specifically, it provides functions to:

- Get the value of a property for a catalog, schema, or table, including options to handle large text values via CLOB format.
- Update the value of properties for catalogs, schemas, or tables, with support for both regular and CLOB data types.

- [Credential Management:](#)

A *mounted* catalog will have zero or more *required credentials*. These are credentials that need to be specified before the catalog can be used. Iceberg catalogs, for example, typically have two *required credentials*.

- **ICEBERG_CREDENTIAL:** A credential used to call the Iceberg REST Catalog endpoints. This is usually a bearer token credential.
- **DATA_STORAGE_CREDENTIAL:** A credential used to access the Iceberg files in the bucket.

You can think of *required credential* as a parameter that needs to be specified. A *local credential* is an actual CREDENTIAL object in the user's schema that plays the role of one of a required credential. You can specify the *local credentials* when you mount the catalog, using arguments like `catalog_credential` and `data_storage_credential` in `DBMS_CATALOG.MOUNT_ICEBERG`. You could also specify them later by calling `SET_LOCAL_CREDENTIAL`.

```
BEGIN
  -- Create a new credential
  DBMS_CLOUD.CREATE_CREDENTIAL('BUCKET_CREDENTIAL', ...);

  -- Make the new credential play the role of the required
  'DATA_STORAGE_CREDENTIAL'
  DBMS_CATALOG.SET_LOCAL_CREDENTIAL(
    catalog_name => 'iceberg_cat',
    required_credential => 'DATA_STORAGE_CREDENTIAL',
    local_credential => 'BUCKET_CREDENTIAL');
END;
```

You can see the current mapping between *required* and *local* credentials with the `GET_LOCAL_CREDENTIAL_MAP` function.

```
SELECT
  JSON_QUERY(dbms_catalog.get_local_credential_map('iceberg_cat'), '$'
  PRETTY)
FROM dual;
  2      3
JSON_QUERY(DBMS_CATALOG.GET_LOCAL_CREDENTIAL_MAP('ICEBERG_CAT'),'$$' PRETTY)
```

```

-----
{
  "DATA_STORAGE_CREDENTIAL" : "BUCKET_CREDENTIAL" ,
  "ICEBERG_CREDENTIAL" : "UNITY_CRED"
}

```

- [Catalog Entity Management](#):
The subprograms listed here refers to managing various entities within a catalog, including creation, retrieval, and deletion of storage links, schemas, tables, and other catalog objects. It provides functions to:
 - Create and drop cloud storage links linked to catalogs.
 - Create and drop external tables within catalogs.
 - Create and drop schemas and their contents.
 - Retrieve lists of tables, schemas, and other catalog objects with optional filtering.
- [Data Access and Caching](#):
This section encompasses functions that handle querying data, managing result cursors, and controlling metadata caching within catalogs. It includes capabilities to:
 - Generate SELECT statements for logical tables with customizable options.
 - Open cursors to fetch data from logical tables.
 - Flush and prefill the metadata cache for catalogs to keep data up to date.
- [Synchronization and Sharing](#):
This section lists subprograms describes features that enable the management and maintenance of consistent, up-to-date catalogs and schemas across different systems or environments. It provides the following functionalities:
 - `CREATE_SYNCHRONIZED_SCHEMAS`: Creates local schemas on the database that are synchronized with remote catalog definitions, ensuring that the local environment reflects the remote source.
 - `DROP_SYNCHRONIZED_SCHEMAS`: Removes one or more schemas that were previously synchronized.
 - `UPDATE_SYNCHRONIZED_SCHEMA_PROPERTY`: Modifies properties of synchronized schemas, such as change tracking or synchronization behavior, by updating properties with new values in CLOB format.
 - `GRANT_TO_RECIPIENT` & `REVOKE_FROM_RECIPIENT`: Manage access permissions by granting or revoking access to catalogs for specific share recipients, controlling who can view or modify the catalog data.

Summary of DBMS_CATALOG

This section covers the DBMS_CATALOG subprograms provided with Autonomous AI Database.

Catalog Mounting and Unmounting

Subprogram	Description
mount_db_link	This procedure mounts a catalog based on a database link.
mount_shares	This procedure mounts multiple catalogs from a share provider, with overloads to support different input types.

Subprogram	Description
mount_share	This procedure mounts a single catalog based on a share.
mount_virtual_catalog	This procedure creates and mounts a new virtual catalog.
mount_data_catalog	This procedure mounts a data catalog such as AWS GLUE or OCI Data Catalog.
mount_iceberg	This procedure mounts an Iceberg catalog with specific configurations.
unmount	This procedure unmounts an existing external catalog.

Catalog Property Management

Subprogram	Description
get_catalog_property and get_catalog_property_clob	These procedures retrieves property values for a catalog.
update_catalog_property and update_catalog_property_clob	These procedures updates property values for a catalog.
get_schema_property and get_schema_property_clob	These procedures retrieves schema property values.
update_schema_property and update_schema_property_clob	These procedures updates schema property values.
get_table_property and get_table_property_clob	These procedures retrieves table property values.
update_table_property and update_table_property_clob	These procedures updates table property values.

Credential Management

Subprogram	Description
add_required_credential	This procedure defines a required credential for a catalog.
remove_required_credential	This procedure removes a required credential from a catalog.
rename_required_credential	This procedure renames a required credential.
update_required_credential_property	This procedure updates a property of a required credential.
get_required_credential_property	This procedure retrieves a property of a required credential.
set_local_credential	This procedure specifies a local credential for a required credential.
get_local_credential_map	This procedure retrieves mappings between required credentials and local credentials, with overloads for various output formats.

Catalog Entity Management

Subprogram	Description
create_cloud_storage_link and drop_cloud_storage_link	These procedure manages cloud storage links within a catalog.
create_external_table	This procedure creates an external table in a catalog.
drop_schema and drop_table	This procedure drops schemas and tables from a catalog.
get_tables , get_schemas and get_objects	This procedure retrieves metadata for tables, schemas, and objects within a catalog using pipelined functions.

Data Access and Caching

Subprogram	Description
generate_table_select	This procedure creates a <code>SELECT</code> statement for a logical table, with overloads for both procedure and function.
open_table_cursor	This procedure opens a cursor to return data from a logical table.
flush_catalog_cache and prefill_catalog_cache	This procedure manages catalog metadata caching.

Synchronization and Sharing

Subprogram	Description
create_synchronized_schemas and drop_synchronized_schemas	These procedure manages synchronized schemas with remote catalog definitions.
update_synchronized_schema_property	This procedure updates properties of synchronized schemas.
grant_to_recipient and revoke_from_recipient	These procedure manages access permissions for catalog recipients.

- [MOUNT_DB_LINK Procedure](#)
This procedure mounts an external catalog based on a database link.
- [MOUNT_SHARES Procedure](#)
This procedure mounts multiple catalogs based on shares from a share provider.
- [MOUNT_SHARE Procedure](#)
This procedure mounts a catalog based on a single share.
- [MOUNT_VIRTUAL_CATALOG Procedure](#)
This procedure creates and mounts a new virtual catalog.
- [MOUNT_DATA_CATALOG Procedure](#)
This procedure mounts a data catalog (e.g., AWS GLUE or OCI Data Catalog).
- [MOUNT_ICEBERG Procedure](#)
This procedure mounts an Iceberg catalog.
- [UNMOUNT Procedure](#)
This procedure unmounts an external catalog.

- [GET_CATALOG_PROPERTY Procedure](#)
This procedure returns the value of a property for a catalog.
- [GET_CATALOG_PROPERTY_CLOB Procedure](#)
This procedure returns the value of a property for a catalog in CLOB format.
- [UPDATE_CATALOG_PROPERTY Procedure](#)
This procedure updates the value of a property for a catalog.
- [UPDATE_CATALOG_PROPERTY_CLOB Procedure](#)
This procedure updates the value of a property for a catalog in CLOB.
- [UPDATE_TABLE_PROPERTY_CLOB Procedure](#)
This procedure updates the value of a property for a table in CLOB format.
- [GET_SCHEMA_PROPERTY Procedure](#)
This procedure returns the value of a property for a schema.
- [GET_SCHEMA_PROPERTY_CLOB Procedure](#)
This procedure returns the value of a property for a schema in CLOB format.
- [UPDATE_SCHEMA_PROPERTY Procedure](#)
This procedure returns the value of a property for a schema.
- [GET_TABLE_PROPERTY Procedure](#)
This procedure returns the value of a property for a table.
- [GET_TABLE_PROPERTY_CLOB Procedure](#)
This procedure returns the value of a property for a table in CLOB format.
- [UPDATE_TABLE_PROPERTY Procedure](#)
This procedure updates the value of a property for a table.
- [UPDATE_TABLE_PROPERTY_CLOB Procedure](#)
This procedure updates the value of a property for a table in CLOB format.
- [ADD_REQUIRED_CREDENTIAL Procedure](#)
This procedure defines a credential required for using the catalog.
- [REMOVE_REQUIRED_CREDENTIAL Procedure](#)
This procedure removes a required credential from a catalog.
- [RENAME_REQUIRED_CREDENTIAL Procedure](#)
This procedure renames a required credential from a catalog.
- [UPDATE_REQUIRED_CREDENTIAL_PROPERTY Procedure](#)
This procedure updates a property of a named required credential.
- [GET_REQUIRED_CREDENTIAL_PROPERTY Procedure](#)
This procedure retrieves a property of a named required credential.
- [SET_LOCAL_CREDENTIAL Procedure](#)
This procedure specifies a local credential to use when consuming a catalog.
- [GET_LOCAL_CREDENTIAL_MAP \(Overloaded\) Procedure](#)
This procedure retrieves a map of required credentials to local credentials.
- [CREATE_CLOUD_STORAGE_LINK Procedure](#)
This procedure creates a cloud storage link in a catalog.
- [DROP_CLOUD_STORAGE_LINK Procedure](#)
This procedure drops a cloud storage link from a catalog.
- [CREATE_EXTERNAL_TABLE Procedure](#)
This procedure creates an external table in a catalog.

- [DROP_SCHEMA Procedure](#)
This procedure drops a schema from a catalog along with all its contents.
- [DROP_TABLE Procedure](#)
This procedure drops a table from a catalog.
- [CREATE_SCHEMA Procedure](#)
This procedure creates a new schema in a catalog.
- [GET_TABLES Procedure](#)
This procedure fetches all tables for a catalog.
- [GET_SCHEMAS Procedure](#)
This procedure fetches all schemas for a catalog.
- [GET_OBJECTS Procedure](#)
This procedure fetches all objects for a catalog.
- [GENERATE_TABLE_SELECT \(Overloaded\) Procedure](#)
This procedure generates a SELECT statement for a logical table.
- [OPEN_TABLE_CURSOR Procedure](#)
This procedure opens a cursor that returns data from a logical table.
- [FLUSH_CATALOG_CACHE Procedure](#)
This procedure flushes the catalog metadata cache,
- [PREFILL_CATALOG_CACHE Procedure](#)
This procedure fills the cache for the specified catalog with current data.
- [CREATE_SYNCHRONIZED_SCHEMAS Procedure](#)
This procedure creates local database schemas synchronized with remote catalog definitions.
- [DROP_SYNCHRONIZED_SCHEMAS Procedure](#)
This procedure drops one or more synchronized schemas.
- [UPDATE_SYNCHRONIZED_SCHEMA_PROPERTY Procedure](#)
This procedure updates a property of synchronized schemas.
- [GRANT_TO_RECIPIENT Procedure](#)
This procedure grants access on a catalog to a specific share recipient.
- [REVOKE_FROM_RECIPIENT Procedure](#)
This procedure revokes access on a catalog from a specific recipient.

MOUNT_DB_LINK Procedure

This procedure mounts an external catalog based on a database link.

Syntax

```
PROCEDURE mount_db_link
(
  catalog_name      IN VARCHAR2,
  db_link           IN VARCHAR2,
  enabled           IN BOOLEAN := TRUE
);
```

Parameters

Parameter	Description
catalog_name	The name of the new catalog. This will be converted to uppercase.
db_link	The database link to use for mounting the catalog.
enabled	Indicates if the catalog is enabled for search. Defaults to TRUE.

MOUNT_SHARES Procedure

This procedure mounts multiple catalogs based on shares from a share provider.

Syntax

```
PROCEDURE mount_shares
(
  share_provider      IN VARCHAR2,
  shares              IN SYS.JSON_ARRAY_T,
  created_catalogs    IN OUT NOCOPY SYS.JSON_ARRAY_T,
  share_provider_owner IN VARCHAR2 := NULL
);
```

Parameters

Parameter	Description
share_provider:	The name of the share provider.
shares	A JSON array defining the list of shares to mount. Each element specifies properties like shareName, required, shareNameRule, catalogName, skipExisting, placeholder, and enabled.
created_catalogs	A JSON array of newly created catalogs, returned as output with properties like shareName and catalogName.
share_provider_owner	The owner of the share provider. Defaults to the current catalog if NULL.

Syntax of Second Procedure

```
PROCEDURE mount_shares
(
  share_provider      IN VARCHAR2,
  shares              IN SYS.JSON_ARRAY_T,
  share_provider_owner IN VARCHAR2 := NULL
);
```

Syntax of Third Procedure

```
PROCEDURE mount_shares
(
  share_provider      IN VARCHAR2,
  shares              IN CLOB := ' [{shareName:"%"} ]',
);
```

```

    share_provider_owner    IN VARCHAR2 := NULL
);

```

Parameters of the Second Procedure

Parameters are Identical to Procedure 1, except without `created_catalogs`.

Parameters of the Third Procedure

Parameters are Identical to Signature 1, except `shares` is a CLOB with a default value that includes all shares.

Example

```

/***** Share Catalog *****/
BEGIN
  -- Create a share provider
  dbms_share.create_or_replace_share_provider(
    provider_name    => 'MY_SHARE_PROVIDER',
    endpoint         => 'https://...');
  dbms_cloud.create_credential(
    credential_name  => 'SHARE_PROVIDER_CREDENTIAL', ...);
  dbms_share.set_share_provider_credential(
    provider_name    => 'MY_SHARE_PROVIDER',
    share_credential => 'SHARE_PROVIDER_CREDENTIAL');

  -- Mount shares as catalogs
  dbms_catalog.mount_shares(
    share_provider   => 'MY_SHARE_PROVIDER');
END;
/

```

MOUNT_SHARE Procedure

This procedure mounts a catalog based on a single share.

Syntax

```

PROCEDURE mount_share
(
  catalog_name           IN VARCHAR2,
  share_provider        IN VARCHAR2,
  share_name            IN VARCHAR2,
  share_provider_owner  IN VARCHAR2 := NULL,
  enabled               IN BOOLEAN := TRUE
);

```

Parameters

Parameter	Description
<code>catalog_name</code>	The name of the new catalog. This will be converted to uppercase.
<code>share_provider</code>	The name of the share provider.
<code>share_name</code>	The name of the share to mount.

Parameter	Description
share_provider_owner	The owner of the share provider. Defaults to the current catalog if NULL.
enabled	Indicates if the catalog is enabled for search. Defaults to TRUE.

Example

```

/***** Share Catalog *****/
BEGIN
  -- Create a share provider
  dbms_share.create_or_replace_share_provider(
    provider_name      => 'MY_SHARE_PROVIDER',
    endpoint           => 'https://...');
  dbms_cloud.create_credential(
    credential_name    => 'SHARE_PROVIDER_CREDENTIAL', ...);
  dbms_share.set_share_provider_credential(
    provider_name      => 'MY_SHARE_PROVIDER',
    share_credential   => 'SHARE_PROVIDER_CREDENTIAL');

  -- Mount a share catalog
  dbms_catalog.mount_share(
    catalog_name       => 'SHARE_CAT',
    share_provider     => 'MY_SHARE_PROVIDER',
    share_name         => 'MY_SHARE_1');
END;
/

```

MOUNT_VIRTUAL_CATALOG Procedure

This procedure creates and mounts a new virtual catalog.

Syntax

```

PROCEDURE mount_virtual_catalog
(
  catalog_name      IN VARCHAR2,
  enabled           IN BOOLEAN := TRUE
);

```

Parameters

Parameter	Description
catalog_name	The name of the new catalog. This will be converted to uppercase.
enabled	Indicates if the catalog is enabled for search. Defaults to TRUE.

MOUNT_DATA_CATALOG Procedure

This procedure mounts a data catalog (e.g., AWS GLUE or OCI Data Catalog).

Syntax

```
PROCEDURE mount_data_catalog
(
  catalog_name          IN VARCHAR2,
  data_catalog_type    IN VARCHAR2,
  data_catalog_region  IN VARCHAR2,
  data_catalog_credential IN VARCHAR2,
  data_catalog_id      IN VARCHAR2 := NULL,
  data_storage_credential IN VARCHAR2 := NULL,
  enabled              IN BOOLEAN := TRUE
);
```

Parameters

Parameter	Description
catalog_name	The name of the new catalog. This will be converted to uppercase.
data_catalog_type	Type of data catalog (e.g., OCI_DCAT or AWS_GLUE).
data_catalog_region	The cloud resource region of the data catalog.
data_catalog_credential	The local credential used to access the data catalog.
data_catalog_id	The data catalog name or identifier. Defaults to NULL.
data_storage_credential	It specifies the local credential for data access. Defaults to NULL. This field is optional.
enabled	Indicates if the catalog is enabled for search. Defaults to TRUE.

Example

```

/***** Glue Catalog *****/
BEGIN
  -- Create a credential capable of accessing a glue catalog
  dbms_cloud.create_credential('GLUE_CATALOG_CREDENTIAL', ...);

  -- Mount a glue catalog
  dbms_catalog.mount_data_catalog(
    catalog_name          => 'GLUE_CAT',
    data_catalog_type    => 'AWS_GLUE',
    data_catalog_region  => 'us-east-1',
    data_catalog_credential => 'GLUE_CATALOG_CREDENTIAL');
END;
/
```

MOUNT_ICEBERG Procedure

This procedure mounts an Iceberg catalog.

See [Iceberg REST Catalog Spec](#) for more information on Iceberg Catalog.

Syntax

```
PROCEDURE mount_iceberg
(
  catalog_name          IN VARCHAR2,
  endpoint              IN VARCHAR2,
  catalog_credential    IN VARCHAR2,
  data_storage_credential IN VARCHAR2,
  configuration         IN SYS.JSON_OBJECT_T := NULL,
  enabled              IN BOOLEAN := TRUE,
  catalog_type         IN VARCHAR2 := 'ICEBERG_GENERIC'
);
```

Parameters

Parameter	Description
catalog_name	The name of the new catalog. This will be converted to uppercase.
endpoint	The Iceberg base server URL.
catalog_credential	The credential used to access the Iceberg catalog.
data_storage_credential	The credential used to access the data.
configuration	This describes the catalog configuration as a JSON object. The following are supported configurations: <ul style="list-style-type: none"> namespacePath: Defaults to NULL. namespaceSeparator: Defaults to dot (.) for Unity, %1F for other Iceberg catalogs. isCaseSensitive: Defaults to FALSE for Unity, TRUE for other Iceberg catalogs) isPublicCatalog: Defaults to FALSE. bucketRegion: Defaults to NULL. It is relevant only if the catalog uses S3 storage.
enabled	This indicates if the catalog is enabled for search. Defaults to TRUE.
catalog_type	Type of Iceberg catalog (e.g., ICEBERG_GENERIC, ICEBERG_POLARIS, ICEBERG_UNITY). Defaults to ICEBERG_GENERIC.

Example

```
/****** Iceberg Catalog - Unity *****/
DECLARE
  config      JSON_OBJECT_T := JSON_OBJECT_T();
BEGIN
  -- Create a credential capable of accessing an external iceberg REST catalog
  dbms_share.create_bearer_token_credential(
    credential_name => 'ICEBERG_CATALOG_CREDENTIAL',
    token_endpoint => 'https://.../v1/tokens',
```

```

    client_id => ...,
    client_secret => ...,
    token_scope => 'all-apis');

-- Create a credential capable of accessing the bucket where the
-- iceberg data files are stored.
dbms_cloud.create_credential('ICEBERG_STORAGE_CRED', ...);

-- Mount a iceberg catalog
config.put('namespacePath', 'sales_catalog');
dbms_catalog.mount_iceberg(
    catalog_name          => 'ICEBERG_CAT',
    endpoint              => 'https://.../api/2.1/unity-catalog/iceberg/
v1',
    catalog_credential    => 'ICEBERG_CATALOG_CREDENTIAL',
    data_storage_credential => 'ICEBERG_STORAGE_CRED',
    configuration         => config,
    catalog_type          => 'ICEBERG_UNITY');
END;
/

/***** Iceberg Catalog - Polaris *****/
DECLARE
    config          JSON_OBJECT_T := JSON_OBJECT_T();
BEGIN
    -- Create a credential capable of accessing an external iceberg REST catalog
    dbms_share.create_bearer_token_credential(
        credential_name => 'ICEBERG_CATALOG_CREDENTIAL',
        token_endpoint => 'https://.../v1/oauth/tokens',
        client_id => ...,
        client_secret => ...,
        token_scope => 'PRINCIPAL_ROLE:ALL');

    -- Create a credential capable of accessing the bucket where the
    -- iceberg data files are stored.
    dbms_cloud.create_credential('ICEBERG_STORAGE_CRED', ...);

    -- Mount a iceberg catalog
    config.put('namespacePath', 'sales_catalog');
    dbms_catalog.mount_iceberg(
        catalog_name          => 'ICEBERG_CAT',
        endpoint              => 'https://.../polaris/api/catalog/v1',
        catalog_credential    => 'ICEBERG_CATALOG_CREDENTIAL',
        data_storage_credential => 'ICEBERG_STORAGE_CRED',
        configuration         => config,
        catalog_type          => 'ICEBERG_POLARIS');
END;
/

```

UNMOUNT Procedure

This procedure unmounts an external catalog.

Syntax

```

PROCEDURE unmount
(
  catalog_name IN VARCHAR2
);

```

Parameters

Parameter	Description
catalog_name	The name of an existing catalog to unmount. This will be converted to uppercase.

GET_CATALOG_PROPERTY Procedure

This procedure returns the value of a property for a catalog.

Syntax

```

FUNCTION get_catalog_property
(
  catalog_name      IN VARCHAR2,
  catalog_property  IN VARCHAR2
)
RETURN VARCHAR2;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
catalog_property	<p>This describes the the property to be retrieved.(for example, PROP_CACHE_DURATION, PROP_IS_ENABLED).</p> <p>The value can be any of the following constants:</p> <ul style="list-style-type: none"> • PROP_CACHE_ASYNC • PROP_CACHE_DURATION • PROP_CACHE_ENABLED • PROP_CONFIGURATION • PROP_CUSTOM • PROP_DCAT_TYPE • PROP_DEFAULT_SCHEMA • PROP_IS_ENABLED • PROP_METADATA <p>See Catalog Types and Constants for more information.</p>

GET_CATALOG_PROPERTY_CLOB Procedure

This procedure returns the value of a property for a catalog in CLOB format.

Syntax

```

FUNCTION get_catalog_property_clob
(
  catalog_name      IN VARCHAR2,
  catalog_property  IN VARCHAR2
)
RETURN CLOB;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
catalog_property	<p>This describes the the property to be retrieved (for example, PROP_CACHE_DURATION, PROP_IS_ENABLED).</p> <p>The value can be any of the following constants:</p> <ul style="list-style-type: none"> • PROP_CACHE_ASYNC • PROP_CACHE_DURATION • PROP_CACHE_ENABLED • PROP_CONFIGURATION • PROP_CUSTOM • PROP_DCAT_TYPE • PROP_DEFAULT_SCHEMA • PROP_IS_ENABLED • PROP_METADATA <p>See Catalog Types and Constants for more information.</p>

UPDATE_CATALOG_PROPERTY Procedure

This procedure updates the value of a property for a catalog.

Syntax

```

PROCEDURE update_catalog_property
(
  catalog_name      IN VARCHAR2,
  catalog_property  IN VARCHAR2,
  new_value         IN VARCHAR2
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.

Parameter	Description
catalog_property	This describes the property to be updated (for example, PROP_CACHE_DURATION, PROP_IS_ENABLED). The value can be any of the following constants: <ul style="list-style-type: none"> PROP_CACHE_ASYNC PROP_CACHE_DURATION PROP_CACHE_ENABLED PROP_CONFIGURATION PROP_CUSTOM PROP_DEFAULT_SCHEMA PROP_IS_ENABLED PROP_METADATA See Catalog Types and Constants for more information.
new_value	The new property value.

UPDATE_CATALOG_PROPERTY_CLOB Procedure

This procedure updates the value of a property for a catalog in CLOB.

Syntax

```
PROCEDURE update_catalog_property_clob
(
  catalog_name      IN VARCHAR2,
  catalog_property  IN VARCHAR2,
  new_value         IN CLOB
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
catalog_property	This describes the the property to be retrieved (for example, PROP_CACHE_DURATION, PROP_IS_ENABLED).
new_value	The new property value in CLOB.

UPDATE_TABLE_PROPERTY_CLOB Procedure

This procedure updates the value of a property for a table in CLOB format.

Syntax

```
PROCEDURE update_table_property_clob
(
  catalog_name      IN VARCHAR2,
  schema_name       IN VARCHAR2,
  table_name        IN VARCHAR2,
  table_property    IN VARCHAR2,
```

```

    new_value          IN CLOB
  );

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
table_name	The name of the table.
table_property	The property to update.
new_value	The new property value in CLOB.

GET_SCHEMA_PROPERTY Procedure

This procedure returns the value of a property for a schema.

Syntax

```

FUNCTION get_schema_property
(
  catalog_name          IN VARCHAR2,
  schema_name          IN VARCHAR2,
  schema_property      IN VARCHAR2
)

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
schema_property	This describes the property to be retrieved (for example, PROP_CUSTOM, PROP_METADATA).

GET_SCHEMA_PROPERTY_CLOB Procedure

This procedure returns the value of a property for a schema in CLOB format.

Syntax

```

FUNCTION get_schema_property_clob
(
  catalog_name          IN VARCHAR2,
  schema_name          IN VARCHAR2,
  schema_property      IN VARCHAR2
)
RETURN CLOB;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
schema_property	This describes the property to be retrieved (for example, PROP_CUSTOM, PROP_METADATA).

UPDATE_SCHEMA_PROPERTY Procedure

This procedure returns the value of a property for a schema.

Syntax

```
PROCEDURE update_schema_property
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  schema_property  IN VARCHAR2,
  new_value        IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
schema_property	This describes the property to be retrieved (for example, PROP_CUSTOM, PROP_METADATA).
new_value	The new property value.

GET_TABLE_PROPERTY Procedure

This procedure returns the value of a property for a table.

Syntax

```
FUNCTION get_table_property
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  table_property   IN VARCHAR2
)
RETURN VARCHAR2;
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
table_property	The property to retrieve.
table_name	The name of the table.

GET_TABLE_PROPERTY_CLOB Procedure

This procedure returns the value of a property for a table in CLOB format.

Syntax

```
FUNCTION get_table_property_clob
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  table_property   IN VARCHAR2
)
RETURN CLOB;
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
table_name	The name of the table.
table_property	The property to update (e.g., PROP_CUSTOM).

UPDATE_TABLE_PROPERTY Procedure

This procedure updates the value of a property for a table.

Syntax

```
PROCEDURE update_table_property
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  table_property   IN VARCHAR2,
  new_value        IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
table_name	The name of the table.
table_property	The property to update.
new_value	The new property value.

UPDATE_TABLE_PROPERTY_CLOB Procedure

This procedure updates the value of a property for a table in CLOB format.

Syntax

```
PROCEDURE update_table_property_clob
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  table_property   IN VARCHAR2,
  new_value        IN CLOB
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
table_name	The name of the table.
table_property	The property to update.
new_value	The new property value in CLOB.

ADD_REQUIRED_CREDENTIAL Procedure

This procedure defines a credential required for using the catalog.

Syntax

```
PROCEDURE add_required_credential
(
  catalog_name      IN VARCHAR2,
  required_credential IN VARCHAR2,
  local_credential  IN VARCHAR2 := NULL,
  credential_type   IN VARCHAR2 := NULL
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
required_credential	The name of the required credential.
local_credential	It is an optional local credential to use. Defaults to NULL.
credential_type	This field is optional. This parameter specifies the type of authentication mechanism being used to access external services, such as object storage or other databases. For example, oci, aws, azure or a slack_credential.

See [Catalog Types and Constants](#) for more information.

REMOVE_REQUIRED_CREDENTIAL Procedure

This procedure removes a required credential from a catalog.

Syntax

```
PROCEDURE remove_required_credential
(
  catalog_name          IN VARCHAR2,
  required_credential  IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
required_credential	The name of the required credential to remove.

RENAME_REQUIRED_CREDENTIAL Procedure

This procedure renames a required credential from a catalog.

Syntax

```
PROCEDURE rename_required_credential
(
  catalog_name          IN VARCHAR2,
  old_credential_name  IN VARCHAR2,
  new_credential_name  IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
old_credential_name	The old name of the required credential.

Parameter	Description
new_credential_name	The new name of the required credential.

UPDATE_REQUIRED_CREDENTIAL_PROPERTY Procedure

This procedure updates a property of a named required credential.

Syntax

```
PROCEDURE update_required_credential_property
(
  catalog_name          IN VARCHAR2,
  required_credential   IN VARCHAR2,
  credential_property   IN VARCHAR2,
  new_value             IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
required_credential	The name of the required credential.
credential_property	The property name (e.g., PROP_REQUIRED_CREDENTIAL_DESCRIPTION, PROP_REQUIRED_CREDENTIAL_VAULT_SECRET).
new_value	The new value for the property.

GET_REQUIRED_CREDENTIAL_PROPERTY Procedure

This procedure retrieves a property of a named required credential.

Syntax

```
FUNCTION get_required_credential_property
(
  catalog_name          IN VARCHAR2,
  required_credential   IN VARCHAR2,
  credential_property   IN VARCHAR2
)
RETURN VARCHAR2;
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
required_credential	The name of the required credential.

Parameter	Description
credential_property	The property name (e.g., PROP_REQUIRED_CREDENTIAL_DESCRIPTION, PROP_REQUIRED_CREDENTIAL_VAULT_SECRET).

SET_LOCAL_CREDENTIAL Procedure

This procedure specifies a local credential to use when consuming a catalog.

Syntax

```
PROCEDURE set_local_credential
(
  catalog_name          IN VARCHAR2,
  required_credential  IN VARCHAR2,
  local_credential     IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
required_credential	The name of the required credential.
local_credential	The credential to use.

GET_LOCAL_CREDENTIAL_MAP (Overloaded) Procedure

This procedure retrieves a map of required credentials to local credentials.

Syntax 1

```
PROCEDURE get_local_credential_map
(
  catalog_name          IN VARCHAR2,
  credential_map       OUT NOCOPY credential_info_map
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
credential_map	The populated map of credentials.

Syntax 2

```
FUNCTION get_local_credential_map
(
  catalog_name          IN VARCHAR2,
```

```

    show_errors      IN NUMBER := 0,
    pretty_json      IN NUMBER := 0
)
RETURN CLOB;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog.
show_errors	This displays error messages for invalid local credentials. Defaults to 0.
pretty_json	Format the JSON with the PRETTY option. Defaults to 0.

Syntax 3

```

PROCEDURE get_local_credential_map
(
    catalog_id      IN NUMBER,
    credential_map  OUT NOCOPY SYS.JSON_OBJECT_T
);

```

Parameters

Parameter	Description
catalog_id	The ID of an existing catalog from the CATALOG_ID column in ALL_MOUNTED_CATALOGS.
credential_map	The populated map as a JSON object.

CREATE_CLOUD_STORAGE_LINK Procedure

This procedure creates a cloud storage link in a catalog.

Syntax

```

PROCEDURE create_cloud_storage_link
(
    catalog_name      IN VARCHAR2,
    schema_name       IN VARCHAR2,
    storage_link_name IN VARCHAR2,
    uri               IN VARCHAR2,
    required_credential IN VARCHAR2 := NULL,
    enabled           IN BOOLEAN := TRUE,
    replace_if_exists IN BOOLEAN := FALSE
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name:	The schema name.

Parameter	Description
storage_link_name	The storage link name.
uri	The bucket URI.
required_credential	The required credential (or NULL for public buckets). Defaults to NULL.
enabled	Indicates if the storage link is enabled for search. Defaults to TRUE.
replace_if_exists	Replace the link if it already exists. Defaults to FALSE.

DROP_CLOUD_STORAGE_LINK Procedure

This procedure drops a cloud storage link from a catalog.

Syntax

```
PROCEDURE drop_cloud_storage_link
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  storage_link_name IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name:	The schema name.
storage_link_name	The storage link name.

CREATE_EXTERNAL_TABLE Procedure

This procedure creates an external table in a catalog.

Syntax

```
PROCEDURE create_external_table
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  storage_link_name IN VARCHAR2,
  file_uri_list    IN CLOB,
  column_list      IN CLOB := NULL,
  field_list       IN CLOB := NULL,
  table_format     IN CLOB := NULL,
  credential_name  IN VARCHAR2 := NULL,
  replace_if_exists IN BOOLEAN := FALSE
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The schema name.
table_name	The table name.
storage_link_name	The storage link name.
file_uri_list	The relative file list as a CLOB.
column_list	Optional column list. Defaults to NULL.
field_list	Optional field list. Defaults to NULL.
table_format	Optional format information. Defaults to NULL.
credential_name	The required credential. Defaults to NULL.
replace_if_exists	Replace the table if it already exists. Defaults to FALSE.

DROP_SCHEMA Procedure

This procedure drops a schema from a catalog along with all its contents.

Syntax

```
PROCEDURE drop_schema
(
  catalog_name    IN VARCHAR2,
  schema_name     IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name:	The schema name to drop.

DROP_TABLE Procedure

This procedure drops a table from a catalog.

Syntax

```
PROCEDURE drop_table
(
  catalog_name    IN VARCHAR2,
  schema_name     IN VARCHAR2,
  table_name      IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The schema name.
table_name	The table name.

CREATE_SCHEMA Procedure

This procedure creates a new schema in a catalog.

Syntax

```
PROCEDURE create_schema
(
  catalog_name    IN VARCHAR2,
  schema_name     IN VARCHAR2
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The schema name.

GET_TABLES Procedure

This procedure fetches all tables for a catalog.

Syntax

```
FUNCTION get_tables
(
  catalog_name    IN VARCHAR2,
  schema_name     IN VARCHAR2 := NULL,
  table_name      IN VARCHAR2 := NULL,
  conditions      IN VARCHAR2 := NULL,
  flags           IN NUMBER := 0,
  result_limit    IN NUMBER := NULL,
  column_flags    IN NUMBER := 0
)
RETURN catalog_tables PIPELINED;
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	Optional schema name. Defaults to NULL.

Parameter	Description
table_name	Optional table name. Defaults to NULL.
conditions	Optional additional conditions using lineage syntax. Defaults to NULL.
result_limit	Optional result limit. Defaults to NULL.
column_flags	Optional column flags (e.g., FETCH_PROPERTIES, FETCH_METADATA). Defaults to 0.

Note

- RECORD types describe structured data with named fields in databases.
- You have to query the database's information schema, system catalogs, or use database introspection tools to see the structure of RECORD types.

GET_SCHEMAS Procedure

This procedure fetches all schemas for a catalog.

Syntax

```
FUNCTION get_schemas
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2 := NULL,
  conditions        IN VARCHAR2 := NULL,
  flags            IN NUMBER := 0,
  result_limit      IN NUMBER := NULL,
  column_flags     IN NUMBER := 0
)
RETURN catalog_schemas PIPELINED;
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	Optional schema name. Defaults to NULL.
conditions	Optional additional conditions using lineage syntax. Defaults to NULL.
result_limit	Optional result limit. Defaults to NULL.
column_flags	Optional column flags (e.g., FETCH_PROPERTIES, FETCH_METADATA). Defaults to 0.

GET_OBJECTS Procedure

This procedure fetches all objects for a catalog.

Syntax

```

FUNCTION get_objects
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2 := NULL,
  object_name      IN VARCHAR2 := NULL,
  conditions      IN VARCHAR2 := NULL,
  flags            IN NUMBER := 0,
  result_limit     IN NUMBER := NULL
)
RETURN catalog_objects PIPELINED;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	Optional schema name. Defaults to NULL.
object_name	Optional object name. Defaults to NULL.
conditions	Optional additional conditions using lineage syntax. Defaults to NULL.
result_limit	Optional result limit. Defaults to NULL.

GENERATE_TABLE_SELECT (Overloaded) Procedure

This procedure generates a SELECT statement for a logical table.

Syntax

```

PROCEDURE generate_table_select
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name      IN VARCHAR2,
  stmt            IN OUT NOCOPY CLOB,
  options          IN CLOB := NULL,
  prev_gen_sql     IN CLOB := NULL,
  prev_ts         IN TIMESTAMP WITH TIME ZONE := NULL
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The schema name.

Parameter	Description
table_name	The table name.
stmt	The generated SELECT statement as output.
options	Generation options as a JSON object (e.g., distinct, rowLimit, select, orderBy). Defaults to NULL.
prev_gen_sql	The previously generated SELECT statement, if known. Defaults to NULL.
prev_ts	The previous generation timestamp, if known. Defaults to NULL.

Syntax

```

FUNCTION generate_table_select
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  options          IN CLOB := NULL,
  prev_gen_sql     IN CLOB := NULL,
  prev_ts         IN TIMESTAMP WITH TIME ZONE := NULL
)
RETURN CLOB;

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The schema name.
table_name	The table name.
stmt	The generated SELECT statement as output.
options	Generation options as a JSON object (e.g., distinct, rowLimit, select, orderBy). Defaults to NULL.
prev_gen_sql	The previously generated SELECT statement, if known. Defaults to NULL.
prev_ts	The previous generation timestamp, if known. Defaults to NULL.

OPEN_TABLE_CURSOR Procedure

This procedure opens a cursor that returns data from a logical table.

Syntax

```

PROCEDURE open_table_cursor
(
  catalog_name      IN VARCHAR2,
  schema_name      IN VARCHAR2,
  table_name       IN VARCHAR2,
  table_cursor     IN OUT NOCOPY SYS_REFCURSOR,
  options          IN CLOB := NULL
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	Optional schema name. Defaults to NULL.
table_name	Optional table name. Defaults to NULL.
table_cursor	The cursor to return data.
options	Generates options as a JSON object (e.g., rowLimit, select, orderBy). Defaults to NULL.

FLUSH_CATALOG_CACHE Procedure

This procedure flushes the catalog metadata cache,

Syntax

```
PROCEDURE flush_catalog_cache
(
  catalog_name      IN VARCHAR2,
  auto_commit       IN BOOLEAN := TRUE
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
auto_commit	Indicates if changes should be committed automatically. Defaults to TRUE.

PREFILL_CATALOG_CACHE Procedure

This procedure fills the cache for the specified catalog with current data.

Syntax

```
PROCEDURE prefill_catalog_cache
(
  catalog_name      IN VARCHAR2,
  schema_name       IN VARCHAR2 := NULL,
  auto_commit       IN BOOLEAN := TRUE
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	The name of the schema.
auto_commit	Indicates if changes should be committed automatically. Defaults to TRUE.

CREATE_SYNCHRONIZED_SCHEMAS Procedure

This procedure creates local database schemas synchronized with remote catalog definitions.

Syntax

```

PROCEDURE create_synchronized_schemas
(
  catalog_name IN VARCHAR2,
  schema_names IN VARCHAR2 := NULL,
  restrictions IN CLOB := NULL
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	A comma-delimited list of schema names (quoted or unquoted). Defaults to NULL to synchronize all schemas.
restrictions	JSON document specifying the scope of catalog entities. Defaults to NULL.

DROP_SYNCHRONIZED_SCHEMAS Procedure

This procedure drops one or more synchronized schemas.

Syntax

```

PROCEDURE drop_synchronized_schemas
(
  catalog_name IN VARCHAR2,
  schema_names IN VARCHAR2 := NULL,
  restrictions IN CLOB := NULL
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
schema_name	A comma-delimited list of schema names (quoted or unquoted). Defaults to NULL to synchronize all schemas.
restrictions	JSON document specifying the scope of catalog entities. Defaults to NULL.

UPDATE_SYNCHRONIZED_SCHEMA_PROPERTY Procedure

This procedure updates a property of synchronized schemas.

Syntax

```

PROCEDURE update_synchronized_schema_property
(
  catalog_name      IN VARCHAR2,
  property_name    IN VARCHAR2,
  new_value        IN CLOB
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
property_name	The property name to update.
new_value	The new value as a CLOB.

GRANT_TO_RECIPIENT Procedure

This procedure grants access on a catalog to a specific share recipient.

Syntax

```

PROCEDURE grant_to_recipient
(
  catalog_name      IN VARCHAR2,
  recipient_name    IN VARCHAR2
);

```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
recipient_name	The name of the share recipient (created by DBMS_SHARE.CREATE_SHARE_RECIPIENT).

REVOKE_FROM_RECIPIENT Procedure

This procedure revokes access on a catalog from a specific recipient.

Syntax

```

PROCEDURE revoke_from_recipient
(
  catalog_name      IN VARCHAR2,
  recipient_name    IN VARCHAR2,

```

```
owner          IN VARCHAR2 := NULL  
);
```

Parameters

Parameter	Description
catalog_name	The name of the existing catalog. This will be converted to uppercase.
recipient_name	The name of the share recipient (created by DBMS_SHARE.CREATE_SHARE_RECIPIENT).
owner	The owner of both catalog and recipient. Defaults to NULL.

The Data Transforms Page

Data Transforms allows you to design data transformations in the form of data loads, data flows, and workflows, without requiring you to write any code.

Data loads provide a convenient way of loading data into Autonomous Database, data flows define how the data is moved and transformed between different systems, while the workflows define the sequence in which the data flows are executed.

After your data flows and workflows are ready, you can execute the mappings immediately or schedule to execute them at a later time. Oracle Data Transforms run-time agent orchestrates the execution of jobs. On execution, Oracle Data Transforms generates the code for you.

You can launch Data Transforms in any of the following ways:

- **Oracle Cloud Marketplace:** Create a Data Transforms instance from Oracle Cloud Marketplace. Data Transforms is available as a separate listing on Marketplace called Data Integrator: Web Edition.
- **Database Actions Data Studio:** Navigate to Database Actions Data Studio Page, and click **Data Transforms** in the Database Actions page. If you have already registered a Data Transforms instance from OCI Marketplace with the Autonomous Database, the **Data Transforms** card on the Database Actions page will continue to take you to your Marketplace instance.

Note

If you do not see the Data Transforms card then your database user is missing the required `DATA_TRANSFORM_USER` role.

Access to the standard set of Data Transforms features may depend on where you launch Data Transforms from. In this documentation, certain topics could include any of the following badges to indicate features that may or may not be available for use:

- APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.
- APPLIES TO:  Data Transforms instance that is registered with Autonomous Database.
- APPLIES TO:  Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.
- [Access Oracle Data Transforms From Data Studio](#)
- [Data Transforms Notes](#)
- [Work with Connections](#)
Connections help you to connect Data Transforms to various technologies reachable from your OCI network.
- [Create and Run Data Loads](#)
A data load allows you to load multiple data entities from a source connection to a target connection.

- [Work with Data Entities](#)
A Data Entity is a tabular representation of a data structure.
- [Work with Projects](#)
A project is the top-level container, which can include multiple folders to organize your data flows or work flows into logical groups.
- [About Data Flows](#)
A data flow defines how the data is moved and transformed between different systems.
- [Schedule Data Flows or Workflows](#)
You can schedule workflows and data flows to run at specified time interval.
- [Monitor Status of Data Loads, Data Flows, and Workflows](#)
When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.
- [Introduction to Workflows](#)
A workflow is made up of multiple flows organized in a sequence in which they must be executed.
- [Create and Use Variables](#)
A variable is an object that stores a single value, which can be a string, a number or a date.
- [Use Machine Learning \(ML\) Models](#)
Data Transforms supports the use of ML Model in a data flow. Learn how to create and use Machine Learning (ML) Models in data flows.
- [Create and Manage Jobs](#)
When you execute a data load, data flow or workflow Oracle Data Transforms creates a job to complete the process in the background.
- [Export and Import Objects](#)
You can move Data Transforms objects between environments. You require an Object Storage connection to store the exported file. You then need to use the same Object Storage connection to import the objects.
- [Reference](#)
Get additional information about using Oracle Data Transforms on an Autonomous Database.

Access Oracle Data Transforms From Data Studio

APPLIES TO:  Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

Data Transforms combines all the elements of data integration - data movement, data synchronization, data quality, and data management to ensure that information is timely, accurate, and consistent across complex systems.

To use the Data Transforms tool you must access Database Actions as the ADMIN user or have the DATA_TRANSFORM_USER role assigned. See Manage User Profiles with Autonomous Database for information on granting roles.

To access the Data Transforms tool:

1. Login as an ADMIN user or a user with the DATA_TRANSFORM_USER role assigned.
2. Click **Data Transforms** in the Database Actions page, or click the Selector icon and select **Data Transforms** from the Data Studio menu in the navigation pane.

Note

Data Transforms is also available as a separate listing on OCI Marketplace. If you have already registered a Data Transforms instance from OCI Marketplace with the Autonomous Database, the Data Transforms card on the Database Actions page will continue to take you to your Marketplace instance. If you wish to use the embedded Data Transforms, then you must unregister the Marketplace instance. See [Unregister the ODI Instance from Autonomous Database](#).

3. When you login to the Data Transforms tool for the first time, you need to provide the database user credentials to sign in.
It takes approximately 1-2 minutes for the Data Transforms service to start. After the service starts, the Data Transforms home page opens.
4. From the left pane of the Home page, click the **Connections** tab to view the newly created Autonomous Database connection.
4. Click the **Actions** icon next to the connection and select **Edit**.
5. In the **Update Connection** page, enter the database username and password to use the connection. Click **Update** to save the changes.

Data Transforms Notes

APPLIES TO:  Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

Notes for using Data Transforms.

- **Enable access to private data sources from Autonomous Database:** If your Autonomous Database is configured to use a Private Endpoint, then you can only access private data sources from clients in the same Virtual Cloud Network (VCN). See [Enable Access to Private Data Sources from Autonomous Database](#) for more information.
Data Transforms jobs stuck in the Running status: If there are any Data Transforms jobs that are stuck in the Running status for an interminably long time, either stop the job immediately or delete the job and then rerun it. You may want to do this to avoid unwanted usage of resources for your tenancy. If the issue persists, file a service request at [Oracle Cloud Support](#) or contact your support representative.

Work with Connections

Connections help you to connect Data Transforms to various technologies reachable from your OCI network.

This section describes the generic steps to create a connection. The displayed connection detail options may vary depending on the selected connection type.

Apart from the connection types listed in [Supported Connection Types](#) you can create custom connectors, which you can use to connect Data Transforms to any JDBC supported data sources. See [Create Custom Connectors](#).

To create a new connection:

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.
2. Click **Create Connection**.
The **Create Connection** page slides in.

3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the type of connection that you wish to create.
 - Databases - Allows you to configure any connection type for supported database types.
 - Applications - Allows you to configure any connection type for supported applications.
 - Services - Allows you to configure any connection type for supported services.
4. After selecting the required connection type, click **Next**. The **Connection Name** field is pre-populated with a default name. You can edit this value.
5. For **Connection Details**, provide the connection details for the selected type such as:
 - Connection -
 - JDBC URL - The URL to connect to the data server. For example:
`jdbc:weblogic:sqlserver://hostname:port[;property=value[;...]]`

Note

If you are accessing Data Transforms from an Autonomous Database on a public endpoint, then the allowed port numbers are 2484, 1521-1525, 1708, 3306, 80.

For connectors that use an autoREST driver that provides the model files along with the driver, specify the servername and other properties required to connect to that datasource. For example:

```
jdbc:weblogic:autoREST://servername[;property=value[;...]]
```

- User - The user name, if required, for connecting to the server.
- Password - The password for connecting to server.
- Advanced Options
 - Array Fetch Size - When reading large volumes of data from a data server, Oracle Data Transforms fetches successive batches of records. This value is the number of rows (records read) requested by Oracle Data Transforms on each communication with the data server.
 - Batch Update Size - When writing large volumes of data into a data server, Oracle Data Transforms pushes successive batches of records. This value is the number of rows (records written) in a single Oracle Data Transforms INSERT command.

Note

Set Batch Update Size to 1000 or less for loading tables with BLOB data type columns.

- Degree of Parallelism for Target - This value indicates the number of threads allowed for a loading task. The default value is 1. The maximum number of threads allowed is 99.

Note

Connection details are specific and the above options vary based on the selected connection type. For the default connection that is created during provisioning, only the User and Password fields are editable. All the other fields are disabled.

- After providing all the required connection details, click **Test Connection** to test the connection.
If the test connection fails, do one of the following:
 - Check whether the Autonomous Database from where you are accessing Data Transforms is configured to use a private endpoint. See [Enable Access to Private Data Sources from Autonomous Database](#) for more information.
 - If a connection between Autonomous Databases on a Private Endpoint fails with a "Mismatch with server cert DN" error, check whether you have specified the `ssl_server_dn_match=yes` property in the JDBC URL. See [Troubleshoot Mismatch with Server Cert DN Error](#) for more information.
- Click **Create**.
The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- Name** of the connection.
- Technology** associated with the created connection.
- [Supported Connection Types](#)
This topic lists the connection types that are supported for connecting to Data Transforms.
- [Create Custom Connectors](#)
- [Create a Data Transforms Connection for Remote Data Load](#)
You can connect to an existing Data Transforms instance and run a data load remotely.
- [Create an Apache Iceberg Connection](#)
Apache Iceberg is an open standard table format that is optimized to manage large analytic datasets. Data Transforms supports the use of Apache Iceberg as a target to load data from any SQL-based data sources.

- [Create a Delta Share Connection](#)
Databricks Delta Share is an open protocol for secure data sharing. Oracle Data Transforms integrates with Delta Share to load data to Oracle Autonomous Database. You can use the Delta Share connection to load data from Databricks or Oracle Data Share.
- [Create an Oracle Business Intelligence Cloud Connector Connection](#)
Oracle Business Intelligence Cloud Connector (BICC) allows you to extract business data from a data source and load it into Autonomous Database.
- [Create and use an Oracle Cloud Infrastructure Generative AI Connection](#)
Oracle Cloud Infrastructure (OCI) Generative AI enables organizations to automate text summarization and dynamic content generation. Data Transforms integrates with OCI Generative AI to support the use of embedding vectors in a data flow.
- [Create an Oracle Enterprise Resource Planning Cloud Connection](#)
- [Create an Oracle Financials Cloud Connection](#)
You can fetch real time transactional data from Oracle Financials Cloud REST endpoints, import the data entities into Data Transforms, and use them as a source in a data flow.
- [Create and Use an Oracle NetSuite Connection](#)
You can use the Oracle NetSuite JDBC Driver or OAuth 2.0 authentication to connect to the Oracle NetSuite application. For Oracle NetSuite connections, Data Transforms allows you to load pre-built dataflows and workflows that you can run to transfer data from NetSuite to your target schema.
- [Create an Oracle Object Storage Connection](#)
You can use Data Transforms to upload data from Oracle Object Storage to Autonomous Database.
- [Create a REST Server Connection](#)

Supported Connection Types

This topic lists the connection types that are supported for connecting to Data Transforms.

Government cloud regions are required to support FIPS compliance standards for data protection. Data Transforms in Government realms is FIPS 140-2 Level 1 compliant. Make sure the connections sources are FIPS compliant to ensure secure communication between the servers.

Note

APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

- For the connectors that require driver installation, you need to copy the jar files to the `/u01/oracle/transforms_home/userlibs` directory before you add the connection.
- Apart from the connection types listed here you can create custom connectors, which you can use to connect Data Transforms to any JDBC supported data sources. See [Create Custom Connectors](#).

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				l	d			
				e	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

Aha!	Applic ation	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> • UrlParameter: jdbc:weblogic:aha:servername;AuthenticationMethod=UrlParameter; SecurityToken=security_token;[property=value[;...]]; • OAuth2: jdbc:weblogic:aha:servername;AuthenticationMethod=OAuth2; AccessToken=access_token;[property=value[;...]]; jdbc:weblogic:aha:servername;AuthenticationMethod=OAuth2;oauthCode=auth_code; ClientID=client_id;ClientSecret=client_secret;RedirectURI=redirect_uri; [property=value[;...]]; • method: jdbc:weblogic:aha:servername;ProxyHost=proxy_host;ProxyPassword=proxy_password; ; ProxyPort=proxy_port;ProxyUser=proxy_user;AuthenticationMethod=method; [property=value[;...]]; 	
------	-----------------	---------	---------	---------	--------	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Ahrefs	Applic	Ye	Ye	Ye	N	•	URLParameter: jdbc:weblogic:autorest:ServerName=https://apiv2.ahrefs.com; AuthenticationMethod=URLParameter; SecurityToken=xxxxxxx;AuthParam=token;	
	ation	s	s	s	o		• jdbc:weblogic:autorest://servername; [property=value[;...]];	
							• jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Amazon Aurora	Database	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> userIDPassword: jdbc:weblogic:postgresql:// servername:port; AuthenticationMethod=userIDPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] kerberos: jdbc:weblogic:postgresql:// servername:port; AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName; [property=value[;...]]; User: [Ignored] Password: [Ignored] EntralDPassword: jdbc:weblogic:postgresql:// servername:port; AuthenticationMethod=EntralDPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] EntralDServicePrincipal: jdbc:weblogic:postgresql:// 	
---------------	----------	-----	-----	-----	-----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

servername:port;
AuthenticationMethod=EntraIDServicePrin
cipal;
ServicePrincipal=your_service_principal
;Secret=your_client_secret;
[property=value[...]];
User: [Mandatory]
Password: [Mandatory]

```

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Amazon EMR Hive	Datab ase	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> none: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[;...]];User: [Ignored]Password: [Ignored] userIdPassword: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[;...]];User: [Mandatory]Password: [Mandatory] kerberos: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[;...]];User: [Ignored]Password: [Ignored] delegationToken: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;De 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
				o	G			
				n	ol			
				o	de			
				m	n			
				o	G			
				us	at			
				D	e			
				at				
				ab				
				as				
				e				
							legationToken=delegationToken; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
Amazon Redshift	Database	Yes	Yes	Yes	Yes	•	userIdPassword: jdbc:weblogic:redshift://server_name:port_number;DatabaseName=database_name;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] <ul style="list-style-type: none"> • Proxy server: jdbc:weblogic:redshift://server_name:port_number;DatabaseName=database_name;ProxyHost=proxy_host;ProxyPassword=proxy_password;ProxyPort=proxy_port;ProxyUser=proxy_user;[property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Apach e Hive	Datab ase	Ye s	Ye s	Ye s	N o	•	none: jdbc:weblogic:hive:// servername:port;DatabaseName=database; AuthenticationMethod=none; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
						•	userIdPassword: jdbc:weblogic:hive:// servername:port;DatabaseName=database; AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:hive:// servername:port;DatabaseName=database; AuthenticationMethod=kerberos;ServicePr incipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
						•	delegationToken: jdbc:weblogic:hive:// servername:port;DatabaseName=database; AuthenticationMethod=delegationToken;De	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
						legationToken=delegationToken;		
						[property=value[;...]];		
						User: [Ignored]		
						Password: [Ignored]		
Apach e Iceber g	Datab ase	Ye s	Ye s	Ye s	Y e s	Authentication modes supported are None, Simple, and OAuth2.0.	http://<host>:<port>/iceberg	For information about creating a connection using Apache Iceberg, see Create an Apache Iceberg Connection.

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			a					
			s					
			e					

Apach e Impala	Datab ase	Ye s	Ye s	Ye s	N o	•	<p>none: jdbc:weblogic:impala:// servername:port;DatabaseName=databaseNa me; AuthenticationMethod=none; [property=value[;...]]; User: [Ignored] Password: [Ignored]</p> <p>• userIdPassword: jdbc:weblogic:impala:// servername:port;DatabaseName=databaseNa me; AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• kerberos: jdbc:weblogic:impala:// servername:port;DatabaseName=databaseNa me; AuthenticationMethod=kerberos;ServicePr incipleName=serviceprincipalname; [property=value[;...]]; User: [Ignored] Password: [Ignored]</p> <p>• Proxy server: jdbc:weblogic:impala:// servername:port;DatabaseName=database_n</p>	
----------------------	--------------	---------	---------	---------	--------	---	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							ame;ProxyHost=proxy_host; ProxyPassword=proxy_password;ProxyPort= proxy_port;ProxyUser=proxy_user; [property=value[;...]];	
Apach e Spark SQL	Datab ase	Ye s	Ye s	Ye s	N o	•	userIdPassword: jdbc:weblogic:sparksq:// servername:port;DatabaseName=database; AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:sparksq:// servername:port;DatabaseName=database; AuthenticationMethod=kerberos;ServicePr incipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
						•	jdbc:weblogic:sparksq:// servername:port;DatabaseName=database; TransportMode=http; [property=value[;...]]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
AWS S3	Datab ase	Ye s	Ye s	Ye s	N o	•	AWS: jdbc:weblogic:autorest:AuthenticationMethod=AWS;servicename=s3;accesskey=*****;secretkey=*****; • jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	
Azure Billing	Applic ation	Ye s	Ye s	Ye s	N o	•	jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	
Azure Comp ute	Datab ase	Ye s	Ye s	Ye s	N o	•	jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			o	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Azure Data Lake Storage	Datab ase	Ye s	Ye s	Ye s	N o	• OAuth2:	<pre> jdbc:weblogic:autorest:ServerName=https:// {accountName}.dfs.core.windows.net; AuthenticationMethod=OAuth2;ClientID=xx xxxxxx;ClientSecret=xxxxxxx;RefreshTok en=xxxxxxx; Scope=https:// storage.azure.com/user_impersonation offline_access; TokenURI=https:// login.microsoftonline.com/{tenantID}/ oauth2/v2.0/token; AuthorizationURI=https:// login.microsoftonline.com/{tenantID}/ oauth2/v2.0/authorize; RedirectURI: http://localhost; </pre> <ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]]; 	
-------------------------	-----------	------	------	------	-----	------------------	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			o	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			a					
			s					
			e					
Azure Reserved VM Instances	Database	Yes	Yes	Yes	No	•	jdbc:weblogic:autorest://servername; [property=value[;...]];	<ul style="list-style-type: none"> • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];
Azure Resource Health	Application	Yes	Yes	Yes	No	•	OAuth2: jdbc:weblogic:autorest:ServerName=https://management.azure.com; AuthenticationMethod=OAuth2; TenantId=xxxxxxxx; ClientID=xxxxxxxx; ClientSecret=xxxxxxxx; AccessToken=xxxxxxxx; RefreshToken=xxxxxxxx; Scope=https://management.azure.com/.defaultoffline_access; TokenURI=https://login.microsoftonline.com/{tenantId}/oauth2/token; RedirectURI=http://localhost/;	<ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			a					
			s					
			e					
Azure SQL Database	Database	Yes	Yes	Yes	Yes	•	userIdPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;Authenticati onMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	activeDirectoryPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;Authenticati onMethod=activeDirectoryPasswo rd;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	ActiveDirectoryServicePrincipal: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;Authenticati onMethod=ActiveDirectoryServic ePrincipal;ServicePrincipal=clientID;Sec ret=clientSecret; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

- **auto:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto; [property=value[...]]; User: [Optional] Password: [Optional]
- **kerberos:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name; [property=value[...]]; User: [Ignored] Password: [Ignored]
- **ntlm:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm; [property=value[...]]; User: [Mandatory] Password: [Mandatory]
- **ntlmjava:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;Aut

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
						henticationMethod=ntlmjava;		
						[property=value[;...]];		
						User: [Mandatory]		
						Password: [Mandatory]		
						• ntlm2java: jdbc:weblogic:sqlserver://		
						hostname:port;DatabaseName=database;Aut		
						henticationMethod=ntlm2java;		
						[property=value[;...]];		
						User: [Mandatory]		
						Password: [Mandatory]		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Azure Synapse Analytics	Database	Yes	Yes	Yes	Yes	•	userIdPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;AuthentificationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	activeDirectoryPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;AuthentificationMethod=activeDirectoryPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	ActiveDirectoryServicePrincipal: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;AuthentificationMethod=ActiveDirectoryServicePrincipal;ServicePrincipal=clientID;Secret=clientSecret; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

- **auto:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto; [property=value[...]]; User: [Optional] Password: [Optional]
- **kerberos:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name; [property=value[...]]; User: [Ignored] Password: [Ignored]
- **ntlm:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm; [property=value[...]]; User: [Mandatory] Password: [Mandatory]
- **ntlmjava:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;Aut

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							henticationMethod=ntlmjava; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
							<ul style="list-style-type: none"> ntlm2java: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database;Aut henticationMethod=ntlm2java; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] 	
BigCo	Applic	Ye	Ye	Ye	N		HTTPHeader: jdbc:weblogic:autoarest:servername=https ://api.bigcommerce.com/stores/ {store_hash};AuthenticationMethod=HTTPH eader;securitytoken=*****;auth header=X-Auth-Token;	
mmmerc	ation	s	s	s	o		<ul style="list-style-type: none"> jdbc:weblogic:autoarest://servername; [property=value[;...]]; jdbc:weblogic:autoarest:Sample=sample_pa th;[property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			a					
			s					
			e					

Cassandra	Database	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> none: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=none; [property=value[...]]; User: [Ignored] Password: [Ignored] userIdPassword: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory] kerberos: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[...]]; User: [Ignored] Password: [Ignored]
-----------	----------	-----	-----	-----	-----	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				o	l			
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

Cloud ra CDH Hive	Datab ase	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> none: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=none; [property=value[;...]]; User: [Ignored] Password: [Ignored] userIdPassword: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] kerberos: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=kerberos;ServicePri ncipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] delegationToken: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=delegationToken;Del 	
----------------------------	--------------	---------	---------	---------	--------	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							egationToken=delegationToken; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
Confluence Cloud	Database	Yes	Yes	Yes	No	• Basic:	jdbc:weblogic:autorest:servername=https://<your-domain.atlassian.net>;authenticationmethod=Basic;	
						•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Data Transforms	Service	Yes	Yes	Yes	No		http://<host>:<port>/odi-rest	For instructions on connecting to an existing Data Transforms instance, see Create a Data Transforms Connection for Remote Data Load .

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	Tr	Tr	O			
		t	an	an	p			
		g	sf	sf	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	Cl				
			o	G				
			n	o	l			
			o	d	e			
			m	n				
			o	G				
			u	s	a			
			D	e				
			a					
			a					
			s					
			e					
DataStax	Application	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> • userIdPassword: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] • kerberos: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored] • none: jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace;AuthenticationMethod=none; [property=value[;...]]; User: [Ignored] Password: [Ignored] 	jdbc:weblogic:cassandra://server:port;KeyspaceName=keyspace; [property=value[;...]] where KeyspaceName specifies the default name of the Cassandra keyspace to which the driver connects.	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Delta Share	Application	Yes	Yes	Yes	No	Enter the Token Endpoint URL.		For instructions on creating a connection using Delta Share, see Create a Delta Share Connection
DocuSign	Database	Yes	Yes	Yes	No	<ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
eBay	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:ServerName=https://api.eBay.com;AuthenticationMethod=OAuth2;ClientID=NameHere-APIUserA-BCD-1234e5f6g-h7i8j890;ClientSecret=xxxxxxx;AccessToken=xxxxxxx;RefreshToken=xxxxxxx;Scope=https://api.eBay.com/oauth/api_scopeTokenURI=https://api.eBay.com/identity/v1/oauth2/tokenRedirectURI=http://localhost/;	
							• jdbc:weblogic:autorest://servername;[property=value[...]];	
							• jdbc:weblogic:autorest:Sample=sample_path;[property=value[...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			a					
			s					
			e					

EnterpriseDB	Database	Yes	Yes	Yes	Yes	•	userIDPassword: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName;[property=value[;...]]; User: [Ignored] Password: [Ignored]	
						•	EntralDPassword: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=EntralDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	EntralDServicePrincipal: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=EntralDServicePrincipal; ServicePrincipal=your_service_principal	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							;Secret=your_client_secret; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Financ	Applic	Ye	Ye	Ye	Y	•	jdbc:weblogic:sforce:// servername;SecurityToken=value[;propert y=value[;...]]	
ialForc	ation	s	s	s	e	•	userIDPassword: jdbc:weblogic:sforce:// servername;AuthenticationMethod=userIDP assword[;property=value[;...]];	
e					s	•	oauth2.0: jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; AccessToken=access-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; RefreshToken=refresh-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; JWTCertStore=file- path;ClaimsIssuer=client- ID;ClaimsSubject=user-name;	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Generic Rest	Application	Yes	Yes	Yes	No	Enter the REST Service URL.		For information about connecting to any REST service endpoint to create a connection, see Create a REST Server Connection .
Generic Rest Config	Application	Yes	No	No	No	<ul style="list-style-type: none"> jdbc:weblogic:autorest://servername;[property=value[;...]]; jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 	For information about connecting to any REST service endpoint to create a connection, see Create a REST Server Connection .	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

GitHub	Applic ation	Ye s	Ye s	Ye s	N o	•	<pre>jdbc:weblogic:github:ServerName=server_ name;SecurityToken=security_token; [property=value[;...]];</pre> <ul style="list-style-type: none"> • Basic: <pre>jdbc:weblogic:github:ServerName=server_ name;AuthenticationMethod=Basic; [property=value[;...]];</pre> • OAuth2: <pre>jdbc:weblogic:github:ServerName=server_ name;AuthenticationMethod=OAuth2; AccessToken=access_token; [property=value[;...]];</pre> <pre>jdbc:weblogic:github:ServerName=server_ name;ProxyHost=proxy_host;ProxyPassword =proxy_password; ProxyPort=proxy_port;ProxyUser=proxy_us er;AuthenticationMethod=OAuth2; AccessToken=access_token; [property=value[;...]];</pre> 	
--------	-----------------	---------	---------	---------	--------	---	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
				o	G			
				n	ol			
				o	de			
				m	n			
				o	G			
				us	at			
				D	e			
				at				
				ab				
				as				
				e				
Googl e Ads	Applic ation	Ye s	No	No	D	Depends on the driver		Requires driver installation
					e			
					p			
					e			
					n			
					d			
					s			
					o			
					n			
					th			
					e			
					d			
					ri			
					v			
					e			
					r			

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

Googl e AdSen se	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=https://adsense.googleapis.com/v2;clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com;clientSecret=*****;authUri=https://accounts.google.com/o/oauth2/auth;tokenUri=https://oauth2.googleapis.com/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=https://www.googleapis.com/auth/adsense;refreshToken=*****;	
						•	jdbc:weblogic:autorest://servername;[property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Googl e Analyti cs	Applic ation	Ye s	Ye s	Ye s	N o	•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	
						•	OAuth2: jdbc:weblogic:googleanalytics4:AddTable s='{myTableDefinitionString}'; ClientID=clientID;ClientSecret=client_s ecret;AuthenticationMethod=OAuth2; RefreshToken=refresh_token;Scope=scope; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	t			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	i	i				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Googl e BigQu ery	Datab ase	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> oauth2: jdbc:weblogic:googlebigquery:Project=pr oject;Dataset=dataset; AccessToken=accesstoken;RefreshToken=re freshtoken;ClientID=clientid; ClientSecret=clientsecret;Authenticatio nMethod=oauth2;[property=value[;...]]; User: [Ignored] Password: [Ignored] serviceaccount: jdbc:weblogic:googlebigquery:Project=pr oject;Dataset=dataset; ServiceAccountEmail=serviceAccountEmail ; ServiceAccountKeyContent=private_key_co ntent; AuthenticationMethod=serviceaccount; [property=value[;...]]; User: [Ignored] Password: [Ignored] 	
----------------------------	--------------	---------	---------	---------	--------	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Googl e Calend ar	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=https://www.googleapis.com/calendar/v3;clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com;clientSecret=*****;authUri=https://accounts.google.com/o/oauth2/auth; tokenUri=https://accounts.google.com/o/oauth2/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=https://www.googleapis.com/auth/calendar https://www.googleapis.com/auth/calendar.readonly https://www.googleapis.com/auth/calendar.eventshttps://www.googleapis.com/auth/calendar.events.readonly https://www.googleapis.com/auth/calendar.settings.readonly https://www.googleapis.com/auth/calendar.addons.execute;	
----------------------------	-----------------	---------	---------	---------	--------	---	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							authuri=https://accounts.google.com/o/oauth2/auth; tokenuri=https://accounts.google.com/o/oauth2/token; refreshtoken=*****;	
							<ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]]; 	
Google Campaign Manager	Application	Yes	Yes	Yes	No		<ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Googl e Contac ts	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=people.googleapis.com/v1; clientId=123456789012- abc123def456ghi789jkl012mno345pq.apps.g oogleusercontent.com; clientSecret=*****; authUri=https://accounts.google.com/o/ oauth2/auth; tokenUri=https:// accounts.google.com/o/oauth2/token; redirectUri=http://localhost; AuthenticationMethod=OAuth2; scope=https://www.googleapis.com/auth/ contacts.other.readonly https:// www.googleapis.com/auth/ contactshttps://www.googleapis.com/ auth/contacts.readonly https:// www.googleapis.com/auth/ directory.readonlyhttps:// www.googleapis.com/auth/ profile.agerange.read https:// www.googleapis.com/auth/ profile.emails.readhttps://	
----------------------------	-----------------	---------	---------	---------	--------	---	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

www.googleapis.com/auth/
profile.language.read https://
www.googleapis.com/auth/
user.addresses.readhttps://
www.googleapis.com/auth/
user.birthday.read https://
www.googleapis.com/auth/
user.emails.readhttps://
www.googleapis.com/auth/
user.gender.read https://
www.googleapis.com/auth/
user.organization.readhttps://
www.googleapis.com/auth/
user.phonenumbers.read https://
www.googleapis.com/auth/
userinfo.emailhttps://
www.googleapis.com/auth/
userinfo.profile; refreshtoken:
*****;

```

- jdbc:weblogic:autorest://servername; [property=value[;...]];
- jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

jdbc:weblogic:autorest:servername=people.googleapis.com/vl; clientId=123456789012-abc123def456ghi789jkl012mno345pq.apps.googleusercontent.com;
clientSecret=*****;
authUri=https://accounts.google.com/o/oauth2/auth; tokenUri=https://accounts.google.com/o/oauth2/token;
redirectUri=http://localhost;
AuthenticationMethod=OAuth2; scope=https://www.googleapis.com/auth/contacts.other.readonly https://www.googleapis.com/auth/contactshttps://www.googleapis.com/auth/contacts.readonly https://www.googleapis.com/auth/directory.readonlyhttps://www.googleapis.com/auth/profile.agerange.read https://www.googleapis.com/auth/profile.emails.readhttps://www.googleapis.com/auth/profile.language.read https://www.googleapis.com/auth/user.addresses.readhttps://
    
```

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

www.googleapis.com/auth/user.birthday.read
https://www.googleapis.com/auth/
user.emails.readhttps://www.googleapis.com/
auth/user.gender.read https://
www.googleapis.com/auth/
user.organization.readhttps://
www.googleapis.com/auth/
user.phonenumbers.read https://
www.googleapis.com/auth/
userinfo.emailhttps://www.googleapis.com/
auth/userinfo.profile; refreshtoken:
*****;
    
```

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a	t			
				a	b			
				a	s			
				e				
Googl e Drive	Datab ase	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=https://www.googleapis.com/drive/v3;clientId=123456789012-abc123def456ghi789jkl.apps.googleusercontent.com;clientSecret=*****;authUri=https://accounts.google.com/o/oauth2/auth;tokenUri=https://accounts.google.com/o/oauth2/token;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=https://www.googleapis.com/auth/drive; refreshToken=*****;	
						•	jdbc:weblogic:autorest://servername;[property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		r	r	r	r			
		t	t	t	t			
		d	d	d	d			
		i	i	i	i	W		
		D	D	D	D	ri		
		a	a	a	a	t		
		In	Tr	Tr	Tr	O		
		te	an	an	an	p		
		gr	sf	sf	sf	e		
		at	or	or	or	r		
		or	m	m	m	a		
		:	s	s	s	ti		
		W	b	b	b	o		
		eb	uil	uil	uil	n		
		E	t	t	t			
		di	i	i	i			
		ti	to	to	to			
		o	A	O	O			
		n	ut	Cl	Cl			
			o	G	G			
			n	ol	ol			
			o	de	de			
			m	n	n			
			o	G	G			
			us	at	at			
			D	e	e			
			at					
			ab					
			as					
			e					

Googl e Search Ads 360	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=https://www.googleapis.com/doubleclicksearch/; clientId:372185870949-v1f1h94mvnad4hkajr4tonlia51vtj2n.apps.googleusercontent.com; clientSecret:*****; authDomain: https://accounts.google.com/o/oauth2/auth; tokenUri: https://oauth2.googleapis.com/token; redirectUri: http://localhost;AuthenticationMethod:OAuth2; scope: https://www.googleapis.com/auth/doubleclicksearch; refreshtoken:*****;	
							• jdbc:weblogic:autorest://servername; [property=value[;...]];	
							• jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Green plum	Datab ase	Ye s	Ye s	Ye s	N o	•	userIdPassword: jdbc:weblogic:greenplum:// servername:port;KeyspaceName=keyspace; AuthenticationMethod=userIdPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:greenplum:// servername:port;DatabaseName=databaseNa me;AuthenticationMethod=kerberos;Servic ePrincipal=servicePrincipal; [property=value[...]]; User: [Ignored] Password: [Ignored]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				l	d			
				e				
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

Horton works Hive	Datab ase	Ye s	Ye s	Ye s	N o	•	<p>none: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=none; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• userIdPassword: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=userIdPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• kerberos: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=kerberos;ServicePri ncipal=servicePrincipal; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• delegationToken: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=delegationToken;Del</p>	
-------------------	-----------	------	------	------	-----	---	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							egationToken=delegationToken; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
HubSpot	Application	Yes	Yes	Yes	No	OAuth2:	jdbc:weblogic:hubspot:AuthenticationMethod=OAuth2;ClientID=client_id;ClientSecret=client_secret;RefreshToken=refresh_token;Scope=scope; [property=value[;...]];	
Hypersonic SQL	Database	Yes	Yes	Yes	Yes		jdbc:hsqldb:<protocol>:<databasePath>[;ifexists=true][;<property>=<value>...];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

IBM BigInsi ghts	Datab ase	Ye s	Ye s	Ye s	N o	•	<p>none: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=none; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• userIdPassword: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=userIdPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• kerberos: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=kerberos;ServicePri ncipal=servicePrincipal; [property=value[...]]; User: [Mandatory] Password: [Mandatory]</p> <p>• delegationToken: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=delegationToken;Del</p>	
------------------------	--------------	---------	---------	---------	--------	---	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							egationToken=delegationToken; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			a					
			s					
			e					

IBM DB2 Hosted	Datab ase	Ye s	Ye s	Ye s	Y s	•	cleartext: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=cleartext; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	client: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=client; [property=value[...]]; User: [Ignored] Password: [Ignored]	
						•	encryptedPassword: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=encryptedPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	encryptedPasswordAES: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

uthenticationMethod=encryptedPasswordAES;
S:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **encryptedUIDPassword:**

jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=encryptedUIDPasswor
d:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **encryptedUIDPasswordAES:**

jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=encryptedUIDPasswor
dAES:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **kerberos:** jdbc:weblogic:db2://

servername:port;DatabaseName=database;A
uthenticationMethod=kerberos;ServicePri
ncipal=servicePrincipal;
[property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

User: [Ignored]

Password: [Ignored]

- **pluginSecurity:** jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=pluginSecurity;Plug
inName=pluginName;
[property=value[;...]];

User: [Ignored]

Password: [Ignored]

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			a					
			s					
			e					
IBM DB2 UDB	Datab ase	Ye s	Ye s	Ye s	Y s	•	cleartext: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=cleartext; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	client: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=client; [property=value[...]]; User: [Ignored] Password: [Ignored]	
						•	encryptedPassword: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=encryptedPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	encryptedPasswordAES: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

uthenticationMethod=encryptedPasswordAES;[property=value[;...]];
User: [Mandatory]
Password: [Mandatory]

- **encryptedUIDPassword:**
jdbc:weblogic:db2://
servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPassword;[property=value[;...]];
User: [Mandatory]
Password: [Mandatory]
- **encryptedUIDPasswordAES:**
jdbc:weblogic:db2://
servername:port;DatabaseName=database;AuthenticationMethod=encryptedUIDPasswordAES;[property=value[;...]];
User: [Mandatory]
Password: [Mandatory]
- **kerberos:** jdbc:weblogic:db2://
servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;
[property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

User: [Ignored]

Password: [Ignored]

- **pluginSecurity:** jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=pluginSecurity;Plug
inName=pluginName;
[property=value[;...]];

User: [Ignored]

Password: [Ignored]

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	t			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	i	i				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
IBM DB2 Warehouse	Datab ase	Ye s	Ye s	Ye s	Y s	•	cleartext: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=cleartext; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	client: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=client; [property=value[...]]; User: [Ignored] Password: [Ignored]	
						•	encryptedPassword: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=encryptedPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	encryptedPasswordAES: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

authenticationMethod=encryptedPasswordAES;
S:[property=value[;...]];
User: [Mandatory]
Password: [Mandatory]

```

- encryptedUIDPassword:**
 jdbc:weblogic:db2://
 servername:port;DatabaseName=database;A
 uthenticationMethod=encryptedUIDPasswor
 d:[property=value[;...]];
 User: [Mandatory]
 Password: [Mandatory]
- encryptedUIDPasswordAES:**
 jdbc:weblogic:db2://
 servername:port;DatabaseName=database;A
 uthenticationMethod=encryptedUIDPasswor
 dAES:[property=value[;...]];
 User: [Mandatory]
 Password: [Mandatory]
- kerberos:** jdbc:weblogic:db2://
 servername:port;DatabaseName=database;A
 uthenticationMethod=kerberos;ServicePri
 ncipal=servicePrincipal;
 [property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

User: [Ignored]

Password: [Ignored]

- **pluginSecurity:** jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=pluginSecurity;Plug
inName=pluginName;
[property=value[;...]];

User: [Ignored]

Password: [Ignored]

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			a					
			s					
			e					
IBM DB2/400	Datab ase	Ye s	Ye s	Ye s	Y s	•	cleartext: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=cleartext; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	client: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=client; [property=value[...]]; User: [Ignored] Password: [Ignored]	
						•	encryptedPassword: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A uthenticationMethod=encryptedPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	encryptedPasswordAES: jdbc:weblogic:db2:// servername:port;DatabaseName=database;A	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

uthenticationMethod=encryptedPasswordAES;
S:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **encryptedUIDPassword:**

jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=encryptedUIDPasswor
d:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **encryptedUIDPasswordAES:**

jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=encryptedUIDPasswor
dAES:[property=value[;...]];
User: [Mandatory]

Password: [Mandatory]

- **kerberos:** jdbc:weblogic:db2://

servername:port;DatabaseName=database;A
uthenticationMethod=kerberos;ServicePri
ncipal=servicePrincipal;
[property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

User: [Ignored]

Password: [Ignored]

- **pluginSecurity:** jdbc:weblogic:db2://
servername:port;DatabaseName=database;A
uthenticationMethod=pluginSecurity;Plug
inName=pluginName;

[property=value[;...]];

User: [Ignored]

Password: [Ignored]

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				l	d			
				e	m			
				n	o			
				G	u			
				s	a			
				t	t			
				D	e			
				a	t			
				a	b			
				a	s			
				e	e			
Informix	Database	Yes	Yes	Yes	No	•	jdbc:weblogic:informix:// servername:port;InformixServer=informix server; Database=databasename; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	jdbc:weblogic:informix:// servername:port;InformixServer=informix server; DatabaseName=databasename;AlternateServers=(alternateserver); ConnectionRetryCount=connectionretrycount; ConnectionRetryDelay=connectionretrydelay; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
Jira	Application	Yes	Yes	Yes	No		jdbc:weblogic:jira:servername;[property= value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Klaviyo	Applic	Ye	Ye	Ye	N	•	URLParameter: jdbc:weblogic:autorest:servername=https://a.klaviyo.com/api; AuthenticationMethod=URLParameter;securitytoken=****;authparam=api_key;	
	ation	s	s	s	o	•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Magen to	Applic ation	Ye s	No	No	D e p e n d s o n t h e d r i v e r	Depends on the driver		Requires driver installation
Mailchi mp	Applic ation	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> • BearerToken: jdbc:weblogic:autorest:servername=us20.api.mailchimp.com;AuthenticationMethod=BearerToken;securitytoken=*****; • jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				o	l			
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				
MapR Hive	Datab ase	Ye s	Ye s	Ye s	N o	•	none: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=none; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	userIdPassword: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=userIdPassword; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=kerberos;ServicePri ncipal=servicePrincipal; [property=value[...]]; User: [Mandatory] Password: [Mandatory]	
						•	delegationToken: jdbc:weblogic:hive:// servername:port;DatabaseName=database;A uthenticationMethod=delegationToken;Del	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							egationToken=delegationToken; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
Market	Applic	Ye	Ye	Ye	N	•	OAuth2: jdbc:weblogic:autoarest:servername=https:// ://{api_id}.mktorest.com; AuthenticationMethod=OAuth2;clientid=a1 234bc5-67d8-9e01- f23g-4h567ijk89l0;clientsecret=***** *****; tokenuri=https:// {api_id}.mktorest.com/identity/oauth/ token;	
o	ation	s	s	s	o		• jdbc:weblogic:autoarest://servername; [property=value[;...]]; • jdbc:weblogic:autoarest:Sample=sample_pa th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	t			
		d	d	d	d			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Micros oft Dynam ics 365	Applic ation	Ye s	Ye s	Ye s	Y e s	OAuth2:	<pre> jdbc:weblogic:dynamics365:ServiceURL=se rviceurl;AuthenticationMethod=OAuth2; AccessToken=access_token; [property=value[...]]; jdbc:weblogic:dynamics365:ServiceURL=se rviceurl;AuthenticationMethod=OAuth2; ClientID=client_id;ClientSecret=client_ secret; TokenURI=POSTtoken_uri;Scope=scope; [property=value[...]]; jdbc:weblogic:dynamics365:ServiceURL=se rviceurl;AuthenticationMethod=OAuth2; ClientID=client_id;ClientSecret=client_ secret;TokenURI=token_uri; RefreshToken=refresh_token; [property=value[...]]; jdbc:weblogic:dynamics365:ServiceURL=se rviceurl;ProxyHost=proxy_host; ProxyPassword=proxy_password;ProxyPort= proxy_port;ProxyUser=proxy_user; AuthenticationMethod=OAuth2;ClientID=cl ient_id;ClientSecret=client_secret; </pre>	
-----------------------------------	-----------------	---------	---------	---------	-------------	----------------	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							TokenURI=token_uri;RefreshToken=refresh_token;[property=value[;...]];	
							<ul style="list-style-type: none"> • NTLM: jdbc:weblogic:dynamics365:ServiceURL=serviceurl; AuthenticationMethod=NTLM;NTLMDomain=nTLMDomain;[property=value[;...]]; 	
Micros oft Share Point	Applic ation	Ye s	Ye s	Ye s	Y e s		<ul style="list-style-type: none"> • jdbc:weblogic:sharepoint:ServiceURL=serviceurl;ClientID=client_id;ClientSecret=client_secret; TokenURI=token_uri;RefreshToken=refresh_token;[property=value[;...]]; • jdbc:weblogic:sharepoint:ServiceURL=serviceurl;ProxyHost=proxy_host;ProxyPassword=proxy_password; ProxyPort=proxy_port;ProxyUser=proxy_user;ClientID=client_id;ClientSecret=client_secret; TokenURI=token_uri;RefreshToken=refresh_token;[property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Microsoft SQL Server	Database	Yes	Yes	Yes	Yes	•	userIdPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database; AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	activeDirectoryPassword: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database; AuthenticationMethod=activeDirectoryPas sword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	ActiveDirectoryServicePrincipal: jdbc:weblogic:sqlserver:// hostname:port;DatabaseName=database; AuthenticationMethod=ActiveDirectorySer vicePrincipal; ServicePrincipal=clientID;Secret=client Secret;[property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

- **auto:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=auto;[property=value[...]];User: [Optional]Password: [Optional]
- **kerberos:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipalName=service_principal_name;[property=value[...]];User: [Ignored]Password: [Ignored]
- **ntlm:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;AuthenticationMethod=ntlm;[property=value[...]];User: [Mandatory]Password: [Mandatory]
- **ntlmjava:** jdbc:weblogic:sqlserver://hostname:port;DatabaseName=database;

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

AuthenticationMethod=ntlmjava;
[property=value[;...]];
User: [Mandatory]
Password: [Mandatory]

```

- ntlm2java:** jdbc:weblogic:sqlserver://
 hostname:port;DatabaseName=database;
 AuthenticationMethod=ntlm2java;
 [property=value[;...]];
 User: [Mandatory]
 Password: [Mandatory]

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Mongo DB	Datab ase	Ye s	Ye s	Ye s	Y e s	•	<p>None: jdbc:weblogic:mongodb:// host:port;AuthenticationMethod=None;Dat abaseName=database; [property=value[;...]]; User: [Ignored] Password: [Ignored] jdbc:weblogic:mongodb:// host:port;AuthenticationMethod=None;Dat abaseName=database;ReplicaSetName=repli ca_set;[property=value[;...]];</p> <p>• auth_db: jdbc:weblogic:mongodb:// host:port;AuthenticationDatabase=auth_d b;DatabaseName=database; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] jdbc:weblogic:mongodb:// host:27017;AuthenticationDatabase=auth_ db;</p>	
----------	-----------	------	------	------	-------	---	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	Tr	Tr	O			
		t	an	an	p			
		g	sf	sf	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o	n			
		o	A	O	n			
		n	u	Cl	o			
			o	G	n			
			o	ol	o			
			o	de	m			
			o	n	o			
			u	s	a			
			D	e	t			
			a	t	a			
			a	b	a			
			s	a	s			
			e		e			

```
DatabaseName=database;EncryptionMethod=SSL;[property=value[;...]];
```

- **kerberos:** jdbc:weblogic:mongodb://host:port;AuthenticationMethod=kerberos;DatabaseName=database;ServicePrincipalName=principal_name;User=username;[property=value[;...]]; User: [Ignored] Password: [Ignored]
- **plain:** jdbc:weblogic:mongodb://host:port;AuthenticationMethod=plain;DatabaseName=database;[property=value[;...]]; User: [Mandatory] Password: [Mandatory]
- jdbc:weblogic:mongodb://host:port;CryptoProtocolVersion=protocol;DatabaseName=database;EncryptionMethod=[SSL | requestSSL];HostNameInCertificate=host_name;KeyPassword=key_password;KeyStore=key_store;TrustStore=trust_store;TrustStorePassword=ts_password;

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							ValidateServerCertificate=vsc_value; [property=value[;...]];	
							<ul style="list-style-type: none"> • jdbc:weblogic:mongodb:// host:port;DatabaseName=database; EncryptionMethod=SSL;ValidateServerCertificate=false;[property=value[;...]]; • jdbc:weblogic:mongodb:// host:port;AuthenticationDatabase=auth_d b;DatabaseName=database; ProxyHost=proxy_host;ProxyPassword=prox y_password;ProxyPort=proxy_port; ProxyUser=proxy_user; [property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
MySQL	Datab	Ye	Ye	Ye	Y	jdbc:mysql://<host>[:<port>]/[<database>]?		Make sure
L	ase	s	s	s	e	relaxAutoCommit=true&useCursorFetch=true[&<		that the
					s	property>=<value>...]		system
								variable
								property
								sql_require
								_primary_ke
								y is set to
								OFF.
								Otherwise, an
								ADW to
								MySQL
								mapping
								could fail with
								a "Table does
								not exist"
								error.

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
MySQL L Heatw ave	Datab ase	Ye s	Ye s	Ye s	Y s	jdbc:mysql://<host>[:<port>]/[<database>]? relaxAutoCommit=true&useCursorFetch=true[&< property>=<value>...]	If MySQL database is created with high availability, then write operation is not supported. Make sure that the system variable property sql_require _primary_ke y is set to OFF. Otherwise, an ADW to MySQL Heatwave mapping could fail with a "Table does	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
								not exist” error.
Netezz a	Datab ase	Ye s	No	No	D e p e n d s o n th e d r i v e r	Depends on the driver		Oracle Data Transforms uses the Netezza JDBC to connect to a NCR Netezza database. This driver must be installed in your Data Transforms userlibs directory. You can download the Netezza JDBC driver from the IBM website.

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle	Datab ase	Ye s	Ye s	Ye s	Y e s	• jdbc:oracle:thin:@<host>:<port>:<sid> • jdbc:oracle:thin:@<host>:<port>/ ServiceName> • jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS =(PROTOCOL=tcps)(HOST=<host>) (PORT=<port>)) (CONNECT_DATA=(SERVICE_NAME=<Service Name>)))	For Data Integrator Web Edition, write operation is supported only on Oracle cloud database targets. For details refer to the Oracle terms of use before deploying the image from OCI marketplace.	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle Analytics Cloud (Oracle BI Cloud Service)	Application	Yes	Yes	Yes	No		Enter the BI Cloud Connector Service URL.	
Oracle Business Intelligence Cloud (BICC) Connector	Application	Yes	Yes	Yes	No		Enter the BI Cloud Connector Service URL.	For information about creating a connection using Oracle Business Intelligence Cloud (BICC) Connector, see Create an Oracle Business Intelligence Cloud Connector Connection .

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle EBS	Application	Yes	Yes	Yes	Yes	• jdbc:oracle:thin:@<host>:<port>:<sid> • jdbc:oracle:thin:@<host>:<port>/<ServiceName> • jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host>)(PORT=<port>))(CONNECT_DATA=(SERVICE_NAME=<ServiceName>)))		
Oracle ERP Cloud	Application	Yes	Yes	No	No	Enter the URL of the BI Publisher web service.	For information about creating a connection using Oracle ERP Cloud, see Create an Oracle Enterprise Resource Planning Cloud Connection .	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle Financials Cloud	Application	Yes	Yes	Yes	No		Enter the REST Service URL.	For information about creating a connection using Oracle Financials Cloud, see Create an Oracle Financials Cloud Connection .
Oracle Fusion ERP	Application	Yes	Yes	Yes	No		Enter the Service URL.	
Oracle Fusion Sales	Application	Yes	Yes	Yes	No		Enter the Service URL.	
Oracle Fusion Service	Application	Yes	Yes	Yes	No		Oracle Fusion Service	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle GoldenGate – OCI	Service	Yes	Yes	Yes	Yes		Enter the deployment URL.	
Oracle Marketing Cloud	Application	Yes	Yes	Yes	Yes		<code>jdbc:weblogic:eloqua:Company=<company_id>;[;property=value[;...]]</code>	
Oracle NetSuite	Application	Yes	Yes	Yes	No		<code>jdbc:ns://{Server Host}:{Server Port};ServerDataSource={Server DataSource};</code>	For information about creating a connection using Oracle Netsuite, see Create and Use an Oracle NetSuite Connection .

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle Object Storage	Datab ase	Ye s	Ye s	Ye s	Y e s		Enter the Object Storage URL.	For information about creating a connection using Oracle Object Storage, see Create an Oracle Object Storage Connection .
Oracle PeopleSoft	Applic ation	Ye s	Ye s	Ye s	N o		<ul style="list-style-type: none"> jdbc:oracle:thin:@<host>:<port>:<sid> jdbc:oracle:thin:@<host>:<port/>ServiceName> jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host>)(PORT=<port>))(CONNECT_DATA=(SERVICE_NAME=<Service Name>))) 	
Oracle Sales Cloud	Applic ation	Ye s	Ye s	Ye s	N o		jdbc:weblogic:oraclesalescloud://<base_url>;WSCompressData=none	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle Service Cloud	Application	Yes	Yes	Yes	No	jdbc:weblogic:oracleservicecloud:loginHost=	jdbc:weblogic:oracleservicecloud:loginHost=host:[property=value[;...]]	When using multiple JDBC connections for Oracle Service Cloud in Oracle Data Transforms, ensure that each connection is uniquely identified to avoid conflict. To do this, add the DatabaseName property to the JDBC URL. For example, jdbc:weblogic:oracleservicecloud:loginHost=h

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```

ost;DatabaseName=<UniqueValue>;
where
<UniqueValue> is a unique value for each environment.
For example, DevConnection or ProdConnection.
This allows separate configuration files for each environment.

```

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Oracle SIEBEL	Application	Yes	Yes	Yes	No	•	jdbc:oracle:thin:@<host>:<port>:<sid>	
						•	jdbc:oracle:thin:@<host>:<port/ServiceName>	
						•	jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<host>)(PORT=<port>))(CONNECT_DATA=(SERVICE_NAME=<ServiceName>)))	
PayPal	Application	Yes	Yes	Yes	No	•	Basic: jdbc:weblogic:autorest:ServerName=https://api-sandbox.paypal.com;AuthenticationMethod=Basic;	
						•	jdbc:weblogic:autorest://servername;[property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			ab					
			as					
			e					
Pivotal HD	Database	Yes	Yes	Yes	No	<ul style="list-style-type: none"> none: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=none;[property=value[...]];User: [Mandatory]Password: [Mandatory] userIdPassword: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=userIdPassword;[property=value[...]];User: [Mandatory]Password: [Mandatory] kerberos: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal;[property=value[...]];User: [Mandatory]Password: [Mandatory] delegationToken: jdbc:weblogic:hive://servername:port;DatabaseName=database;AuthenticationMethod=delegationToken;Del 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	Tr	Tr	O			
		t	an	an	p			
		g	sf	sf	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			u	s	a			
			D	e				
			a					
			ab					
			as					
			e					
							egationToken=delegationToken; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
Pivotal HDB	Datab ase	Ye s	Ye s	Ye s	N o	•	userIdPassword: jdbc:weblogic:greenplum:// servername:port;KeyspaceName=keyspace; AuthenticationMethod=userIdPassword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	kerberos: jdbc:weblogic:greenplum:// servername:port;DatabaseName=databaseName; AuthenticationMethod=kerberos;ServicePrincipal=servicePrincipal; [property=value[;...]]; User: [Ignored] Password: [Ignored]	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			ab					
			as					
			e					

Postgr	Datab	Ye	Ye	Ye	Y	<ul style="list-style-type: none"> • userIDPassword: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=userIDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • kerberos: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=kerberos;ServicePrincipalName=servicePrincipalName; [property=value[;...]]; User: [Optional] Password: [Optional] • EntralDPassword: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=EntralDPassword;[property=value[;...]]; User: [Mandatory] Password: [Mandatory] • EntralDServicePrincipal: jdbc:weblogic:postgresql:// servername:port;AuthenticationMethod=EntralDServicePrincipal;ServicePrincipal= 	
eSQL	ase	s	s	s	e		
					s		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	t			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	i	i				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

```
servicePrincipal;Secret=secret;
[property=value[...]];
User: [Mandatory]
Password: [Mandatory]
```

Qmetr	Applic	Ye	Ye	Ye	N	•	URLParameter:
y	ation	s	s	s	o		jdbc:weblogic:autorest:servername=https://qtmcloud.qmetry.com;AuthenticationMethod=URLParameter;securitytoken=*****;authparam=apiKey;
							• jdbc:weblogic:autorest://servername;[property=value[...]];
							• jdbc:weblogic:autorest:Sample=sample_path;[property=value[...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Quick Books Online	Applic ation	Ye s	Ye s	Ye s	N o	<ul style="list-style-type: none"> • OAuth2: jdbc:weblogic:autorest:servername=https://sandbox-quickbooks.api.intuit.com/v3/company/+++; clientId: AB8oHGUGVpXQnqKehrhBrnnIEQ7vNa7YI4UOr6L EXlJa6dx0CL; clientSecret: *****; authUri: https://appcenter.intuit.com/connect/oauth2; tokenUri: https://oauth.platform.intuit.com/oauth2/v1/tokens/bearer; redirectUri: http://localhost; AuthenticationMethod: OAuth2; refreshtoken: *****; • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];
--------------------	--------------	------	------	------	-----	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				o	l			
				o	d			
				m	n			
				o	G			
				u	s			
				D	e			
				a				
				a				
				s				
				e				

Quick Books Payments	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> OAuth2: jdbc:weblogic:autorest:servername=https://sandbox.api.intuit.com;clientId=ABCd1efGH2Ijkl3mN40PQrsTu5vWxyZa6bCdeFGhIjk7LMnOpQ;clientSecret=*****; authUri=https://appcenter.intuit.com/connect/oauth2;tokenUri=https://oauth.platform.intuit.com/oauth2/v1/tokens/bearer;redirectUri=http://localhost;AuthenticationMethod=OAuth2;scope=com.intuit.quickbooks.payment;refreshToken=*****; jdbc:weblogic:autorest://servername;[property=value[;...]]; jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 	
----------------------	-------------	-----	-----	-----	----	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Quora Ads	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> OAuth2: jdbc:weblogic:autorest:ServerName=api.quora.com/ads;AuthenticationMethod=OAuth2;ClientID=xxxxxxx;ClientSecret=xxxxxxx;AccessToken=xxxxxxx;Scope=ads_read; jdbc:weblogic:autorest://servername;[property=value[;...]]; jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 	
-----------	-------------	-----	-----	-----	----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	Tr	Tr	O			
		t	an	an	p			
		g	sf	sf	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	i			
		E	t	t	n			
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			u	s	a			
			D	e				
			a					
			ab					
			as					
			e					
Sage	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=api.a ccounting.sage.com; clientId: *****; clientSecret: *****; authUri: https:// www.sageone.com/oauth2/auth/central? filter=apiv3.1; tokenUri: https:// oauth.accounting.sage.com/token, redirectUri: http://localhost; AuthenticationMethod: OAuth2; scope: full_access; refreshtoken: *****; • jdbc:weblogic:autorest://servername; [property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]]; 	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Salesf orce Chatte r	Applic ation	Ye s	Ye s	Ye s	N o	•	OAuth2: jdbc:weblogic:autorest:servername=progr esscom4-dev-ed.my.salesforce.com/ services/data/v52.0; AuthenticationMethod=OAuth2; clientid=3MVG9Sow8KERNN08IkuAFNRo0MZbip oL4zeEhQzdT4IrNnqW7_UBvmlK7vZ4Y_OzbNxFn 0.nJSfVBdqUuh4IL; clientsecret=*****; refreshtoken=*****; scope=full refresh_token; authuri=https:// login.salesforce.com/services/oauth2/ authorize; tokenuri=https:// login.salesforce.com/services/oauth2/ token;	
						•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_pa th; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		g	a	a	p			
		r	s	s	e			
		o	r	r	r			
		:	m	m	a			
		W	s	s	t			
		e	b	b	o			
		E	u	u	n			
		d	i	i				
		t	i	i				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	s				
			D	e				
			a					
			ab					
			as					
			e					
Salesf orce.c om	Applic ation	Ye s	Ye s	Ye s	N o	•	userIDPassword: jdbc:weblogic:sforce:// servername;AuthenticationMethod=userIDP assword; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] OAuth2: jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; AccessToken=access-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; RefreshToken=refresh-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2. 0;SchemaMap=schema-map-name; JWTCertStore=file- path;ClaimsIssuer=client- ID;ClaimsSubject=user-name; JWTCertPassword=certificate-	You cannot use a Salesforce connection as a target for mapping.

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
				n	ol			
				o	de			
				m	n			
				o	G			
				us	at			
				D	e			
				at				
				ab				
				as				
				e				
							password;JWTCertAlias=certificate-alias;[property=value[;...]];	
SAP BW/ 4HAN A	Datab ase	Ye s	Ye s	Ye s	N o	•	Basic: jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory]	
						•	HTTPHeader: jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored]	
						•	jdbc:weblogic:s4hana:ServerName=servername;ProxyHost=proxy_host; ProxyPassword=proxy_password;ProxyPort=proxy_port; ProxyUser=proxy_user; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

SAP HANA	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> Basic: jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] HTTPHeader: jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored] jdbc:weblogic:s4hana:ServerName=servername;ProxyHost=proxy_host; ProxyPassword=proxy_password;ProxyPort=proxy_port; ProxyUser=proxy_user; [property=value[;...]]; 	
----------	-------------	-----	-----	-----	----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

SAP NetWeaver	Database	Yes	Yes	Yes	No	<ul style="list-style-type: none"> • Basic: <code>jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic;</code> <code>[property=value[;...]];</code> User: [Mandatory] Password: [Mandatory] • HTTPHeader: <code>jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token;</code> <code>[property=value[;...]];</code> User: [Ignored] Password: [Ignored] • <code>jdbc:weblogic:s4hana:ServerName=servername;ProxyHost=proxy_host;</code> <code>ProxyPassword=proxy_password;ProxyPort=proxy_port; ProxyUser=proxy_user;</code> <code>[property=value[;...]];</code> 	
---------------	----------	-----	-----	-----	----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

SAP S/4HANA Cloud	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> Basic: jdbc:weblogic:s4hana:ServerName=servername;AuthenticationMethod=Basic; [property=value[;...]]; User: [Mandatory] Password: [Mandatory] HTTPHeader: jdbc:weblogic:s4hana:ServerName=servername;AccessToken=your_access_token;AuthenticationMethod=HTTPHeader;AuthHeader=api_key;SecurityToken=security_token; [property=value[;...]]; User: [Ignored] Password: [Ignored] jdbc:weblogic:s4hana:ServerName=servername;ProxyHost=proxy_host; ProxyPassword=proxy_password;ProxyPort=proxy_port;ProxyUser=proxy_user; [property=value[;...]]; 	
-------------------	-------------	-----	-----	-----	----	--	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Semrush	Application	Yes	Yes	Yes	No	•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];	
ServiceNow	Service	Yes	Yes	Yes	No	•	OAuth2: jdbc:weblogic:autorest:servername=instance.service-now.com/api; clientId=123a4b567c8901234567d8901e234fg5;clientSecret=*****; authUri=https://instance.service-now.com/oauth_auth.do; tokenUri=https://instance.service-now.com/ oauth_token.do;redirectUri=http://localhost; AuthenticationMethod=OAuth2;refreshToken=*****;	
						•	jdbc:weblogic:autorest://servername; [property=value[;...]];	
						•	jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Shopify	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> jdbc:weblogic:autorest://servername;[property=value[;...]]; jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Snowflake	Database	Yes	Yes	Yes	Yes	•	jdbc:weblogic:snowflake:AccountName=account_name;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;[property=value[;...]];	
						•	OAuth2: jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=OAuth2;AccessToken=access_token;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;[property=value[;...]];	
							jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=OAuth2;AuthURI=auth_uri;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;ClientID=client_id;ClientSecret=client_secret;RedirectURI=redirect_uri;[property=value[;...]];	
							jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=OAuth2;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			a					
			s					
			e					

ClientID=client_id;ClientSecret=client_secret;TokenURI=token_uri; [property=value[;...]];

jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=OAuth2; DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name; ClientID=client_id;ClientSecret=client_secret;TokenURI=token_uri;RefreshToken=refresh_token; [property=value[;...]];

- **BrowserBasedSSO:**
jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=BrowserBasedSSO; DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name; [property=value[;...]];
- **KeyPair:**
jdbc:weblogic:snowflake:AccountName=account_name;AuthenticationMethod=KeyPair; DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;PrivateKeyContent=private_key_content;

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i				
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
			n	o				
			o	d				
			m	n				
			o	G				
			u	a				
			D	e				
			a					
			a					
			s					
			e					

```
PrivateKeyPassphrase=password;
[property=value[;...]];
• jdbc:weblogic:snowflake:AccountName=account_name;DatabaseName=database_name;Schema=schema_name;Warehouse=warehouse_name;ProxyHost=proxy_host;ProxyPassword=proxy_password;ProxyPort=proxy_port;ProxyUser=proxy_user;[property=value[;...]];
```

Squar	Applic	Ye	Ye	Ye	N	•	BearerToken: jdbc:weblogic:autoREST:servername=sandbox.api.intuit.com; AuthenticationMethod: BearerToken; securitytoken=***** *;
e	ation	s	s	s	o	•	jdbc:weblogic:autoREST://servername; [property=value[;...]];
						•	jdbc:weblogic:autoREST:Sample=sample_path;[property=value[;...]];

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Stripe	Applic ation	Ye s	Ye s	Ye s	N o	BearerToken: jdbc:weblogic:autorest:servername=https:// api.stripe.com;AuthenticationMethod=BearerT oken;securitytoken=***** **;		
SAP Sybas e ASE	Datab ase	Ye s	Ye s	Ye s	Y es	jdbc:weblogic:sybase:// hostname:port[;property=value[;...]]		
SAP Sybas e IQ	Datab ase	Ye s	Ye s	Ye s	Y es	jdbc:weblogic:sybase:// hostname:port[;property=value[;...]]		
Sybas e As Anywh ere	Datab ase	Ye s	Ye s	Ye s	Y es	jdbc:weblogic:sybase:// hostname:port[;property=value[;...]]		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

TeamCity	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> jdbc:weblogic:teamcity:ServerName=server_name;[property=value[;...]]; BearerToken: jdbc:weblogic:teamcity:ServerName=server_name;AuthenticationMethod=BearerToken; SecurityToken=security_token; [property=value[;...]]; jdbc:weblogic:teamcity:ServerName=server_name;ProxyHost=proxy_host; ProxyPassword=proxy_password; ProxyPort=proxy_port; ProxyUser=proxy_user; AuthenticationMethod=BearerToken; SecurityToken=security_token; [property=value[;...]]; 	
----------	-------------	-----	-----	-----	----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Terada	Datab	Ye	No	No	D	Depends on the driver being used		Data
ta	ase	s			e	(com.ncr.teradata.TeraDriver)		Transforms
					p			uses the
					e			Teradata
					n			JDBC Driver
					d			to connect to
					s			a Teradata
					o			Database. To
					n			use Teradata
					th			as a data
					e			source the
					d			Teradata
					ri			Gateway for
					v			JDBC must be
					e			running, and
					r			this driver
								must be
								installed in
								your Data
								Transforms
								userlibs
								directory. You
								can download
								the JDBC
								driver from the

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Teradata
website.

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Terada ta 17+	Datab ase	Ye s	No	No	D e p e n d s o n th e d ri v e r	Depends on the driver being used (com.teradata.jdbc.TeraDriver)		Data Transforms uses the Teradata JDBC Driver to connect to a Teradata Database. To use Teradata as a data source the Teradata Gateway for JDBC must be running, and this driver must be installed in your Data Transforms userlibs directory. You can download the JDBC driver from the

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
								Teradata website..
Tumblr	Application	Yes	Yes	Yes	No	•	jdbc:weblogic:autorest://servername; [property=value[;...]]	
						•	jdbc:weblogic:autorest:Sample=sample_pa th:[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Veeva CRM	Application	Yes	Yes	Yes	Yes	<ul style="list-style-type: none"> jdbc:weblogic:sforce:// servername;SecurityToken=value[;property=value[;...]] userIDPassword: jdbc:weblogic:sforce:// servername;AuthenticationMethod=userIDPassword;[property=value[;...]]; oauth2.0: jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2.0;SchemaMap=schema-map-name;AccessToken=access-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2.0;SchemaMap=schema-map-name;RefreshToken=refresh-token; [property=value[;...]]; jdbc:weblogic:sforce:// servername;AuthenticationMethod=oauth2.0;SchemaMap=schema-map-name;JWTCertStore=file-path;ClaimsIssuer=client-ID;ClaimsSubject=user-name; 		

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
							JWTCertPassword=certificate- password;JWTCertAlias=certificate- alias;[property=value[;...]];	
Volusi on	Applic ation	Ye s	Ye s	Ye s	N o	•	jdbc:weblogic:autorest://servername; [property=value[;...]] • jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	
Wistia	Applic ation	Ye s	Ye s	Ye s	N o	•	Basic: jdbc:weblogic:autorest:servername=https :// api.wistia.com;authenticationmethod=Bas ic; • jdbc:weblogic:autorest://servername; [property=value[;...]] • jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
WooCommerce	Application	Yes	No	No	D	Depends on the driver.		Requires driver installation
					e			
					p			
					e			
					n			
					d			
					s			
					o			
					n			
					th			
					e			
					d			
					ri			
					v			
					e			
					r			

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

WordP	Applic	Ye	Ye	Ye	N	•	OAuth2:	
ress	ation	s	s	s	o		jdbc:weblogic:autoREST:servername=public-api.wordpress.com/rest/v1.1;AuthenticationMethod=OAuth2;clientid=***;clientsecret=*****;accesstoken=*****;	
							scope=global;oauthcode=*****;authuri=https://public-api.wordpress.com/oauth2/authorize; tokenuri=https://public-api.wordpress.com/oauth2/token;	
							• jdbc:weblogic:autoREST://servername;[property=value[;...]]	
							• jdbc:weblogic:autoREST:Sample=sample_path;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
				o	G			
				n	ol			
				o	de			
				m	n			
				o	G			
				us	at			
				D	e			
				at				
				ab				
				as				
				e				
Workd	Applic	Ye	No	No	D	Depends on the driver.		Requires
ay	ation	s			e			driver
					p			installation
					e			
					n			
					d			
					s			
					o			
					n			
					th			
					e			
					d			
					ri			
					v			
					e			
					r			
X	Applic	Ye	Ye	Ye	N	<ul style="list-style-type: none"> jdbc:weblogic:autorest://servername; [property=value[;...]] jdbc:weblogic:autorest:Sample=sample_path; [property=value[;...]]; 		
	ation	s	s	s	o			

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Xero	Applic ation	Ye s	Ye s	Ye s	N o	•	jdbc:weblogic:autorest://servername; [property=value[;...]]	
						•	jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	
Yelp	Applic ation	Ye s	Ye s	Ye s	N o	•	BearerToken: jdbc:weblogic:autorest:servername=https ://api.yelp.com/v3; AuthenticationMethod=BearerToken;securi tytoken=*****;	
						•	jdbc:weblogic:autorest://servername; [property=value[;...]]	
						•	jdbc:weblogic:autorest:Sample=sample_pa th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		or	or	or	o			
		te	te	te	rt			
		d	d	d	s			
		in	in	in	W			
		D	D	D	ri			
		at	at	at	t			
		a	a	a	e			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	ti			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	in	in				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					

Zende	Applic	Ye	Ye	Ye	N	•	OAuth2:	
sk	ation	s	s	s	o		jdbc:weblogic:autorest:servername=https	
							::/	
							example.zendesk.com;AuthenticationMetho	
							d=OAuth2;	
							clientid=client_id_zendesk;clientsecret	
							=*****;scope=readwrite;authuri=/	
							oauth/authorizations/new; tokenuri=/	
							oauth/tokens;	
							• Basic:	
							jdbc:weblogic:autorest:servername=https	
							::/	
							example.zendesk.com;authenticationmetho	
							d=Basic;	
							• jdbc:weblogic:autorest://servername;	
							[property=value[;...]];	
							• jdbc:weblogic:autorest:Sample=sample_pa	
							th;[property=value[;...]];	

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		I	T	T	O			
		t	a	a	p			
		g	s	s	e			
		a	o	o	r			
		o	m	m	a			
		:	s	s	t			
		W	b	b	o			
		e	u	u	n			
		E	t	t				
		d	i	i	n			
		t	o	o				
		o	A	O				
		n	u	C				
			o	G				
				n	o			
				l	d			
				e	n			
				o	G			
				u	s			
				D	e			
				a	t			
				a	b			
				a	s			
				e				

Zoho CRM	Application	Yes	Yes	Yes	No	<ul style="list-style-type: none"> OAuth2: jdbc:weblogic:autorest:AuthenticationMethod=OAuth2; clientid=a1234bc567d89e01f23g; clientsecret: *****; scope=ZohoCRM.settings.ALL,ZohoCRM.modules.ALL,ZohoCRM.notifications.ALL,ZohoCRM.org.READ,ZohoCRM.users.READ,ZohoCRM.settings.roles.READ,ZohoCRM.settings.profiles.READ,ZohoCRM.modules.Contacts.READ,ZohoSearch.securesearch.READ; authuri=https://accounts.zoho.com/oauth/v2/auth; tokenuri=https://accounts.zoho.com/oauth/v2/token; redirecturi=http://localhost; refreshToken=*****; • jdbc:weblogic:autorest://servername;[property=value[;...]]; jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 	
----------	-------------	-----	-----	-----	----	---	--

Name	Type	S	S	S	S	Authentication Mode:	Connection URL Example	Notes
		u	u	u	u			
		p	p	p	p			
		o	o	o	o			
		t	t	t	r			
		d	d	d	s			
		i	i	i	W			
		D	D	D	r			
		a	a	a	t			
		In	Tr	Tr	O			
		te	an	an	p			
		gr	sf	sf	e			
		at	or	or	r			
		or	m	m	a			
		:	s	s	t			
		W	b	b	o			
		eb	uil	uil	n			
		E	t	t				
		di	i	i				
		ti	to	to				
		o	A	O				
		n	ut	Cl				
			o	G				
			n	ol				
			o	de				
			m	n				
			o	G				
			us	at				
			D	e				
			at					
			ab					
			as					
			e					
Zoom	Applic ation	Ye s	Ye s	Ye s	N o	• OAuth2:	<pre> jdbc:weblogic:autorest:servername=https://api.zoom.us/v2; clientId=0aBcDeF_GhI2j_KlMnOpQr; clientSecret=*****; authUri=https://zoom.us/oauth/authorize; tokenUri=https://zoom.us/oauth/token; redirectUri=http://localhost;AuthenticationMethod=OAuth2; accesstoken: *****; </pre> <ul style="list-style-type: none"> • jdbc:weblogic:autorest://servername;[property=value[;...]]; • jdbc:weblogic:autorest:Sample=sample_path;[property=value[;...]]; 	

Create Custom Connectors

APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

The Custom Connections page of the Administration tab of Oracle Data Transforms helps you to create custom connectors that point to any JDBC supported data sources.

The custom connectors will be listed in the Create Connection page where you can use them to connect data sources to Data Transforms. See [Work with Connections](#) for more information. To create a new connector:

1. In the left pane, click **Administration**.
A warning message appears.
2. Click **Continue**.
3. In the left pane, click **Custom Connections**.
Custom Connections screen appears.
4. Click **Create Connection Type**.
The Create Connection Type page appears.
5. From the **Category** drop-down select the type of connection that you wish to create whether database, application, or service.
6. Enter a name for the connection.
7. Enter the name of the JDBC Driver of the source connection. For example, `oracle.jdbc.OracleDriver`.

Note

For connectors that require driver installation, you need to copy the jar files to the `/u01/oracle/transforms_home/userlibs` directory before you add the connection.

8. Click **OK**.

The newly created custom connection appears in the list and are available in the Create Connection page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete**, to delete the created connection.

Note

You cannot delete custom connectors that have existing connections.

Create a Data Transforms Connection for Remote Data Load

You can connect to an existing Data Transforms instance and run a data load remotely.

To create this connection, you need to specify the URL of the Data Transforms instance along with the name of the ODI rest API from where you want to run the data load.

To define a Data Transforms connection:

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.
2. Click **Create Connection**.
Create Connection page slides in.
3. For **Select Type**,
 - In the **Name** field, enter the name of the newly created connection
 - Select **Services** as the type of connection that you wish to create.
4. In the **Endpoint URL** textbox, enter the URL of the ODI rest API from where you want to run the data load. Enter the URL in the format `http://<host-ip-address>:<port>/odi-rest`.
5. In the **User** text box enter SUPERVISOR as the user name.
6. In the **Password** text box enter the ODI Supervisor password.
7. After providing all the required connection details, click **Test Connection** to test the established connection.
8. Click **Create**.
The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Create an Apache Iceberg Connection

Apache Iceberg is an open standard table format that is optimized to manage large analytic datasets. Data Transforms supports the use of Apache Iceberg as a target to load data from any SQL-based data sources.

Data Transforms supports Oracle Object Storage (S3 compatibility) and AWS S3 storage services to store the parquet files for the Apache Iceberg tables.

The Data Transforms Apache Iceberg Connector requires that a REST Catalog already exists. This REST Catalog is setup based on Apache Gravitino ([external link](#)) with Iceberg Open API specification.

Note

Data Transforms supports the use of Apache Gravitino version 0.7.0-incubating or lower to bring up the REST service.

This topic has the following sections:

- [Creating an Apache Iceberg Connection](#)
- [Creating and Running an Apache Iceberg Data Load](#)

Creating an Apache Iceberg Connection

You can configure an Apache Iceberg connection with the Iceberg REST Catalog by providing the REST URL and authentication details such as the username and password. You can also use the more secure OAuth 2.0 authentication to create the connection.

To create an Apache Iceberg connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Databases** tab.
4. Select **Apache Iceberg** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **Catalog Name** textbox, enter a name.
8. In the **Rest URL** textbox, enter the URL of the REST server. Enter the value in the `<host>:<port>/<ServiceName>/iceberg` format.
9. From the **Authentication** drop-down section, do one of the following:
 - Select **None**.
 - Select **Simple** and enter the **Rest User** and **Rest Password**.
 - Select **OAuth** and enter the following details:
 - **Warehouse Location:** The location where you want to store the data. For example, `s3://my-bucket/my/table/location`
 - **Token URI:** The URL to obtain the OAuth Token in the format `http://<host>:<port>`
 - **Token Path:** The path to the OAuth token. For example, `/oauth2/token`.
 - **Client ID:** The OAuth Client ID.
 - **Client Secret:** The OAuth Client secret.
 - **Auth Scope:** The permissions granted to a client when accessing the Gravitino server. For example, a `test` Auth Scope value might indicate that the client is authorized to access resources related to the "test" scope within Gravitino. [Optional]

- **Grant Type:** The method that the authorization server should use to issue the access token. For example, `client_credentials` and `authorization_code`. [Optional]

10. Click **Test Connection**, to test the established connection.
11. After providing all the required connection details, click **Create**.
The Apache Iceberg connection is configured with REST Catalog, which stores the Iceberg data in Oracle Object Storage.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Creating and Running an Apache Iceberg Data Load

You can create a data load for any SQL-based source data source, such as Oracle, to load data into Apache Iceberg target tables. To use Apache Iceberg as a target data source, you need to provide the name of the connection and the namespace. A namespace in Apache Iceberg is similar to schema in relational databases.

After you create the data load, all the tables in the source schema are listed on the Data Load Detail page along with options to incrementally load, append, and merge the data for each of the selected source tables. When the data load run completes, you can read the data from the Iceberg tables. You can add the data load as a step in a workflow and create a schedule to run the workflows at a predefined time interval. See [Create a New Workflow](#).

To create and run an Apache Iceberg Data Load:

1. Do one of the following:
 - On the Home page, click **Load Data**. The Create Data Load wizard appears. In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
 - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**. The Create Data Load wizard appears.
2. Enter a name if you want to replace the default value and add a description.
3. For Load Processing do one of the following:
 - Select the **Internal** radio button and from the **Deployment Type** drop-down select **Data Transforms (Batch)**.

- Select the **Delegate** radio button and from the **Deployment Type** drop-down select OCI GoldenGate. From the GoldenGate Deployment Connection select a connection.
4. Click **Next**.
 5. In the Source Connection tab,
 - a. From the **Connection Type** drop-down, select a SQL-based data source.
 - b. from the **Connection** drop-down, select the required connection from which you wish to add the data entities.
 - c. Click **Next**.
 6. In the Target Connection tab,
 - a. From the **Connection Type** drop-down, select **Apache Iceberg** as the connection type.
 - b. From the **Connection** drop-down, select the connection you want to use to load the data into.
 - c. Specify the **Namespace**. You can either select from existing namespaces or create a new namespace.
 - d. Click **Save**.

The Data Load Detail page appears listing all the source tables.
 7. Select the required tables to load and the corresponding data load operation. The data load options you can use are Incremental Merge, Incremental Append, Append, and Do Not Load.
 8. Click  to save the changes. A green checkmark () in the row indicates that the changes are saved.
 9. Click  to run the data load.
A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Data Load Status panel on the right below the Target Schema details. For details about the panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see [Create and Manage Jobs](#).

All the loaded tables along with their details are listed in the Data Entities page. To view the

statistics of the data entities, click the **Actions** icon () next to the data entity, click **Preview**, and then select the **Statistics** tab. See [View Statistics of Data Entities](#) for information.

Create a Delta Share Connection

Databricks Delta Share is an open protocol for secure data sharing. Oracle Data Transforms integrates with Delta Share to load data to Oracle Autonomous Database. You can use the Delta Share connection to load data from Databricks or Oracle Data Share.

To use Databricks as a source, you need to specify the URL of the Delta Sharing server along with the bearer token that lets you access the Delta Lake share server. To use Oracle Data Share as a source, you need to specify the URL for the token end point along with a client ID and the secret key.

This topic has the following sections:

- [Creating the Delta Share Connection](#)
- [Creating and Running a Delta Share Data Load](#)

Creating the Delta Share Connection

To define a Delta Share connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Databases** tab.
4. Select **Delta Share** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **Share Endpoint URL** textbox, enter the URL of the Delta Sharing server. Enter the value in the <host>:<port>/<shareEndpoint>/ format.
8. In the Connection section, do one of the following:
 - Select **Oracle Data Share** and provide the **Token Endpoint URL**, **Client ID**, and **Client Secret** for accessing the share.
You can get this information from the Delta Share Profile JSON document that you will need to download from supplied to you by the Share Provider. (This is also where they get the Share Endpoint URL from)

You can get this information from the Delta Share Profile JSON document that you can download from the activation link that is provided by the Data Share provider to access their share.
 - Select **Databricks** and in the **Bearer Token** text box enter the token for connecting to the Delta Sharing server.
9. If you need to use a proxy to access the Delta Share Server or Delta Share Storage configure the following settings:
 - In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
 - In the **Proxy Port** textbox, enter the port number of the proxy server.
 - Select the following checkboxes depending on where the proxy is required:
 - **Use Proxy to access Delta Share Server**
 - **Use Proxy to access Delta Share Storage**
10. Click **Test Connection**, to test the established connection.
11. After providing all the required connection details, click **Create**.
The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Creating and Running a Delta Share Data Load

To load data from Delta Share into Oracle Autonomous Database, the Oracle connection user must be an Admin user. Admin privileges are required so that the Oracle user can create and insert data into tables in another schema.

When you run the data load, Data Transforms loads the data onto a corresponding table in the target schema. The data load runs incrementally. The very first time you run a data load, Data Transforms copies all the data into new tables. For every subsequent data load run, it only uploads the changes. Any additions or deletions in the records will reflect in the target tables. Note that if there is any metadata change in the table, for example a column is added, Data Transforms creates a new table to load the data on to the target server. You could create a workflow, add the data load as a step, create a schedule to run the workflows at a predefined time interval. See [Create a New Workflow](#).

To create and run a Delta Share data load:

1. Do one of the following:
 - On the Home page, click **Load Data**. The Create Data Load wizard appears. In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
 - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**. The Create Data Load wizard appears. In the Create Data Load tab, enter a name if you want to replace the default value and add a description.
2. Click **Next**.
3. In the Source Connection tab,
 - a. From the **Connection Type** drop-down, select **Delta Share**.
 - b. from the **Connection** drop-down, select the required connection from which you wish to add the data entities.
 - c. Select the share that you want to load tables from the **Share** drop-down. The drop-down lists all the shares for the selected connection.
 - d. Click **Next**.
4. In the Target Connection tab,
 - a. From the **Connection Type** drop-down, select **Oracle** as the connection type.

Note

This drop-down lists only JDBC type connections.

- b. From the **Connection** drop-down, select the required connection from to you wish to load the data entities.
- c. Enter a unique name in the **Schema** textbox.
- d. Click **Save**.

The Data Load Detail page appears listing all the tables in the selected share with their schema names.

Note

For Delta Share data loads the Data Load Detail page only includes the  option. You cannot apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema. This is to make sure that the data is consistent between the Delta Sharing server and the target schema.

5. Click  to run the data load.
A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Status panel on the right below the Target Schema details. For details about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see [Create and Manage Jobs](#).

All the loaded data entities along with their details are listed in the Data Entities page. To view

the statistics of the data entities, click the **Actions** icon () next to the data entity, click **Preview**, and then select the **Statistics** tab. See [View Statistics of Data Entities](#) for information.

Create an Oracle Business Intelligence Cloud Connector Connection

Oracle Business Intelligence Cloud Connector (BICC) allows you to extract business data from a data source and load it into Autonomous Database.

To create an Oracle BICC connection you need to first configure external storage using the OCI Object Storage Connection tab in the BICC Console. You need to specify these connection details when you define the connection in Oracle Data Transforms.

You can use the BICC connection to choose the offerings whose data stores you want to extract. Data Transforms uses an Oracle Object Storage Data Server used by Oracle BICC to stage the extracted files, which you can then use as a source for mapping. Note that you cannot use an Oracle BICC connection as a target for mapping.

To define an Oracle BICC connection,

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.

2. Click **Create Connection**.
Create Connection page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Oracle BI Cloud Connector** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Enter the URL in the **BI Cloud Connector Service URL** textbox.
8. In the **Connection** section, enter the following details:
 - In the **User** text box enter the user name configured in the Oracle BI Cloud Connector Console.
 - In the **Password** text box enter the password configured in the Oracle BI Cloud Connector Console.
9. In the **Storage** section, enter the following details:
 - In the **External Storage BICC Name** text box enter the name of the external storage as it appears in the Oracle BI Cloud Connector Console.
 - In the **External Storage Bucket** text box specify the bucket into which extracts are uploaded. Bucket names are obtained in the OCI Console.
 - In the **External Storage Name Space** text box specify the namespace. Namespace is obtained in the OCI Console.
 - In the **External Storage Region** text box enter the OCI Object Storage region.
 - In the **External Storage User** text box enter your Oracle Cloud Infrastructure username.
 - In the **External Storage Token** text box enter the auth token.
10. Click **Test Connection** to test the established connection.
11. Click **Create**.

The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.

- **Technology** associated with the created Connection.

Create and use an Oracle Cloud Infrastructure Generative AI Connection

Oracle Cloud Infrastructure (OCI) Generative AI enables organizations to automate text summarization and dynamic content generation. Data Transforms integrates with OCI Generative AI to support the use of embedding vectors in a data flow.

Before you create an OCI Generative AI connection you create an Oracle Database 23ai connection. Data Transforms will use this connection to test the OCI Generative AI connection. To create the OCI Generative AI connection you need to specify details such as the OCI URL, User OCID, Tenancy OCID, Compartment OCID, Private Key, and Fingerprint information.

See [Use Text Embedding Vector in a Data Flow](#) for information on how you will use this connection to add vector embedding in a data flow.

To define Oracle Cloud Infrastructure Generative AI Connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. For **Select Type**,
 - In the **Name** field, enter the name of the newly created connection.
 - Select **Services** as the type of connection.
4. Select **OCI Generative AI**, and **Next**.
5. For **Connection Details**, provide the following details:
 - **OCI URL** - The endpoint URL of the OCI Generative AI service.
 - **User OCID** - The user OCID from the Oracle Cloud Infrastructure Console.
 - **Tenancy OCID** - The tenancy OCID from the Oracle Cloud Infrastructure Console.
 - **Compartment OCID** - The compartment OCID from the Oracle Cloud Infrastructure Console.
 - **Private Key** - The private key in the PEM format. Specify the path to your downloaded private key file.
 - **Fingerprint** - The fingerprint of the key that was just added.
6. After providing all the required connection details, click **Create**. The new connection is created.
7. Click **Test Connection**, to test the established connection. A pop-up appears listing the Oracle Database 23ai connections that you have configured. Select the option you want to use to test this connection.

Note

If test connection fails see [Troubleshooting OCI Generative AI connection issues](#) for instructions to fix the issue.

The newly created connection is displayed in the Connections page.

Troubleshooting OCI Generative AI connection issues

If **Test Connection** fails for an OCI Generative AI connection, do the following to troubleshoot the issue:

1. Make sure that you have entered all the connection information correctly. For example,
 - OCI URL sample: `https://inference.generativeai.us-chicago-1.oci.oraclecloud.com/20231130/actions/embedText`
 - Private Key sample: `MIIEvg.....beE/`
2. When you click **Test Connection**, you will be asked to choose an Oracle Connection. Make sure that it is an Oracle 23ai connection.
3. Log into that Oracle 23ai database as admin and run the following query to make sure the statuses are valid:

```
SELECT object_name, object_type, status
FROM dba_objects
WHERE object_name = 'DBMS_VECTOR_CHAIN' AND owner = 'CTXSYS';
```

4. Check whether the following plsql block works in your database. Data Transforms uses the plsql block to run the **Test Connection**.

```
exec dbms_vector_chain.drop_credential('OCI_CRED');
declare
  jo json_object_t;
begin
  -- create an OCI credential
  jo := json_object_t();
  jo.put('user_ocid', '<your user ocid>');
  jo.put('tenancy_ocid', '<your tenancy ocid>');
  jo.put('compartment_ocid', '<your compartment ocid>');
  jo.put('private_key', '<your private key>');
  jo.put('fingerprint', '<your fingerprint>');
  dbms_output.put_line(jo.to_string);
  dbms_vector_chain.create_credential(
    credential_name => 'OCI_CRED',
    params => json(jo.to_string));
end;
```

```
select dbms_vector_chain.utl_to_embedding('hello', JSON('{"provider":
"ocigenai","credential_name" : "OCI_CRED", "url": "https://
inference.generativeai.us-chicago-1.oci.oraclecloud.com/20231130/actions/
embedText","model": "cohere.embed-english-light-v2.0"}')) from dual;
```

If you get an HTTP request failed error when running the plsql block query, try to grant network access to your user:

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host => '*',
    ace => xs$ace_type(privilege_list => xs$name_list('connect'),
      principal_name => '<your username>');
```

```
principal_type => xs_acl.ptype_db));  
END;
```

Create an Oracle Enterprise Resource Planning Cloud Connection

Oracle Enterprise Resource Planning (ERP) Cloud is a cloud based end-to-end Software as a service (SaaS) suite to manage functions such as accounting, financial management, project management, procurement, and risk management. The Data Transforms Oracle ERP Cloud connector allows you to extract report data from an ERP Cloud server and load it into Autonomous Database.

To use an Oracle ERP Cloud connection in Data Transforms you need the URL of the BI Publisher web service used for the Oracle ERP Cloud instance and BI Publisher report file that is output in the Data (CSV) format.

Note the following:

- You cannot use an Oracle ERP Cloud connection as a target for mapping.
- Data load is not supported for Oracle ERP Cloud connections.

This topic has the following sections:

- [Creating an Oracle ERP Cloud Server Connection](#)
- [Importing Data Entity Definitions from an Oracle ERP Cloud Server](#)

Creating an Oracle ERP Cloud Server Connection

To create this connection, you need to specify the WSDL URL of the Report service and the username and password to connect to the ERP Cloud instance.

To define an Oracle ERP Cloud Server Connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Oracle ERP Cloud** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **WSDL URL** textbox, enter the URL of the BI Publisher web service used for the Oracle ERP Cloud instance. The format of the URL is *BI Publisher Hostname/xmlpserver/services/PublicReportWSSService?wsdl*.
For example: `https://fa-eudu-saasfademo1.ds-fa.oraclepdemos.com/xmlpserver/services/PublicReportWSSService?wsdl`.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.
10. In the **User** text box enter the user name for connecting to the Oracle ERP Cloud instance.

11. In the **Password** text box enter the password for connecting to the Oracle ERP Cloud instance.
12. After providing all the required connection details, click **Test Connection** to test the connection.
13. Click **Create**.
The newly created connection is displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the connection.
- **Technology** associated with the created connection.

Importing Data Entity Definitions from an Oracle ERP Cloud Server

For Oracle ERP Cloud connections, the Import Data Entities page requires input in key/value pairs. To import data, you need to specify the `BIPReportLocation` key, which refers to the location of the BI publisher report on the BI server. You can find this information on the BI server when the corresponding report is open.

To import data entities from Oracle ERP Cloud:

1. From the left pane of the Home page, click the **Data Entities** tab.
Data Entities page appears.
2. Click **Import Data Entities**. The Import Data Entities page slides-in.
3. In the **Connection** drop-down, select the Oracle ERP Cloud connection that you created in [Creating an Oracle ERP Cloud Server Connection](#). The options on the page change to display only the **Entity Name** field.
4. In the **Entity Name** field enter a name for the data entity.
5. Enter `BIPReportLocation` in the **Key** field.
6. Enter the location of the BI Publisher report file in the **Value** field corresponding to the key. Specify the path till the name of the `ReportFile.xdo` file.
7. Click **Start**.
A Job is created, and the corresponding Job ID is displayed for you to track the session. Click the Job ID to view the details of the job. Upon successful execution of the job, the data entity is displayed on the Data Entities page.
8. Click the **Actions** icon () next to the selected Data Entity and select **Edit**.
The **Edit Data Entity** page appears that displays a table with the following information:
 - Name
 - Data Type

- Length
- Scale
- Not Null

Verify the data type and length for the imported entities. Oracle Data Transforms determines the data types and length from the first record of the file, but may set default values (for example, 50 for the string field length) or incorrect data types. In case of an empty field, data type is set to String with length 50.

9. Click **Save**.

Create an Oracle Financials Cloud Connection

You can fetch real time transactional data from Oracle Financials Cloud REST endpoints, import the data entities into Data Transforms, and use them as a source in a data flow.

To create an Oracle Financials Cloud connection you need to choose a temporary schema where Data Transforms can create data entities after the reverse-engineering operation.

To define an Oracle Financials Cloud connection,

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.
2. Click **Create Connection**.
Create Connection page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Oracle Financials Cloud** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **REST Service URL** textbox, enter the URL of the endpoint that services the REST resources.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.
10. In the **User** text box enter the user name for connecting to the REST endpoint.
11. In the **Password** text box enter the password for connecting to the REST endpoint.
12. Choose a connection from the **Staging Connection** drop-down list. The list displays only existing Autonomous Database connections. To use a different connection, create the connection before you reach this page.
13. After providing all the required connection details, click **Create**.
The new connection is created.
14. Click **Test Connection**, to test the established connection.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Create and Use an Oracle NetSuite Connection

You can use the Oracle NetSuite JDBC Driver or OAuth 2.0 authentication to connect to the Oracle NetSuite application. For Oracle NetSuite connections, Data Transforms allows you to load pre-built dataflows and workflows that you can run to transfer data from NetSuite to your target schema.

This topic has the following sections:

- [Configuring Access Permissions Required for Building Data Warehouse](#)
- [Creating the Oracle NetSuite Connection](#)
- [Using the Build Data Warehouse Wizard](#)
- [Running the Pre-Built Workflows to Load Data into the Target Schema](#)

Configuring Access Permissions Required for Building Data Warehouse

Before you create a NetSuite connection or use the Build Data Warehouse Wizard in Data Transforms, you need to login to NetSuite as an administrator, enable SuiteAnalytics Connect, create a custom role, and set the access permissions that are required to build the NetSuite Data Warehouse.

To configure the access permissions:

1. Log in to NetSuite as an administrator using the following URL:
<https://system.netsuite.com/pages/customerlogin.jsp>
2. Check whether the SUITEANALYTICS CONNECT feature is enabled.
Go to *Setup > Company > Enable Features*. Click the **Analytics** tab and verify that the **SuiteAnalytics Connect** box is checked.
3. Add a custom role.
Go to *Setup > Users/Roles > User Management > Manage Roles*. Click **New Role**, add the required details, and click **Save** to create a custom role.
4. Assign the required permissions to the custom role.
Go to *Setup > Users/Roles > User Management > Manage Roles*. Click **Customize** next to the name of the custom role for which you would like to add the SuiteAnalytics Connect permission.

Click the **Transactions** tab under the **Permissions** tab and assign the following permissions:

- Account Detail
- Accounting Lists
- Accounting Management
- Accounts
- Accounts Payable
- Accounts Payable Graphing
- Accounts Payable Register
- Accounts Receivable
- Accounts Receivable Graphing
- Accounts Receivable Register
- Adjust Inventory
- Adjust Inventory Worksheet
- Amortization Reports
- Amortization Schedules
- Audit Trail
- Balance Sheet
- Bank Account Registers
- Bill Purchase Orders
- Billing Schedules
- Bills
- Bin Putaway Worksheet
- Bin Transfer
- Blanket Purchase Order
- Build Assemblies
- CRM Groups
- Calendar
- Cash Sale
- Cash Sale Refund
- Charge
- Charge - Run Rules
- Charge Rule
- Check
- Classes
- Commission Feature Setup
- Commission Reports
- Commit Orders
- Commit Payroll
- Competitors

- Component Where Used
- Contacts
- Count Inventory
- Create Allocation Schedules
- Credit Card
- Credit Card Refund
- Credit Card Registers
- Credit Memo
- Credit Returns
- Currency
- Currency Revaluation
- Custom Recognition Event Type
- Customer Deposit
- Customer Payment
- Customer Refund
- Customers
- Deferred Expense Reports
- Deleted Records
- Departments
- Deposit
- Deposit Application
- Documents and Files
- Edit Forecast
- Edit Manager Forecast
- Email Template
- Employee Commission Transaction
- Employee Commission Transaction Approval
- Employee Record
- Employee Reminders
- Employees
- Enter Opening Balances
- Enter Vendor Credits
- Equity Registers
- Establish Quotas
- Estimate
- Events
- Expense Report
- Expenses

- Export Lists
- Fair Value Dimension
- Fair Value Formula
- Fair Value Price
- Financial Statements
- Find Transaction
- Fixed Asset Registers
- Fulfill Orders
- Fulfillment Request
- General Ledger
- Generate Price Lists
- Generate Statements
- Imported Employee Expenses
- Inbound Shipment
- Income
- Income Statement
- Inventory
- Inventory Status Change
- Invoice
- Invoice Approval
- Invoice Sales Orders
- Item Fulfillment
- Item Receipt
- Item Revenue Category
- Items
- Lead Snapshot/Reminders
- Locations
- Long Term Liability Registers
- Make Journal Entry
- Manage Accounting Periods
- Manage Payroll
- Mass Updates
- Memorized Transactions
- Mobile Device Access
- Net Worth
- Non Posting Registers
- Notes Tab
- Opportunity

- Other Asset Registers
- Other Current Asset Registers
- Other Current Liability Registers
- Other Lists
- Other Names
- Ownership Transfer
- Pay Bills
- Pay Sales Tax
- Pay Tax Liability
- Paycheck Journal
- Payroll Items
- Perform Search
- Phone Calls
- Post Vendor Bill Variances
- Posting Period on Transactions
- Price Books
- Price Plans
- Process Payroll
- Project Revenue Rules
- Purchase Contract
- Purchase Order
- Purchase Order Reports
- Purchases
- Reconcile
- Reconcile Reporting
- Refund Returns
- Report Customization
- Report Scheduling
- Request For Quote
- Requisition
- Resource
- Return Authorization
- Revalue Inventory Cost
- Revenue Arrangement
- Revenue Arrangement Approval
- Revenue Element
- Revenue Recognition Field Mapping
- Revenue Recognition Plan

- Revenue Recognition Reports
- Revenue Recognition Rule
- Revenue Recognition Schedules
- SOAP Web Services
- Sales
- Sales By Partner
- Sales By Promotion
- Sales Order
- Sales Order Approval
- Sales Order Fulfillment Reports
- Sales Order Reports
- Sales Order Transaction Report
- Set Up Budgets
- Set Up SOAP Web Services
- Statement Charge
- Statistical Account Registers
- Store Pickup Fulfillment
- Subscription Change Orders
- Subscription Plan
- Subscriptions
- Subsidiaries
- SuiteAnalytics Connect
- SuiteAnalytics Workbook
- Tasks
- Tax
- Track Messages
- Transaction Detail
- Transfer Funds
- Transfer Inventory
- Transfer Order
- Transfer Order Approval
- Trial Balance
- Unbilled Receivable Registers
- Unbuild Assemblies
- Units
- Vendor Bill Approval
- Vendor Payment Approval
- Vendor Request For Quote

- Vendor Return Auth. Approval
 - Vendor Return Authorization
 - Vendor Returns
 - Vendors
 - Work Calendar
 - Work Order
 - Work Order Close
 - Work Order Completion
 - Work Order Issue
5. Add the **SuiteAnalytics Connect – Read All** permission. Click the **Setup** tab under the **Permissions** tab, select **SuiteAnalytics Connect – Read All** from the drop-down, and click **Add**.
 6. Click **Save** to apply these permissions to the custom role.
 7. Assign the custom role to a user. When you create a connection to NetSuite from Data Transforms, you will need to enter the credentials of this user to connect to the data server. See [Creating the Oracle NetSuite Connection](#) for information about creating the connection.
To assign the custom role to the user, go to *Setup > Users / Roles > Manage Users*. Click **Edit** next to the name of the user, assign the custom role, and click **Save**.
 8. To verify the access permissions, log in as the user that has the custom role assigned. Go to *Analytics > Datasets*. Click **New Dataset**. This page will list all the tables and record types that the user has access to. Search for “transaction” table, for example, to verify whether the user has access to the transaction table.

Creating the Oracle NetSuite Connection

You can create an Oracle NetSuite connection using JDBC connectivity or OAuth 2.0 authentication.

To define an Oracle NetSuite connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Oracle NetSuite** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. To specify the connection details, do one of the following:
 - To use JDBC connectivity, specify the following details:
 - **JDBC URL** - Enter the URL of the SuiteAnalytics Connect server to be used for the connection.
 - **User** - Enter the user name for connecting to the data server.

- In the **Password** textbox enter the password for connecting to the data server.
- In the **Account ID** textbox, enter the account ID for connecting to the data server.
- In the **Role ID** textbox, enter the role ID for connecting to the data server.
- To use OAuth 2.0 authentication, click the **OAuth 2.0** switch and then specify the following details:
 - **Username** - Enter the name of the user who has role access to login to NetSuite using OAuth 2.0 connection. This is the user you have assigned the custom role to in [Configuring Access Permissions Required for Building Data Warehouse](#).
 - **Account ID** - Enter the account ID for connecting to the data server. You can get this information by logging into the NetSuite account and viewing the SuiteAnalytics connect information.
 - **Role ID** - Enter the role ID for connecting to the data server. You can get this information by logging into the NetSuite account and viewing the SuiteAnalytics connect information.
 - **Client ID** - Enter the client ID for connecting to the data server. To obtain the client ID, create an Integration record in NetSuite by enabling OAuth 2.0 Client Credentials Flow. Copy and save the Client ID that is displayed when the Integration Record is successfully created.

Note

NetSuite no longer supports the RSA PKCSv1.5 scheme for token signing for NetSuite OAuth 2.0 client credentials flow. Any integrations that rely on the RSA PKCSv1.5 scheme will need to be updated to use the RSA-PSS scheme. Refer to the Oracle NetSuite documentation for more information.

- **Public Certificate and Private Key** - Use the OpenSSL commands to generate the key pair in the required PEM format. For example,


```
openssl req -new -x509 -newkey rsa:4096 -keyout private.pem -sigopt rsa_padding_mode:pss -sha256 -sigopt rsa_pss_saltlen:64 -out public.pem -nodes -days 365
```

Paste the contents of `public.pem` in the Public Certificate field. Paste the contents of `private.pem` in the Private Key field.
 - **Certificate ID** - Enter the Certificate ID for connecting to the data server. To get the certificate ID, use the NetSuite OAuth 2.0 Client Credentials (M2M) Setup to add the public certificate file (`auth-cert.pem`) to the certificate key list and copy the generated Certificate ID.
8. If the source that you want to use for mapping is a saved search, you need to also specify the following details in **Saved Search Extraction**:
 - **Application ID**: Enter the NetSuite Application ID for Data Transforms.
 - **Version**: Enter the NetSuite version number.
 9. Select the checkbox in **Build Data Model** to install pre-built dataflows and workflows that you can run to extract data from NetSuite and move it to your Oracle target schema using the Build Data Warehouse wizard.
 10. Click **Test Connection**, to test the established connection.
 11. After providing all the required connection details, click **Create**.

The new connection is created.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Select **Build Data Warehouse**, to select the functional areas and create the NetSuite Data Warehouse in the target schema. See [Using the Build Data Warehouse Wizard](#) for more information.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schema.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Using the Build Data Warehouse Wizard

Data in your NetSuite account is grouped into business or subject areas in the Analytics Warehouse. The Build Data Warehouse wizard allows you to select the areas that you want to include in the newly created Data Warehouse.

To use the Build Data Warehouse Wizard:

1. On the Home page, click the **Connections** tab. The **Connections** page appears.
2. Click the Actions icon () next to the Oracle NetSuite connection that you want to use to build the data warehouse and click **Build Data Warehouse**. The Build Data Warehouse wizard opens.
3. From the **Connection** drop-down list, choose the Autonomous Database connection where your target schema resides.
4. From the **Staging Schema** drop-down, all schema corresponding to the selected connection are listed in two groups:
 - Existing Schema (ones that you've imported into Oracle Data Transforms) and
 - New Database Schema (ones that you've not yet imported).Select the schema that you want to use from the drop-down.
5. Similarly select the **Target Schema**.
6. Click **Next**.
7. Select the **NetSuite Business Areas** that you want to use to transfer data from the NetSuite Data Warehouse to the target schema.
8. Click **Save**.
Data Transforms starts the process to build the data warehouse. Click **Jobs** on the left pane of the Home page to monitor the progress of the process. When the job completes successfully, Data Transforms creates a Project folder that includes all the pre-built

workflows and dataflows, which you can run to transfer data from the NetSuite connection to your target schema. See [Running the Pre-Built Workflows to Load Data into the Target Schema](#) for more information.

Running the Pre-Built Workflows to Load Data into the Target Schema

When the Build Data Warehouse wizard completes successfully, Data Transforms creates a project that includes all the pre-built data flows and workflows that you can run to extract data from a NetSuite connection and load it into your target schema.

To view and run the pre-built workflows:

1. Click **Projects** on the left pane of the Home page and select the newly created NetSuite project.
2. Click **Workflows** in the left pane. The following pre-built workflows are listed in the Project Details page:
 - Stage NetSuite Source to SDS
 - Extract Transaction Primary Keys
 - Load SDS to Warehouse
 - Apply Deletes
 - All Workflows
3. Click the Actions icon () next to the workflow you want to run and click **Start**. Oracle recommends that you run **All Workflows** to execute all the pre-built workflows.

To see the status of the workflow, click **Jobs** from the left pane in the current project. When the job completes successfully, all the data from the NetSuite connection is loaded into the target schema.

Create an Oracle Object Storage Connection

You can use Data Transforms to upload data from Oracle Object Storage to Autonomous Database.

The OCI Object Storage dedicated endpoints feature allows OCI customers to securely access the storage buckets. See [Object Storage Dedicated Endpoints](#) for more information. You need to use the new URL format when you create Object Storage connections in Data Transforms. For users that already have an Object Storage connection, the existing URL is automatically updated to the new URL format.

To create an Oracle Object Storage connection you need to have an Oracle Cloud Infrastructure username and an auth token. See [Getting an Auth Token](#) for information about how to generate the auth token. You need to specify these details when you define the connection in Oracle Data Transforms.

Note the following:

- To use an Oracle Object Storage connection to import data into Data Transforms, you must use a public IP address to access the compute node. If you want to use a private IP address to access the Object Storage service, make sure that you have access to the Internet.
- The supported file format for loading data from Oracle Object Storage to Autonomous Database and vice versa is CSV.
- The supported data types are Numeric, Double, String, and Date.

- Data load is not supported for Oracle Object Storage connections.
- To create a mapping to Object Storage, the source technology must support the `DBMS_CLOUD` package. Otherwise, the mapping will fail.
- Data Transforms does not support cross-realm connectivity for Object Storage. For example, you cannot connect from Data Transforms configured in a Government Cloud region to Object Storage that is in a commercial realm.

To define an Oracle Object Storage connection,

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.
2. Click **Create Connection**.
Create Connection page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Databases** tab.
4. Select **Oracle Object Storage** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Enter the URL in the **Object Storage URL** textbox. You can enter the URL in either of the following formats:
 - URL with fully qualified domain name.
For example,

```
https://<namespace>.swiftobjectstorage.<your-region>.oci.customer-oci.com/v1/<your-namespace>/<your-bucket>
```

```
https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/<your-namespace>/b/<your-bucket>/o
```

- If you want to use the URL provided by the OCI Console, specify the URL only till the name of the bucket.
For example,

```
https://<namespace>.swiftobjectstorage.<your-region>.oci.customer-oci.com/v1/<your-namespace>/<your-bucket>
```

```
https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/<your-namespace>/b/<your-bucket>/o
```

- If you choose **Credential** as the **Connection Mode** (see step 6), specify the URL till `bucketname/o`
For example,

```
https://<namespace>.objectstorage.<your-region>.oci.customer-oci.com/n/<your-namespace>/b/<your-bucket>/o/
```

Note

Credential mode is available only to Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

The values for Region, Namespace and Bucket are auto-populated based on the URL provided.

8. To select the **Connection Mode** do one of the following:
 - Select **Swift Connectivity**, and provide the following details:
 - In the **User Name** text box enter your Oracle Cloud Infrastructure username. For tenancies that support identity domains, specify the domain name along with the username. For example, `<identity-domain-name>/<username>`.
 - In the **Token** text box enter the auth token.
 - (This is applicable only to Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.) Select **Credential** and provide the ODI credential in the **Enter Credential** text box. You must create the credential in the repository and in the Autonomous Database that you created during instance creation. When you create a data flow to map data from Object Storage to Autonomous Database you need to create the ODI credential in the target schema as well. Before you run the mapping, make sure that you select the step and in the Properties panel, set the **Create credential** KM option to `false`. Otherwise, the credential-based connection will fail.

To create the credential, execute the following script:

```
begin DBMS_CLOUD.create_credential( credential_name => '<Credential
name>',
    username => '<oci username>', password => '<auth token>' ); end;
```

9. Click **Create**.

The new connection is created.

10. Click **Test Connection**, to test the established connection.

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit**, to edit the provided connection details.
- Select **Test Connection**, to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema**, to delete schemas.
- Select **Delete Connection**, to delete the created connection.

You can also search for the required Connection to know its details based on the following filters:

- **Name** of the Connection.
- **Technology** associated with the created Connection.

Create a REST Server Connection

You can connect to any REST service endpoint, import the data entities into Data Transforms, and use them as source in a data flow.

To create a generic REST connector, you need to provide the JDBC URL, username, and password to connect to the endpoint. You can also create and upload a config file that contains information such as the authentication methods, endpoints, and tables that you want to import data entities from.

The **Application** tab on the **Create Connection** page includes two connection options to create a generic REST connection - Generic REST and Generic REST Config. This topic has the following sections:

- [Creating a Generic REST Connection](#)
- [Creating a Generic Rest Connection Using a Config File](#)

The newly created connections are displayed in the **Connections** page.

Click the **Actions** icon () next to the selected connection to perform the following operations:

- Select **Edit** to edit the provided connection details.
- Select **Test Connection** to test the created connection.
- Click **Export** to export the connection. See [Export Objects](#).
- Select **Delete Schema** to delete schemas.
- Select **Delete Connection** to delete the created connection.

You can also search for the required connection to know its details based on the following filters:

- **Name** of the connection.
- **Technology** associated with the created connection.

Creating a Generic REST Connection

To create this connection you need to specify the REST service URL and choose a temporary schema where Data Transforms can create data entities after the reverse-engineering operation.

To define a REST server connection:

1. From the left pane of the Home page, click the **Connections** tab. **Connections** page appears.
2. Click **Create Connection**. **Create Connection** page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Generic Rest** as the connection type.
5. Click **Next**.

6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. In the **REST Service URL** textbox, enter the URL of the endpoint that services the REST resources.
8. In the **Proxy Host** textbox, enter the host name of the proxy server to be used for the connection.
9. In the **Proxy Port** textbox, enter the port number of the proxy server.
10. In the **User** text box enter the user name for connecting to the REST endpoint.
11. In the **Password** text box enter the password for connecting to the REST endpoint.
12. Choose a connection from the **Staging Connection** drop-down list. The list displays only existing Autonomous Database connections. To use a different connection, create the connection before you reach this page.
13. After providing all the required connection details, click **Test Connection** to test the connection.
14. Click **Create**.
The new connection is created.

Creating a Generic Rest Connection Using a Config File

APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

To create a generic REST connector, you need the JDBC URL, username, password, and a config file. The config file is a model file with the *file_name.rest* naming convention that you need to upload when you create a REST Server connection. You need to specify the endpoints, table mappings, and the authentication methods to create the config file. You can create the config file using any text editor.

To define a REST server connection using a config file:

1. From the left pane of the Home page, click the **Connections** tab.
Connections page appears.
2. Click **Create Connection**.
Create Connection page slides in.
3. Do one of the following:
 - In the **Select Type** field, enter the name or part of the name of the connection type.
 - Select the **Applications** tab.
4. Select **Generic Rest Config** as the connection type.
5. Click **Next**.
6. The **Connection Name** field is pre-populated with a default name. You can edit this value.
7. Use the **Config File** text box to upload the config file that you want to use.
8. In the **JDBC URL** textbox, enter the URL to connect to the server.
9. In the **User** and **Password** text boxes enter the user name and password for connecting to the REST endpoint. You may leave these fields blank if these values are not applicable or are already mentioned in the JDBC URL.
10. After providing all the required connection details, click **Test Connection** to test the connection.
11. Click **Create**.
The new connection is created.

The Data Load Page

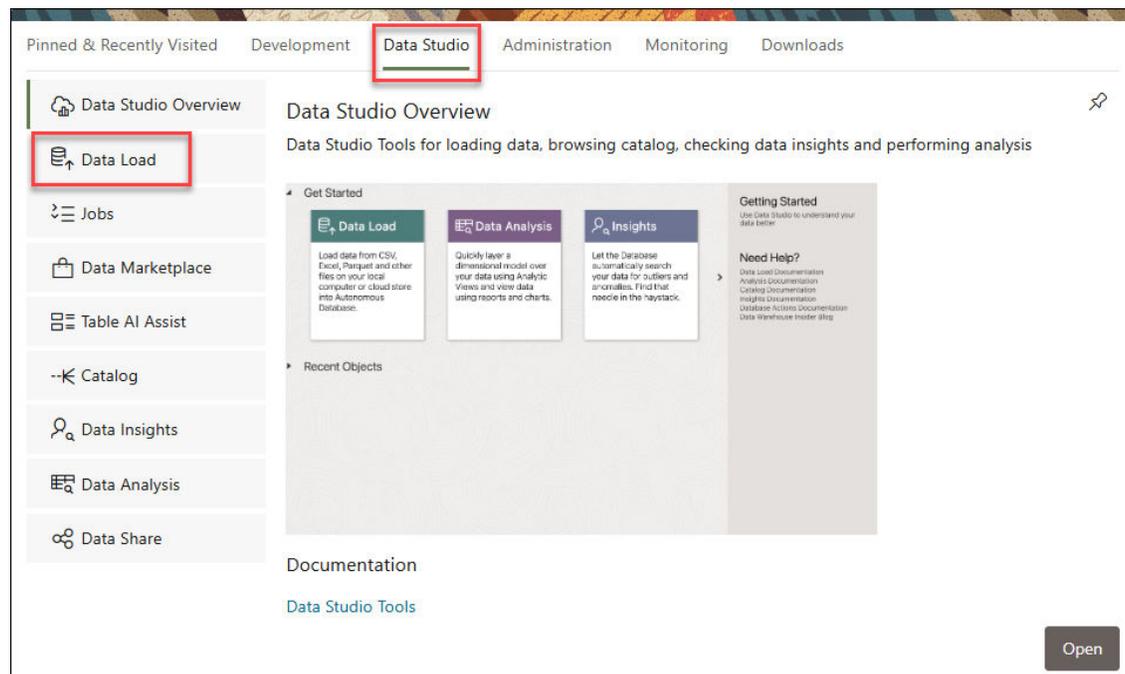
Use the Data Load dashboard page to make more data available to your Oracle Autonomous Database. You can load data from files or databases, from links to external databases or cloud storage files, or from a live feed of data from cloud storage.

From the Data Load page, you can also explore the data in your Autonomous Database and manage your cloud storage locations.

Note

If you do not see the Data Load card then your database user is missing the required DWROLE role.

To reach the Data Load page, click the **Data Studio** tab in the Database Actions page and select the **Data Load** pane.



You can alternatively click the Selector  icon and select **Data Load** from the Data Studio menu in the navigation pane.

The Data Load Dashboard provides a central location for all data load and live feed tasks. The various icons on the dashboard, provide access to loading data, linking data, setting up an ongoing feed, checking data load jobs, and managing connections.

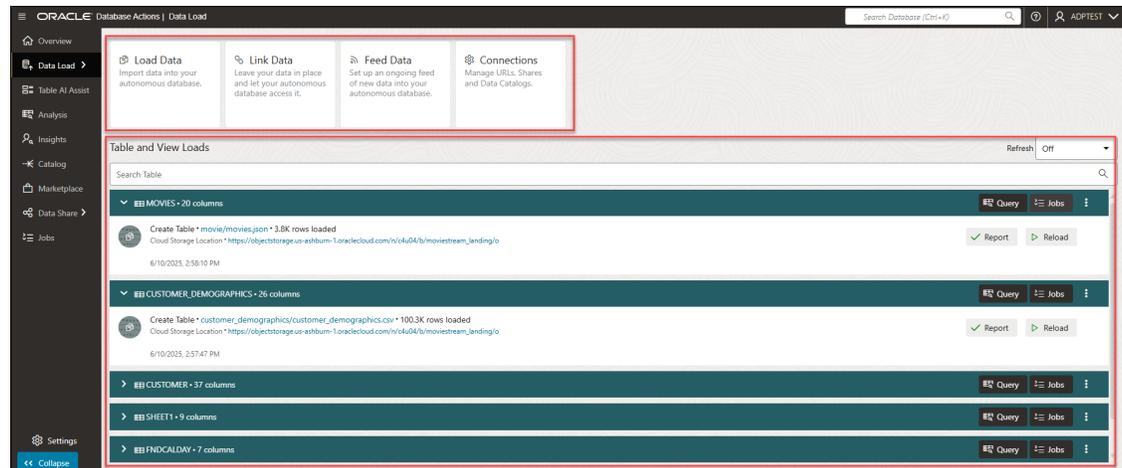
You can view the following tiles on the top section of the dashboard:

- Load Data
- Link Data
- Feed Data

- **Connections**

The bottom section of the dashboard displays the tables, views, and life feed you load into the tool.

The following image illustrates the **Data Load Dashboard**.



After drilling down onto a table, in the **Table and View Loads** section, you can view the details of the table load such as the name of the table, the number of rows and columns in the table, and the time you load the table.

Tables which are not active for last 7 days are collapsed.

You can also view the source location of the data you load. Click the location to view objects in the location.

Click the **Actions** icon beside the table name and select **Table** to view the following properties of the table:

- [View Details](#)
- [Gather Statistics](#)
- [Register to Cloud Link](#)
- [Publish Listings to Marketplace](#)
- [Create Analytic View](#)
- [The Table AI Assist Tool](#)
- [Export Data to Cloud](#)
- [The Data Analysis Tool](#)
- [Edit](#)
- **Drop** : Select **Drop** to delete the table.

Select **Report** to view a report of total rows processed successfully and failed for the selected table.

Select **Reload** to reload the source of the data load job. Clicking Reload takes you to the **Link Data** page where you can link your data source to the Autonomous Database.

Note

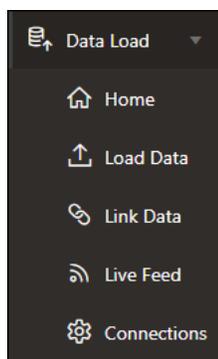
For tables you load via Cloud Store, you can view **Create Live Feed** icon. See [Create Live Feed from Data Load](#). After the creation of live feed is successful, you will view a Live Feed badge besides the table load header.

In case of errors, you can reload data with suggested fixes.

To load or create links to data or create live table feeds, on the Data Load dashboard, select a combination of an operation and a data source location. The following table lists the operations and the source locations that support those operations.

Operation	Source Location	Description
Load data	Local file Database Cloud storage Directory Share AI Sources	Load data from files on your local device, from remote databases, or from cloud storage into tables in your Oracle Autonomous Database.
Link data	Database Objects in Oracle Cloud Infrastructure (OCI) Catalog	Create external tables or views in your Oracle Autonomous Database that link to data in cloud storage or remote databases. Changes to the source data automatically appear in the target objects.
Feed data	Cloud storage	Set up a feed of data from a cloud storage bucket into a table. Changes to the source data load into the target table as scheduled or on demand.
Connections	Cloud storage Data Catalog links Share Providers	Create a cloud storage location. Subscribe to Share Provider and Register Data Catalog. You can manage cloud storage locations, Data Catalog and Shares.

The left navigation pane of the Data Load Dashboard provides links to the Data Load features and pages. When you select a link from the left navigation pane, the corresponding page appears in the target content area.



- Home: This is the current Data Load dashboard that you view.
- [Load Data](#)
- [Link Data](#)
- [Live Feed](#)
- [Connections](#)

Topics:

- [Data Studio Settings](#)
This topic explains the Data Studio settings on the Data Studio tool.
- [Managing Connections](#)
Connections that are established from the Data Studio to the cloud, catalogs and shares are listed on this page.
- [Loading Data](#)
You can load data from files on your local device, from remote databases, or from cloud storage buckets from directories and share providers. The file formats that you can load are CSV, XLS, XLSX, TSV, TXT, XML, JSON, GEOJSON, AVRO, PARQUET, GZ, GZIP, ZIP, PDF, PNG, JPG, JPEG and TIFF.
- [Linking Data](#)
You can link to data in remote databases or in cloud storage buckets.
- [Feeding Data](#)
You can run a live table feed on demand, on a schedule, or as the result of a notification.
- [DBMS_LIVE_FEED Package Reference](#)
This chapter provides information about the packages you use with the Data Load Tool in Data Studio. The `DBMS_LIVE_FEED` topic also covers the procedures included in the `DBMS_LIVE_FEED` package.

Create and Run Data Loads

A data load allows you to load multiple data entities from a source connection to a target connection.

- [Create a Data Load](#)
Use the Create Data Load wizard to define the source connection and the target connection for the data load.
- [Run a Data Load](#)
After you create the data load, you are taken to the Data Load Detail page that displays the details that you need to run a data load.

Create a Data Load

Use the Create Data Load wizard to define the source connection and the target connection for the data load.

Note

Data load is not supported for Oracle Object Storage connections.

The wizard also allows you to choose the load processing option based on the deployment type. The newly created data load is displayed in the Data Loads page of the associated

project. Click the Actions icon () next to the selected data load to edit, rename, start, export, or delete it.

To create a data load from the Home page:

1. Do one of the following:
 - On the Home page, click **Load Data**. The Create Data Load wizard appears. In the Create Data Load tab, enter a name if you want to replace the default value, add a description, and select a project from the drop-down.
 - On the Home page, click **Projects**, and then the required project tile. In the left pane, click **Data Loads**, and then click **Create Data Load**.

The Create Data Load wizard appears.

2. In the Name field, enter a name for the data load. The field is pre-populated with a default name. You can edit this value.
3. Add a description. This is optional.
4. If you have opened the Create Data Load wizard from the Home page, then you need to select a project name from the drop-down. If this your first time here, click the + icon to create a project. If you have logged in as SUPERVISOR, the default project name is Home. For other users, the default project name is in the format <username>_Home. You can edit the default value. See [Work with Projects](#) for more information about projects.
5. For **Load Processing** do one of the following:
 - Select the **Internal** radio button and from the **Deployment Type** drop-down select **Data Transforms (Batch)**.
 - Select the **Delegate** radio button and from the **Deployment Type** drop-down select OCI GoldenGate. From the GoldenGate Deployment Connection drop down select a connection.
6. Click **Next**.
7. To define your source connection, from the **Connection** drop-down, select the required connection from which you wish to add the data entities. Alternatively, click the + icon to create a new connection. See [Work with Connections](#) for more details about connections.
8. In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups:
 - Existing Schema (ones that you've imported into Oracle Data Transforms) and
 - New Database Schema (ones that you've not yet imported).

Select the schema that you want to use from the drop down.

Note

If there is missing information such as user name or password not specified, wallet missing, and so on, the list may fail to populate with a “This connection has missing information.” error. Click the Edit icon () to open the Update Connection page where you can fill in the missing details.

9. Click **Next**.

10. Similarly, define the target connection. If you plan to use an Apache Iceberg connection as a target to load data, see [Creating and Running an Apache Iceberg Data Load](#) for specific instructions.
11. Click **Save**.
The Data Load Detail page appears listing all the loaded data entities.

Run a Data Load

After you create the data load, you are taken to the Data Load Detail page that displays the details that you need to run a data load.

It includes the details of the source schema, the data entities that are loaded from the source schema, and the details of the target schema. You can choose the action that you want to apply on each data entity – recreate, truncate, append - and load tables in bulk to the target schema. When you run a data load, multiple jobs run in the background to complete the request.

You can run a data load in either of the following ways:

- [Using the Data Load Detail Page](#).
- [From the Workflow Details Page](#).

Note

Data load is not supported for Oracle Object Storage connections.

Running a Data Load from the Data Load Detail Page

The Data Load Detail page displays the information that you need to run a data load. You can apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema.

Note

APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

If the data load is huge, you might want to increase the memory of the ODI Agent to avoid any issues. Follow the instructions in [Increase the Memory of ODI Agent](#) before you start to run the data load.

To run a data load from the Data Load Detail Page:

1. In the Data Load Detail page, select the data entities that you want to move to the target schema.
To filter the list, you can do one of the following:
 - Enter the name or part of the name in the **Name** text box. This search returns data entities that include the specified string in the name. For example, the search string AD returns results such as ADMIN, ADDRESS, BADGE, UPGRADE, WORKLOAD, and so on.

- Turn on the **Use Regular Expression** toggle to use pattern matching to search for particular strings of characters. For example, the search string CO.* returns results such as CONTACT, COUNT, COUNTRY and so on.

You can select multiple data entities and load them to the target schema. You can also sort the displayed list using the following options:

- All - Displays all the data entities that match the search criteria.
- Selected - Displays all the rows that you selected.
- Unselected - Displays all the unselected rows.
- Invalid – Displays all the invalid rows.

Note

These options display the list of data entities based on the search criteria. To view the list of all data entities, clear any applied filters.

2. Click on the required icon to choose any of the following actions:

- Incremental Merge - Updates the data in the selected column by comparing the source table with the target table based on the specified merge key. To use this option, select the column that you want to merge and then select the merge key. To use this option, select the column that you want to merge and then select the merge key. Click the Validate icon (✓) to validate the selected values.

Note

- The last update values for the Incremental Column are stored in the Data Load. If you create a new data load with the same table, this option acts like Append, Truncate, and Recreate the first time you run it.
- If the target table is not truncated, do not start the Data Load using the Append, Truncate, or Recreate mode and then switch the action to Incremental Merge. This will cause every existing row in the target table to be updated. Oracle recommends that you start the Data Load using the Incremental Merge action.

- Incremental Append - Updates data in the selected column in the target schema. To use this option, select the column that you want to update and click the Validate icon (✓) to validate the selection.
The first time you run the Data Load, this option works in the same way as the Append, Truncate, and Recreate options, meaning all rows are inserted into the target. On all subsequent Data Load runs, the Incremental Column is used to filter only new rows to append data to avoid the same rows from being loaded more than once.

Note

- The last update values for the Incremental Column are stored in the Data Load. If you create a new data load with the same table, this option acts like Append, Truncate, and Recreate the first time you run it.
- If the target table is not truncated, do not start the Data Load using the Append, Truncate, or Recreate mode and then switch the action to Incremental Append. Oracle recommends that you start the Data Load using the Incremental Append action.

- Recreate – If the table is already present in the target schema, drops the existing table and recreates it.

Note

This option is not available for data entities that are loaded using OCI GoldenGate.

- Truncate – If the table is already present in the target schema, deletes all the data from the selected table. Nothing is dropped.

Note

For Delta Share data loads the Data Load Detail page only includes the  option. You cannot apply different actions - incremental merge, incremental append, recreate, truncate, append - on the data entities before loading it to the target schema. This is to make sure that the data is consistent between the Delta Sharing server and the target schema.

- Append – Inserts all rows in the Dataset into the target. If the Dataset contains records that already exist in the target, there will be duplicate records. Use Append if the source dataset is expected to have only new records or duplicate rows in the target are not a concern.
- Do Not Load – Skips the selected data entity from the data load job. After you click **Save**, these data entities are no longer available for future data load jobs.

You can select multiple data entities and apply different actions. The unsaved rows are highlighted in bold.

Note

These options are not available for Delta Share connections.

- To specify how you want to store the source column names in the target tables, click **Settings**, which is on the right side of the Data Load Detail page. Choose one of the following:
 - **Retain original names by enclosing all names with delimiters** - Creates column names with the same names as is from the source tables in the target table.
 - **Use no delimiters** - This is the default selection. Converts all the column names to upper case and replaces spaces and special characters with underscores.

The following options are applicable to reserved words such as `Date`, `Timestamp`, `Start`, and so on.

- **Enclose with delimiters** - This is the default selection. Encloses column names that are reserved words with delimiters (not all column names).
- **Use a prefix** - Adds the specified prefix to column names that are reserved words (not all column names).

For column names that have the same name after conversion, the names are suffixed with a numeric value to maintain uniqueness. For example column names

`Date`, `date`, `DATE`, `Item_@Code`, `Item$$Code`, `Item%%Code`

are created in the target table as

`DATE`, `DATE_0`, `DATE_1`, `ITEM__CODE`, `ITEM__CODE_0`, `ITEM__CODE_1`.

Note

Once the data load is run, the selected options are applied and retained for all subsequent runs. You cannot change the configuration.

4. Click  to save the changes. A green checkmark () in the row indicates that the changes are saved.
5. To start the data load,
 - Click .
 - For GoldenGate data loads, click .

A confirmation prompt appears when the data load starts successfully.

To check the status of the data load, see the Status panel on the right below the Target Schema details. For details about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel shows links to the jobs that execute to run this data load. Click the link to monitor the progress on the Job Details page. For more information about jobs, see [Create and Manage Jobs](#).

Running a Data Load from the Workflow Details Page

You can add multiple data loads to a workflow along with data flows or workflows and run them as separate steps. The left panel of the Workflow Details page lists the data flows, workflows, and data loads that are available for use.

For data loads, the left panel lists the following two folders:

- **Data Loads** - This folder lists all the data loads that you have created in the local Data Transforms instance. When you select a step in the workflow that is a local data load, the Properties Panel available on the right side of the design canvas displays the Type as `Data Load` and the Linked Object as `Home><nameofDataLoad>`.
- **Remote Data Loads** - This folder lists all the data loads that you have created in a remote Data Transforms instance. See [Create a Data Transforms Connection for Remote Data Load](#). When you select a step in the workflow that is a remote data load, the Properties Panel available on the right side of the design canvas displays the Type as `Remote Data Load` and the Linked Object as `Name of the Data Transforms connection ><nameofDataLoad>`.

After you have added the data loads to the workflow, click  to execute them.

Work with Data Entities

A Data Entity is a tabular representation of a data structure.

It includes Database Tables or Views that can be used in a mapping as a source or target. They are simply the metadata for sources and targets. They are used in creating data flows.

You can add Data Entities to your newly created project in one of the following two ways:

- Importing Data Entities
- Creating Data Entities

All the newly created or imported Data Entities along with their details are displayed in the **Data Entities** page. The details include:

- **Name** of the Data Entity
- **Connection** for which the Data Entity was created
- **Schema** to which the Data Entity is associated
- Click the **Actions** icon () next to the selected Data Entity to perform the following operations:
 - Select **Edit**, to edit the existing details.
 - Select **Preview**, to preview the selected Data Entity. If the data entity belongs to an Oracle database, you can also view statistics of the table. See [View Statistics of Data Entities](#) for more details.
 - Select **Delete**, to delete the selected Data Entity.
- To delete the Data Entities in bulk, in the Data Entities page, select the check boxes of the respective Data Entities and click **Delete**.
- You can also search for the required Data Entity to know its details based on the following filters:
 - **Name** of the Data Entity
 - **Connection** for which the Data Entity was created
 - **Schema** to which the Data Entity is associated
 - **Tag** that is associated with the Data Entity
- [Import Data Entities](#)
The easiest and most common way to create a Data Entity is by importing its structure from the connection type (for example, Oracle database).
- [Create Data Entities](#)
You can manually create or update the Data Entities through the Oracle Data Transforms interface.
- [Create Data Entities within the Data Flow editor](#)
If you have already created or imported your target data entity, then you would drag the data entity onto the Design Canvas and complete the column mappings and options.
- [View Statistics of Data Entities](#)
The Preview tab displays detailed statistics of each data entity.

Import Data Entities

The easiest and most common way to create a Data Entity is by importing its structure from the connection type (for example, Oracle database).

To import existing Data Entities:

1. From the left pane of the Home page, click the **Data Entities** tab. **Data Entities** page appears.
2. Click **Import Data Entities**, to import the existing data entities. **Import Data Entities** page slides-in. If you plan to use an Oracle ERP Cloud Server connection for the import, see [Importing Data Entity Definitions from an Oracle ERP Cloud Server](#).
3. Select the **Connection** followed by **Schema** and then finally select the **Type of Objects** you want to import.
For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.
4. [For Oracle Business Intelligence Cloud Connector (BICC) connections only] From the **Offerings to import for collection**, choose the offerings whose data stores you want to import. You must select at least one offering to import the BICC public view objects (PVO).

Note

The import of BICC PVOs can take a long time depending on the number of selected objects. To improve performance, Oracle recommends that you use a mask to filter and limit the number of PVOs that you want to import.

5. Choose a **Mask/filter** if you don't want to import every object in the schema. Depending on the Connection Type, you will be presented with further options for importing.

Note

For Oracle Object Storage connections, this value is case-sensitive. If **Batch similar files** is set to `True`, all the files that match the mask and have the same structure are grouped together into a single data entity.

6. [For Oracle Financials Cloud connections only] From the list in the **Resources** section, select the items that you want to import. When the import process completes, a table is created for each selected resource.
7. [For REST server connections only] In the **Resources** section, do the following:
 - In the **Resource URI** field, enter the URL of the REST service you want to import resources from.
 - Click the + icon.
 - In the **Name** column enter an identifier for the resource.
 - In the **Operation URI** column enter the URI of the resource.
 - Click **Test Resource** to check whether the entries are valid.
8. Click **Start**.

A Job is created and the corresponding Job ID is displayed for you to track the session. Click the Job ID to view the details of the job.

Upon successful execution of the job, all the selected Data Entities are imported. Click the Refresh icon  present at the right corner of the Data Entities page, to see the new imported Data Entities.

Create Data Entities

You can manually create or update the Data Entities through the Oracle Data Transforms interface.

Data entities should possess the corresponding objects in the source connection to be used as a source in a data flow. Usually the import process makes sure that these objects are in coordination. However, whenever you manually create or update Data Entities always make sure to check if both the definitions are in coordination with each other.

When you use a Data Entity as a target then it doesn't have to exist previously in the target connection and can be created as a part of Data Flow execution.

To create a new Data Entity:

1. From the left pane of the Home page, click the **Data Entities** tab. **Data Entities** page appears.
2. Click **Create Data Entity**, to create a new data entity. **Create Data Entity** page appears.
3. In the Name text box, enter the name of the new Data Entity that you are creating.
4. From the **Connection** drop-down, select the required connection from which you wish to add the newly created Data Entity.

Note

Oracle Financials Cloud connections are not listed here because you cannot manually create data entities for such connections. You can only import data entities from Oracle Financials Cloud REST endpoints using the Import Data Entities page. See [Import Data Entities](#).

5. In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.
 - New Database Schema (ones that you've not imported from before) and
 - Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

From the **Schema** drop-down, select the required schema.

Note

For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.

6. From the **Type** drop-down, select the data entity type.

- **Table:** To define the table structure for the newly created Data Entity, click the + icon to add columns. For each column, you can specify parameters such as Name, Data Type, Length, Scale, Not Null. Double click on the cell to configure the value. Click the 'x' icon, to delete a row. Click the Up and Down arrows to sort the table rows.
 - **Inline View:** To create the data entity using inline code, enter the Select statement in the **Query** tab. For example, `SELECT * FROM CUSTOMER`. Click **Validate**. The **Columns** tab displays a read-only list of the columns that the query returns. Click the **Preview** tab to see the column data.
7. In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
 8. For Oracle Object Storage connections, this page displays the following options:
 - **Contents** – Select the CSV file that contains the data you want to import. The metadata displayed in the preview table, such as the data type and length of columns, is based on the first row of the CSV file. Make sure that the CSV file has a header line. The header should contain only alphanumeric characters and no special characters.
 - **Group Files** – Select this check box if you want to group data from multiple CSV files into one data entity. For example, say you want to merge data from `Employee_Data1.csv`, `Employee_Data2.csv`, and `Employee_Data3.csv` into a single data entity.
If you want to upload the CSV files using Windows, make sure that you first convert the files to Unix format to avoid character encoding issues during grouping.
 - **Resource Name** – Use this option along with **Group Files**. Enter the value you want to use to identify the files. The resource name should be a regular expression. You can use only an asterisk (*) as a wildcard character in the resource name. For example, `Employee_Data*.csv`.
 9. For Oracle database connections, you can mark the data entities as a feature group. Expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
 10. Click **Save**.
The new Data Entity is created.

Create Data Entities within the Data Flow editor

If you have already created or imported your target data entity, then you would drag the data entity onto the Design Canvas and complete the column mappings and options.

To create the definition of an entity while in the Data Flow editor,

1. Drag the data entity onto the Design Canvas.
2. Select the component and click the Add Data Entity icon  present on the top right corner of the target component.
3. **Add Data Entity** page appears allowing you to configure the following details of the target component:

General tab

 - In the **Name** text box, enter the name of the newly created Data Entity.
 - From the **Entity Type** drop-down, select the data entity type.
 - **Table**
 - **ML Model**
 When you select this entity type the user interface changes as follows:

- * The **Connection Type** drop-down only lists **Oracle** as the option. Consequently, the **Connection** drop down only lists the Oracle connections that you have created.
 - * The Add Data Entity wizard displays the **Properties** tab where you can select the Type of Learning, Function, Algorithm, and configure parameters to define the ML Model. See [Use Machine Learning \(ML\) Models](#) for detailed information about creating and using an ML Model data entity.
- From the **Connection Type** drop-down, select the required connection from which you wish to add the newly created Data Entity. The Connection drop-down is populated with the connections you have created with the associated connection type.
 - From the **Connection** drop-down, select the server name from which you wish to add the newly created Data Entity.
 - In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.
 - New Database Schema (ones that you've not imported from before) and
 - Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

From the **Schema** drop-down, select the required schema.

Note

For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.

- In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
- For Oracle database connections, you can mark the data entities as a feature group. Expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
- Click **Next**.

Columns tab



- Click the  Add Columns icon, to add new columns to the newly created Data Entity. A new column is added to the displayed table.
- The table displays the following columns:
 - Name
 - Data Type - Click the cell to configure the required Data Type.
 - Scale
 - Length
 - Actions - Click the cross icon to delete the created column.
- To delete the columns in bulk, select the columns and click the  Delete icon.
- To search for the required column details, in the Search text box enter the required column name and click enter. The details of the required column are displayed.

- Click **Next**.

Preview Data Entity tab

It displays a preview of all the created columns and their configured details. If the data entity belongs to an Oracle database, you can also view statistics of the table. See [View Statistics of Data Entities](#) for more information.

4. Click **Save**.
The new target Data Entity is created.
5. Expand the **Properties Panel** in the right pane to view the following settings of the created components:
 - General - Displays the Name of the component along with its Connection and Schema details.
 - Attributes - Displays the details of all the attributes associated to the component.
 - Column Mapping - Click Auto Map to map all the columns automatically.
 - Preview - Click to have a preview of the component.
 - Options - Change the options as appropriate.

View Statistics of Data Entities

The Preview tab displays detailed statistics of each data entity.

Note

This feature is available for Oracle database tables only.

You can view the statistics of a selected data entity in one of following ways:

- In the Data Entities list, click the **Actions** icon () next to the Data Entity and click **Preview**. Select the **Statistics** tab to view the statistics of the selected data entity.
- On any data flow click on any source or target data entity, and expand the properties panel in the right pane. Click **Preview**.

The statistical data is presented as follows:

- The total number of rows and columns in the data entity is displayed at the top.
- The statistics panel displays the thumbnail graphs for each column with information about the Min, Max, Distinct, and Null values.
- Two types of thumbnail representations are displayed based on the histogram:
 - A bar chart represents data for frequency and top-frequency histograms. The bar chart show the first top 10 values for the number of rows in the table.
 - A table lists data for Hybrid and Height-Balanced histograms. The table displays the entire data and is scrollable. The table displays the range for the values and the percentage of rows in each range.
- You can click each thumbnail to view the statistics of the column in a new browser tab.
- The detailed view of each chart also shows the type of histogram.

Work with Projects

A project is the top-level container, which can include multiple folders to organize your data flows or work flows into logical groups.

The Projects page displays all the existing projects, which includes projects created by all users not just the logged in user.

You can perform the following operations on a Project folder:

- View Details
- Edit the name
- Export the Project. See [Export Objects](#).
- Delete

Creating a Project

To create a new project:

1. Click **Create Project** on the Projects page.
2. On the **Create Project** page, provide a project name.
3. Click **Create**.
After creating the project, you're redirected to the **Projects** page.

Viewing Project Details

On the Projects page, select a project and click **View Details** from the project's **Actions** icon

() or select a project to open its details page. The left panel displays all the resources associated with the selected project such as data loads, data flows, workflows, variables, schedules, and jobs.

Deleting a Project

When you delete a project, you also delete the resources it contains. Once you delete a project, it cannot be restored.

Note

Be sure to review all contained resources before you delete the project.

To delete a project, on the Projects page, select Delete from the **Actions** icon () for the project you want to delete or select a project to open its details page, and then click **Delete**.

In the **Delete Project** dialog, click **Delete** to confirm the delete operation.

About Data Flows

A data flow defines how the data is moved and transformed between different systems.

A data flow in Data Transforms connects sources to targets through a flow of components such as Join, Filter, Aggregate, Set, Split, and so on. See [Supported Database Functions](#) for more information.

When you run a data flow, Data Transforms uses the joins, filters, mappings, and constraints to transform source data and load it to target tables. Note that you can run only one execution flow at a time. You cannot put multiple flows on a Data Flow and a flow cannot diverge into multiple flows.

Topics

- [Create a Data Flow](#)
Create data flows to load data from a source connection, run transformations, and move the data to a target database.
- [About Data Flow Editor](#)
The Data flow editor is divided into five parts, the Data Entity Panel, the Database Functions Toolbar, the Design Canvas, the Properties Panel, and the Status Panel.
- [Supported Database Functions](#)
Oracle Data Transforms supports various database functions that you can drag and drop on the Design Canvas to connect components within a data flow.
- [Add Components](#)
Add the data entities and database functions to the Design Canvas, and connect them in a logical order to complete your data flows.
- [Use Text Embedding Vector in a Data Flow](#)
Data Transforms supports the use of vector datatype and embedding vectors in a data flow. Currently, Data Transforms integrates with OCI Generative AI service to convert input text into vector embeddings that you can use for data analysis and searches.
- [Component Properties](#)
The Properties Panel displays various settings for components selected in the Design Canvas.
- [Map Data Columns](#)
When you connect the source data entity with the target data entity, the column names are automatically mapped by the column names. You have a choice to map the columns by Position or by Name or map the columns manually using the Expression Editor.
- [Validate and Execute a Data Flow](#)
After your mappings are ready, you can proceed to validate and execute the data flow.

Create a Data Flow

Create data flows to load data from a source connection, run transformations, and move the data to a target database.

You can create data flows in any of the following ways:

- [From the Projects page](#)
- [From the Data Flows page within a project](#)
- [From the Home page](#)

The newly created data flow is displayed in the Data Flows page of the associated project.

Click the Actions icon () next to the selected data flow to edit, rename, copy, change folder, start, export, or delete it.

From the Projects page

To create a data flow from the Projects page,

1. On the Projects page, click **Create Data Flow**.
Create Data Flow page appears:
2. In the **Name** field, enter a name for the new data flow.
3. Select **Create New Project**, if you wish to create new project folder for the newly created data flow.
4. Else, click **Add to Existing Projects**, if you wish to add the newly created data flow to an existing project folder.
5. If you have selected **Create New Project** for the previous option, in the **Project Name** field, enter the name of the newly created project.
6. Else, if you have selected **Add to Existing Projects** for the previous option, select the required project from the **Project Name** drop-down arrow.
7. In the **Description** field, enter a description for the newly created data flow.
8. Click **Create**.

From the Data Flows page within a project

To create a data flow from the Data Flows page within a project,

1. On the Projects page click the project tile you wish to create a new data flow for. The **Project Details** page appears.
2. In the **Data Flows** page, click **Create Data Flow**.
3. Provide the **Name** and **Description** of the new data flow.
4. Click **Next**.
5. To define your source connection, from the **Connection** drop-down, select the required connection from which you wish to add the data entities.
6. In the Schema drop-down, all schema corresponding to the selected connection are listed in two groups:
 - Existing Schema (ones that you've imported into Oracle Data Transforms) and
 - New Database Schema (ones that you've not yet imported).

Select the schema that you want to use from the drop down. For Oracle Object Storage connections, the Schema drop-down lists the name of the bucket that you specified in the URL when you created the connection.

7. Click **Save**.
The Data Flow Editor appears that allows you to create a new data flow.

From the Home page

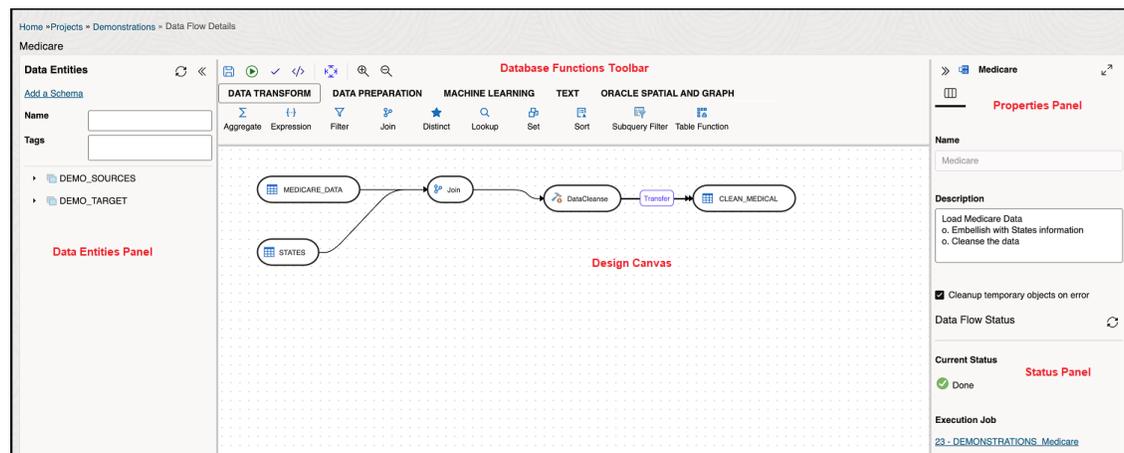
To create a data flow from the Home page,

1. On the Home page, click **Transform Data**. The Create Data Flow page appears.
2. Provide the **Name** and **Description** of the new data flow.

3. Select a project name from the drop-down. Alternatively, click the + icon to create a project.
4. Click **Next**.
5. From the **Connection** drop-down, select the required connection from which you wish to add the data entities. Alternatively, click the + icon to create a new connection.
6. In the Schema drop-down, all schema corresponding to the selected connection are listed in two groups:
 - Existing Schema (ones that you've imported into Oracle Data Transforms) and
 - New Database Schema (ones that you've not yet imported).
 Select the schema that you want to use from the drop down.
7. Click **Save**.

About Data Flow Editor

The Data flow editor is divided into five parts, the Data Entity Panel, the Database Functions Toolbar, the Design Canvas, the Properties Panel, and the Status Panel.



- **Data Entities Panel:** The data entity panel displays the Data Entities that are available to use in your Data flows. The displayed list can be filtered using the Name and Tags fields. The panel includes options that let you add schemas, import data entities, remove any of the schemas that are associated with the data flow, and refresh data entities. See [Add Components](#) for information about how to use these options.
- **Database Functions Toolbar:** The Database Functions toolbar display the database functions that can be used in your data flows. Just like Data Entities, you can drag and drop the Database tools you want to use on the design canvas. See [Supported Database Functions](#) for more information.
- **Design Canvas:** The design canvas is where you build your transformation logic. After adding the Data Entities and Database Functions to the design canvas, you can connect them in a logical order to complete your data flows.
- **Properties Panel:** The properties panel displays the properties of the selected object on the design canvas. The Properties Panel is grouped into four Tabs. **General, Attributes, Preview Data, Column Mapping, and Options.** Not all tabs are available as they vary based on the selected object. See [Component Properties](#) to know more about these options.

- **Status Panel:** When you run a data flow, the Status Panel shows the status of the job that is running in the background to complete the request. You can see the status of the job that is currently running or the status of the last job. For more information about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#).

After designing the required data flow,

- Click , to save the created/designed data flow.
- Click , to align the nodes of the designed data flow.
- Click , to execute the created data flow.
- Click , to validate the created data flow.
- Click  , to maximize or minimize the created data flow diagram in the design canvas.

Supported Database Functions

Oracle Data Transforms supports various database functions that you can drag and drop on the Design Canvas to connect components within a data flow.

The Database Functions toolbar in the Data Flow editor includes the following database functions that can be used in your data flows. See Oracle Database SQL Language Reference for information about the database functions.

1. Data Transformation

It contains the following components:

- Aggregate
- Expression
- Filter
- Join
- Distinct
- Lookup
- Set
- Sort
- Subquery Filter
- Table Function

2. Data Preparation

It contains the following components:

- Data Cleanse
- Substitution
- Equi_Width Binning
- Quantile Binning
- Lead
- Lag

- Replace

3. Machine Learning

It contains the following components:

- Prediction
- Prediction Model
- Outlier Detection
- Text Embedding Vector

4. Text

It contains the following components:

- REGEXP COUNT
- REGEXP INSTR
- REGEXP SUBSTR
- REGEXP REPLACE
- Edit Distance Similarity
- Contains

5. Oracle Spatial and Graph

It contains the following components:

- Buffer Dim
- Buffer Tol
- Distance Dim
- Distance Tol
- Nearest
- Simplify
- Point
- Geocode Tools:

Note

The following Geocode Tools work only in non-Autonomous Database environment.

- Geocode As Geometry
- Geocode
- Geocode Addr
- Geocode All
- Geocode Addr All
- Reverse Geocode

Note

The following Geocode Tool works only in an Autonomous Database environment.

- Geocode Cloud
- Spatial Join

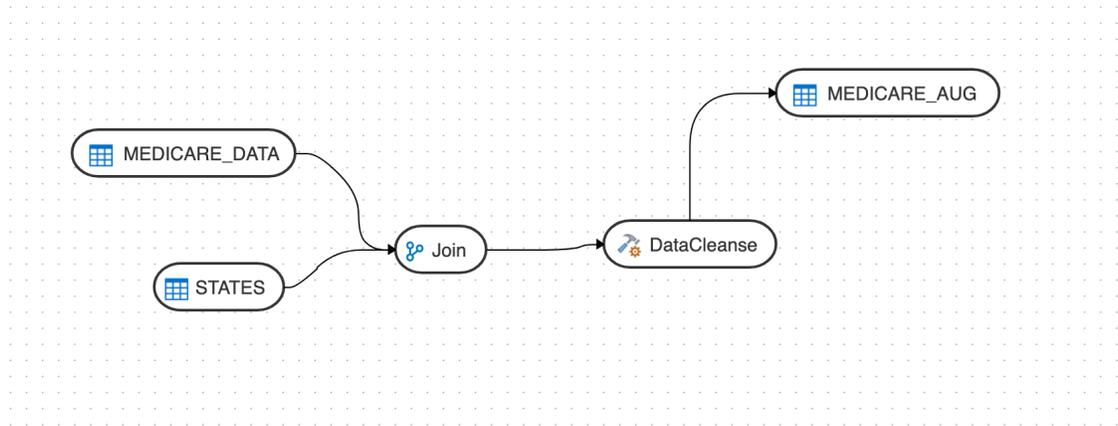
Add Components

Add the data entities and database functions to the Design Canvas, and connect them in a logical order to complete your data flows.

To add components to your data flow:

1. In the Data Entities panel, click **Add a Schema** to add schemas that contain the data entities that you want to use in the data flow.
2. In the Add a Schema page, select the connection and schema name.
3. Click **Import**.
4. In the Import Data Entities page, select the Type of Objects you want to import. Choose a Mask/filter if you don't want to import every object in the schema and click **Start**.
5. The Data Entities panel lists the imported data entities. The panel includes various options that let you do the following:
 - Refresh Data Entities – Click the Refresh icon  to refresh the displayed list.
 - **Name** - Search for data entities by name.
 - **Tags** - Filter the data entities by the name of the tag used.
 - **Import Data Entities** - Right-click the schema to see this option. Use this option to import the data entities.
 - **Remove Schema** - Right-click the data entity to see this option. Use this option to remove the schema from the list. Note that this option does not delete the schema, it only removes the association of the schema with this data flow.
6. Similarly add more schemas to the Data Flow, if required.
7. Drag the required Data Entities that you want to use in the data flow and drop them on the design canvas.
8. From the **Database Functions** toolbar, drag the transformation component that you want to use in the data flow and drop them on the design canvas. You can use variables in the data flow. See [Use Variables in a Data Flow](#) for more information.
9. Select an object on the design canvas, and drag the **Connector** icon () next to it to connect the components.
10. After saving the data flow, there may be a Transfer icon overlaid on one or more of the component connections. This indicates that ODI has detected an additional step and it is required to move the data between data servers. You can click on this Icon to view properties associated with this step.

For example:



Use Text Embedding Vector in a Data Flow

Data Transforms supports the use of vector datatype and embedding vectors in a data flow. Currently, Data Transforms integrates with OCI Generative AI service to convert input text into vector embeddings that you can use for data analysis and searches.

For text embedding Data Transforms supports both the text stored in a column and from the http links stored in a column. Before you use embedding vectors in a data flow, you need to do the following:

- Create an Oracle Database 23ai connection. See [Work with Connections](#) for generic instructions on how to create a connection in Data Transforms.
- Create an Oracle Cloud Infrastructure (OCI) Generative AI connection. See [Create and use an Oracle Cloud Infrastructure Generative AI Connection](#).

To use vector embeddings in a data flow:

1. Follow the instructions in [Create a Data Flow](#) to create a new data flow.
2. In the Data Flow Editor click **Add a Schema** to define your source connection. From the **Connection** drop-down, select the Oracle Database 23ai connection and the schema that you want to use from the drop down. Click **OK**.
3. Drag the tables that you want to use as a source in the data flow and drop them on the design canvas.
4. From the Database Functions toolbar, click **Machine Learning** and drag the **Text Embedding Vector** transformation component drop it on the design canvas.
5. Click the **Text Embedding Vector** transformation component to view its properties.
6. In the **General** tab, specify the following:
 - **AI Service** - Select **OCI Generative AI** from the drop-down.
 - **Connection** - The drop-down lists all the available connections for the selected AI Service. Select the connection that you want to use.
 - **AI Model** - The drop-down lists all the available models for the selected AI Service and Connection. The following models are listed:
 - "cohere.embed-english-light-v2.0"
 - "cohere.embed-english-light-v3.0"
 - "cohere.embed-english-v3.0"

- "cohere.embed-multilingual-light-v3.0"
- "cohere.embed-multilingual-v3.0"

You can also type in the model name.

7. In the **Column Mapping** tab, map the source column that you want to embed to the INPUT attribute of the operator. The only column available in the column mappings is `input_text`. Drag a text column from the available columns to the Expression column. This is the data that the vectors will be built on.
8. Drag the table that you want to use as a target in the data flow and drop it on the design canvas.
9. Save and execute the data flow.
Data Transforms will build vectors for each of the rows in the source table and write that to the target table.

Component Properties

The Properties Panel displays various settings for components selected in the Design Canvas.

Depending on the component selected, you may see any of the following icons:

- General () - Displays the name of the component along with its connection and schema details. You can edit some of these properties.
- Attributes () - Displays the details of all the attributes associated with the component.
- Column Mapping () - Allows you to map all the columns automatically. See [Map Data Columns](#) for more information.
- Preview () - Displays a preview of the component. For Oracle tables, you can also view the statistics of the selected data entity. See [View Statistics of Data Entities](#) for details about the statistical information available.
- Options () - Displays options such as
 - Truncate Table - Replaces any existing target table content with new data.
 - Append - Inserts records from the flow into the target. Existing records are not updated.
 - Incremental - Integrates data in the target table by comparing the records of the flow with existing records and updating the records when their associated data is not the same. Those that don't yet exist in the target are inserted.
The option includes an Auto compression feature that is set to `TRUE` by default. For data flow jobs that use the Incremental Update mode to load data onto a compressed Oracle target partition, the Auto compression feature recompresses the modified target partitions after the load completes successfully. For table partitions that are not originally compressed, the compression is skipped irrespective of whether Auto compression is set to true.

Note

The Auto compression option is available to the ADMIN user or to a user with the DWROLE role. For data flows that have schema users other than ADMIN you need to either assign the DWROLE to the user or disable Auto compression to avoid execution errors.

Map Data Columns

When you connect the source data entity with the target data entity, the column names are automatically mapped by the column names. You have a choice to map the columns by Position or by Name or map the columns manually using the Expression Editor.

To map columns by Position or by Name:

1. Select the target Data Entity.
2. Click the arrow icon present on the top right corner to expand the Properties Panel. This will give you more space to work with.
3. In the **Properties Panel**, click the **Column Mapping** icon ().
4. To map the columns by Position or by Name, from the **Auto Map** drop-down menu, select **By Position** or **By Name**.

To map the columns manually:

1. From the **Auto Map** drop-down menu, select **Clear** to clear the existing mappings.
2. Drag and drop the attributes from the tree on the left to map with the Expression column.
3. To edit an expression, click the **Edit** icon of the respective column. The **Expression Editor** appears allowing you to perform the required changes (for example, you can just add an expression-"UPPER" or open the Expression Editor to edit the expression).

Note

Use the expression editor only if you have complex expressions for a particular column.

4. Click **OK**.

Validate and Execute a Data Flow

After your mappings are ready, you can proceed to validate and execute the data flow.

Do the following:

1. Click **Save**.
After saving, if data needs to be staged before transforming, Transfer button is added to one or more links. You can click these buttons to set more options, if available.
2. Click the **Code Simulation** icon () if you want to check the code that will run to complete the tasks that are performed when you execute the data flow job. The source and target details are displayed in different colors for ease of reference. This is handy if you want to check if the mapping is correct before you run the job or if the job fails. Note that

the code cannot be used for debugging. For detailed information about the job, see the Job Details page.

3. Click the **Validate** icon (✓) in the toolbar above the design canvas to validate the data flow.
4. After a successful validation, click the **Execute** icon (▶) next to the **Validate** icon to execute the data flow.
If you have added variables to the data flow, the **Variable values** page appears that displays the list of variables that you have added to the data flow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the data flow. The custom value does not persist for any subsequent sessions.

A message appears that displays the execution Job ID and name. To check the status of the data flow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel also shows the link to the Job ID that you can click to monitor the progress on the Jobs page. For more information, see [Create and Manage Jobs](#).

For data flows created using Oracle Object Storage connections, the data from the source CSV file is loaded into the target Oracle Autonomous Database. You can also export data from an Oracle Autonomous Database table to a CSV file in Oracle Object Storage.

Schedule Data Flows or Workflows

You can schedule workflows and data flows to run at specified time interval.

Each schedule allows you to specify a start date and time, and execution frequency.

To schedule a data flow or a workflow:

1. In the left pane, click **Schedules**.
2. Click **Create Schedule**.
3. From the **Resource** drop-down menu, select **Data Flow** or **Workflow** as appropriate.
4. From the **Resource Name** drop-down list, select a data flow or workflow that you want to schedule.
5. From the **Frequency** drop-down list, select the time frame in which you wish to execute the created schedule. You can schedule it On Startup, Simple, Daily, Hourly, Weekly, Monthly or Yearly. Based on the selected frequency, **Time** field appears allowing you to schedule the selected Data Flow or Workflow. For detailed information refer to the below table:

Frequency	Time Values
On Startup	NIL
Simple	Click Select Date Time icon next to Date and Time field, to select the required date and time in MM/DD/YY and HH:mm format.
Daily	Click the clock icon next to Time field, to select the required time in HH:mm format.
Hourly	Click up and down arrows next to Time field, to select the required time in Minutes:Seconds format.

Frequency	Time Values
Weekly	<ul style="list-style-type: none"> • Run Every parameter appears with check-boxes for all the days of the week. • Click the clock icon next to Time field, to select the required time in HH:mm format.
Monthly (day of the month)	<ul style="list-style-type: none"> • Select the required date from the Monthly Date drop-down box. • Click the clock icon next to Time field, to select the required time in HH:mm format.
Monthly (week day)	<ul style="list-style-type: none"> • From the Monthly date drop down box select the required value. • From the Week day drop-down box, select the required weekday in which you wish to schedule the job.
Yearly	<ul style="list-style-type: none"> • From the Month drop down box, select the required month in which you wish to schedule the job. • From the Monthly date drop down box, select the required date in which you wish to schedule the job. • Click the clock icon next to Time field, to select the required time in HH:mm format.

6. In the **Number of Attempts on Failure** text box, enter the required value or click the up or down arrows next to the text box, to select the required value. This value denotes the number of retry attempts that should happen after the schedule failure.
7. **Stop Execution After** field denotes the time after which the schedule has to stop after executing it. It can be in **Hours**, **Minutes** or **Seconds**. Select the value and its unit.
8. Select the status of the scheduled Data Flow or Workflow from the **Status** options. It can be **Active**, **Inactive** or **Active for Period**. Through **Active for Period** option you can configure the exact time frame of the created schedule. It's Starting and Ending Date with time, time interval in which you wish to run the schedule and exceptions, if any.
9. After configuring the above details, click **Save**.
The created schedules get listed in the **Schedules** page along with the following details:
 - Resource
 - Status
 - Frequency
 - Valid or Invalid

Click the Actions menu () of the respective schedule to perform the following operations:

- Click **Edit** to edit the details of the created schedule.
- Click **Disable** to disable the created schedule. The status of a schedule becomes inactive when you disable it and gets displayed in the Scheduled list. Select the schedule and click **Enable**, to enable it whenever required. You can enable it again for a specific period, through **Enable for Period** option.
- Click **Validate** to validate the created schedule.
- Click **Export** to export the schedule. See [Export Objects](#).

- Click **Delete** to delete the created schedule. Upon confirmation the created schedule is deleted.

To schedule a data flow or a workflow specific to a project:

- Click the Project title displayed on the **Projects** page. **Project Details** page appears.
- In the left pane, click **Schedules**. **Schedules** page appears. It displays all the schedules pertaining to the selected project.

Note

This page is project specific and displays only the Data Flows and Workflows pertaining to the selected project.

- To create a new schedule, click **Create Schedule**.
- Enter all the required details and click **Save**.
The new schedule is created and is added to the existing list of schedules for the project.

Monitor Status of Data Loads, Data Flows, and Workflows

When you run a data load, data flow, or workflow Oracle Data Transforms runs jobs in the background to complete the request. You can view the status of the job in the panel on the bottom right of the Data Load Details, the Data Flow Editor, and the Workflow Editor page.

This panel includes the following:

- A Refresh icon () to refresh the displayed status.
- The current status. You can see any of the following statuses:
 - Not Started – If no data load, data flow, or workflow has been executed yet.
 - Running – When the job starts.
 - Done – When the job completes.
The status of the last job run persists in this panel till you run a new data load, data flow, or workflow.

Note

For data loads created using OCI GoldenGate, the status panel shows the link to the GoldenGate Deployment Console and the status of the Extract and Replicat processes.

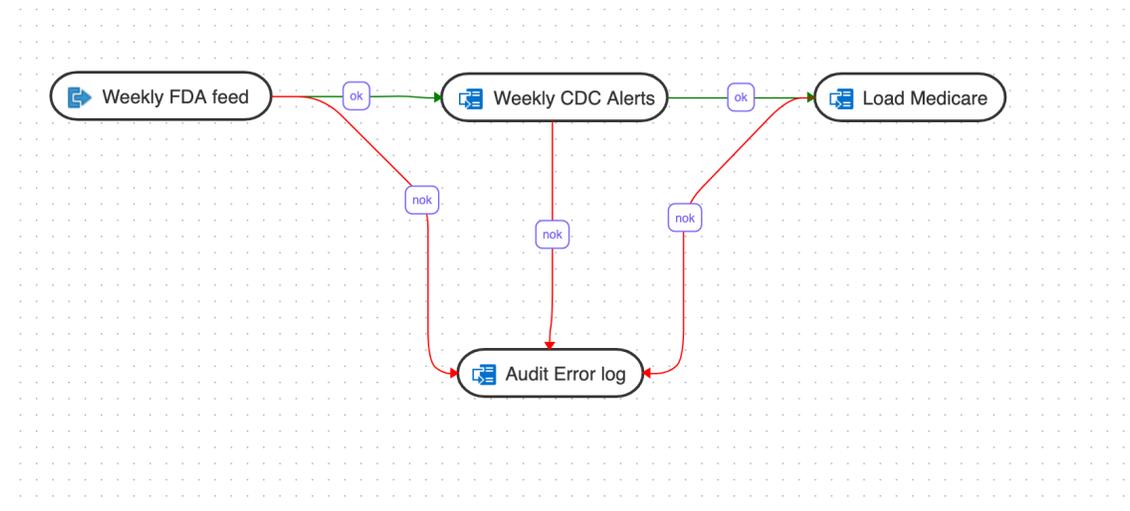
- The name of the job and a link that takes you to the Jobs Details page where you can see detailed information about the job and monitor the execution status. For more information, see [Create and Manage Jobs](#).
When you run a data load, multiple jobs run in the background to complete the request. This panel shows a link for each job that executes when running a data load.

Introduction to Workflows

A workflow is made up of multiple flows organized in a sequence in which they must be executed.

Each data flow is executed as a step. You can also add workflows, SQL queries, data loads, as well as variables as steps within a workflow. When you execute a workflow, a step either succeeds or fails. Depending on whether the first step succeeds or fails, you can choose the next step that must be executed.

Here is an example of a workflow:



In this example, the workflow performs the following actions:

1. Execute the "Weekly FDA feed" data flow.
2. If the "Weekly FDA feed" data flow execution is successful, execute the "Weekly CDC alerts" data flow.
3. If the "Weekly CDC alerts" data flow execution is successful, execute the "Load Medicare" data flow.
4. If any of the above data flow fails, then execute the "Audit_error_log" data flow.

Topics

- [Create a New Workflow](#)
You can add data flows, workflows, variables, or data loads in a workflow.

Create a New Workflow

You can add data flows, workflows, variables, or data loads in a workflow.

To create a new workflow for your project:

1. On the **Home** page, click the required Project title.
You are navigated to the **Project Details** page.
2. In the left pane, click **Workflows**.
The Workflow page appears.
3. On the **Workflow** page, click **Create Workflow**.
The **Create Workflow** page appears.
4. In the **Name** field provide a name for the new workflow and click **Create**.
The new workflow is created and listed in the **Workflow** page with its **Name** and **Folder**

details. Click the Actions icon () next to the workflow to edit, rename, copy, change folder, start, export, or delete it.

5. Click the Workflow to configure the **Workflow Details**.
6. From the left panel drag the data flows, workflows, variables, or data loads that you want to run in the workflow. To use variables in the workflow, see [Use Variables in a Workflow](#) for more information.

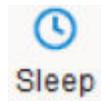
If you have connected to any other Data Transforms instance, you can also add data loads that you have created in that Data Transforms instance. See [Create a Data Transforms Connection for Remote Data Load](#) and [Run a Data Load](#) for more information.

7. Select either the ok (green arrow) icon, the not ok (red arrow) or the ok/not ok (black arrow) in the toolbar.

This defines the mode that subsequent links drawn on the canvas will be set to.

- ok (green) defines the success path.
- not ok (red) defines the failure path.
- ok/not ok (black) defines a path that is followed on success or failure.

8. Use the Sleep icon (



) to add a delay in the workflow. Drag the Sleep icon on the canvas and connect it in the flow with either the ok (green), not ok (red) or ok/not ok (black) links. This will add a delay at that point in the flow.

9. If you want to add a SQL or PL/SQL query as a step in the workflow, do the following:



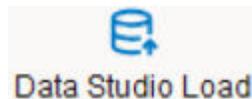
- Drag the SQL icon () on the canvas.
- Double click the SQL step in the editor to open the step properties page.
- Select the **Attributes** tab.
- From the **Connection** drop-down select the connection you want to run the query on.

Note

The drop-down lists only Oracle database connections.

- In the **SQL** textbox add the query that you want to run.

10. If you want to add a Data Studio Data Load as a step in the workflow, use the Data Studio Load icon (



) on the canvas. See [Define a Data Studio Data Load in a Work Flow](#) for detailed instructions on how to use a Data Studio Data Load to load data to Autonomous Database.

11. Select the step and click the **Connector** icon () next to it to connect it with the next step.

12. After defining all the required Workflow details,

- Click  , to save the created/designed workflow.
- Click  , to align the nodes of the designed workflow.
- Click  , to execute the created workflow.
If you have added variables to the workflow, the **Variable values** page appears that displays the list of variables that you have added to the workflow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the workflow. The custom value does not persist for any subsequent sessions.
- Select a single step in the canvas and click the **Execute Step** icon (), to execute only the selected data flow or workflow.
To check the status of the workflow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel shows the link to the Job ID that you can click to monitor the execution status on the Jobs page. For more information about jobs see [Create and Manage Jobs](#).
- Click   , to maximize or minimize the created workflow diagram in the design canvas.
The newly created workflows get listed in the Project Details page. The following details are displayed:

- Name of the Workflow
- Folder corresponding to the workflow - Default Folder

To view the details of a workflow, click the name of the workflow and you are navigated to the Workflow Details page.

In the left pane, you can search for the required Data Flow or Workflow using the Name filter. In the **Name** text box, enter the name of the required Data Flow or Workflow.

Select a step in the canvas and check the **Properties Panel** available at the right side of the design canvas to know the following properties of the selected step in a created Data Flow or Workflow:

- Name
- Linked Object

 **Note**

You cannot edit this field.

- Step -
 - First Step - Select the **First Step** check-box, to execute the selected step as the first step for execution in a Data Flow or Workflow.

Note

You can select only a single step as a first step for execution in a Data Flow or Workflow.

- Number of attempts on Failure
- Time between attempts in seconds(s)
- Log steps in journal - Select any of the following options from the drop-down box:
 - Always
 - Never
 - Error
- [Define a Data Studio Data Load in a Work Flow](#)
You can add a Data Studio Data Load as a step in a workflow and run the data load at scheduled intervals.

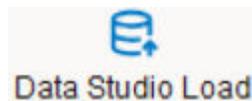
Define a Data Studio Data Load in a Work Flow

You can add a Data Studio Data Load as a step in a workflow and run the data load at scheduled intervals.

To do this, you need to create a connection, import data entities, create a data flow and then schedule the data load run.

To define and run a Data Studio Data Load within a workflow:

1. On the **Home** page, click the required Project title.
You are navigated to the **Project Details** page.
2. In the left pane, click **Workflows**.
The Workflow page appears.
3. On the **Workflow** page, click **Create Workflow**.
The **Create Workflow** page appears.
4. In the **Name** field provide a name for the new workflow and click **Create**.
The new workflow is created and listed in the **Workflow** page with its **Name** and **Folder** details. Click the Actions icon () next to the workflow to edit, rename, copy, change folder, start, export, or delete it.
5. Click the Workflow to configure the **Workflow Details**.
6. To add a Data Studio Data Load as a step in the workflow, drag the Data Studio Load icon () on the canvas.
7. Double click the Data Studio Load step in the editor to open the step properties page.
8. Select the **Attributes** tab.
9. From the **Connection** drop-down select the connection you want to load the data from.



Note

The drop-down lists only Oracle database connections.

10. Select a Data Studio Load from the drop down.
11. Click , to save the created/designed workflow.
12. Click , to execute the created workflow.
13. To check the status of the workflow, see the Status panel on the right below the Properties Panel. For details about the Status panel, see [Monitor Status of Data Loads, Data Flows, and Workflows](#). This panel shows the link to the Job ID that you can click to monitor the execution status on the Jobs page. For more information about jobs see [Create and Manage Jobs](#).
The newly created workflows get listed in the Project Details page.

Create and Use Variables

A variable is an object that stores a single value, which can be a string, a number or a date.

- [Create a Variable](#)
You can specify the value of the variables when you create the variable or set the value using a query expression.
- [Use Variables in a Data Flow](#)
You can use variables in a data flow by using them as a filter definition. You can refer to variables in a data flow using the format `#<variable_name>`. During execution the variable is substituted by the value.
- [Use Variables in a Workflow](#)
The left panel of the Workflow Details page lists the variables that you can use as steps within a workflow.

Create a Variable

You can specify the value of the variables when you create the variable or set the value using a query expression.

The Variables page lists all the existing variables. Click the **Actions** icon () of a variable to edit, refresh, or delete it. All the available variables are also listed in the left panel of the Workflow Details page. You can use variables as steps in data flows and workflows.

To create a variable:

1. On the **Home** page, click the required Project title. You are navigated to the **Project Details** page.
2. In the left pane, click **Variables**. The Create Variable page appears.
3. Enter a name for the variable.
4. From the **Data Type** drop down, select one of the following:
 - **Short text** - The value can be alphanumeric, can contain special characters, and cannot exceed 255 characters.

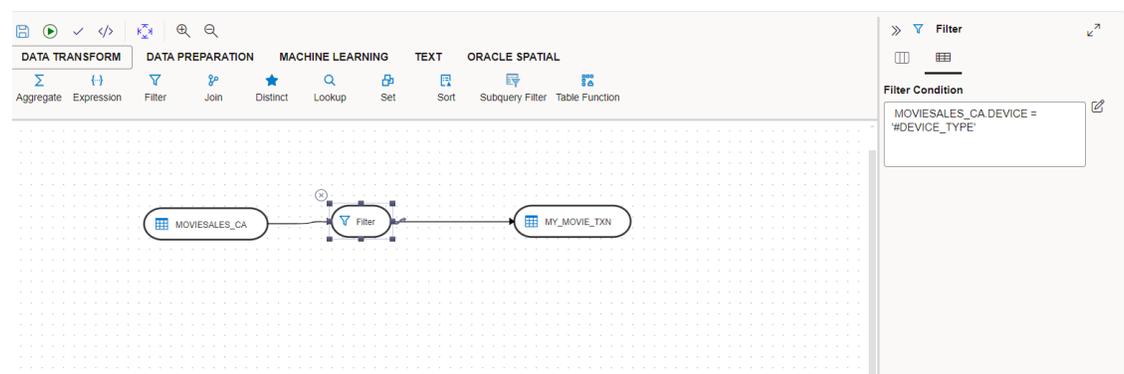
- **Long text** - The value can be alphanumeric, can contain special characters, and cannot exceed 64000 characters.
 - **Numeric** - The value can be a numeric value and can be preceded by a minus (-) sign. The value cannot exceed 10 digits. Possible value range is -999999999 to 999999999.
 - **Date** - The value is a date format.
5. In the **Default Value** field enter the value you want to assign to the variable. The allowed value depends on the **Data Type** you select.
 6. For variables that you modify, you can click the Actions menu next to the variable to open the Refresh Variable page to update the value. From the **Keep History** drop down select how you want the variable value to be displayed in the Refresh Variable page:
 - **All Values** - You can see the history of all the values held by this variable.
 - **Latest Value** - You can see only the latest value specified for the variable.
 - **No History** - Data Transforms does not keep the history of the values held by this variable.
 - **Secure Value** - This is useful when the variable contains passwords or other sensitive data. The value is not displayed in the Refresh Variable page.
 7. Enter a description for the variable.
 8. If you want the variable value to be set by a query, click the **Refresh** tab. Select the connection type and schema where you want to execute the command. Enter the query and click **Validate** to check the syntax of your expression.
 9. Click **Save**.
The newly created variable is listed in the Variables page as well as in the Variables node in the left panel of the Workflow Details page.

Use Variables in a Data Flow

You can use variables in a data flow by using them as a filter definition. You can refer to variables in a data flow using the format `#<variable_name>`. During execution the variable is substituted by the value.

When you execute the data flow, the **Variable values** page appears that displays the list of variables that you have added to the data flow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the data flow. The custom value does not persist for any subsequent sessions.

Here is an example of the use of a variable in a data flow:



In this example, the data flow uses the variable `DEVICE_TYPE` as a step with the filter condition set as follows:

```
MOVIESALES_CA.DEVICE = '#DEVICE_TYPE'
```

The variable is quoted because the substituted value needs to be quoted. Since the variable value is substituted during execution, you can use it for many use cases such as in the transformation expression in the mapping.

Use Variables in a Workflow

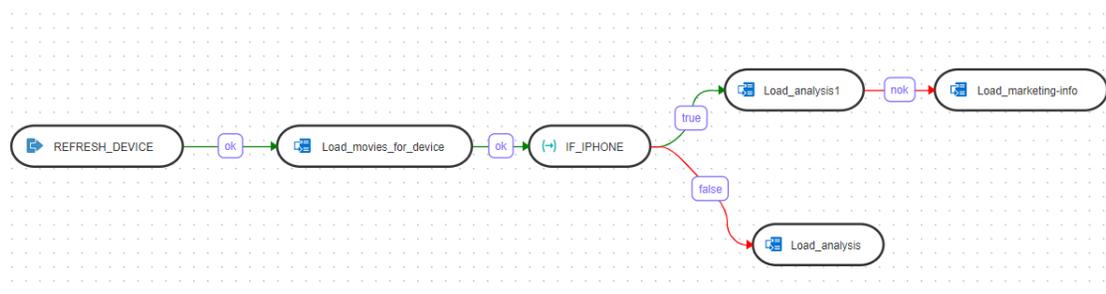
The left panel of the Workflow Details page lists the variables that you can use as steps within a workflow.

You can drag and drop the variables you want to use in the workflow on the design canvas. The Properties Panel available on the right side of the design canvas displays various details about the step such as the following.

- **Name, Type,** and the **Step** sequence number. These are non-editable values.
- Click the link under **Linked Object** to view and update the details about the variable.
- **Operation:** Select any one from the following:
 - **Set Variable** - There are two functions for this step:
 - Update** sets the current value of a variable.
 - Increment** increases or decreases a numeric value by the specified amount.
 - **Refresh Variable** - This variable step refreshes the variable by running the query specified in the variable definition.
 - **Evaluate Variable:** This variable step type compares the value of the variable with a given value according to an operator. If the condition is met, then the evaluation step is true, otherwise it is false.
- **Number of attempts on Failure**
- **Time between attempts in seconds(s)**
- **Log steps in journal** - You can select from Always, Never, or Error.

When you execute the workflow, the **Variable values** page appears that displays the list of variables that you have added to the workflow. You can choose to use the current value, the default value, or set a custom value for each variable. Note that the custom value is applied only to the current run of the workflow. The custom value does not persist for any subsequent sessions.

Here is an example of the use of a variable in a workflow:



In this example, a variable called `DEVICE_TYPE` with the value "iphone" is used as a step in the workflow. The workflow performs the following actions:

1. Execute the "REFRESH_DEVICE" data flow.
2. If the "REFRESH_DEVICE" data flow execution is successful, execute the "Load_movies_for_device" data flow.
3. If the "Load_movies_for_device" data flow execution is successful, execute the "IF_IPHONE" variable evaluation step.
4. If the variable value is equal to 'iphone', execute path for "true".
5. If the variable value is not equal to 'iphone', execute path for "false".

Use Machine Learning (ML) Models

Data Transforms supports the use of ML Model in a data flow. Learn how to create and use Machine Learning (ML) Models in data flows.

- [Create an ML Model Data Entity in the Data Flow editor](#)
To use ML models in Data Transforms you need to create two data flows. You need to first build the ML model data entity using the Data Flow editor, and then you can use the data entity in a data flow to mine data from a source connection and load it into a target server.
- [ML Model Data Entity Properties](#)
The **Properties** tab of the **Add Data Entity** wizard provides data mining options that you can use to define the ML Model data entity.
- [Use ML Model in a Data Flow](#)
You can use the **Prediction Model** database function to run ML Model algorithms on source data and load the output to a target database.

Create an ML Model Data Entity in the Data Flow editor

To use ML models in Data Transforms you need to create two data flows. You need to first build the ML model data entity using the Data Flow editor, and then you can use the data entity in a data flow to mine data from a source connection and load it into a target server.

To build an ML Model data entity in the Data Flow editor,

1. Drag the data entity that you want to build the ML Model on onto the Design Canvas.
2. Select the component and click the Add Data Entity icon  present on the top right corner of the target component.
3. **Add Data Entity** page appears allowing you to configure the following details of the target component:
General tab
 - In the **Name** text box, enter the name of the newly created Data Entity.
 - From the **Entity Type** drop-down, select **ML Model** as the data entity type. When you select this entity type the user interface changes as follows:
 - The **Connection** drop down only lists the Oracle connections that you have created.
 - The Add Data Entity wizard displays the **Properties** tab where you can select the Type of Learning, Function, Algorithm, and configure parameters to define the ML Model. See [ML Model Data Entity Properties](#) for more information.

- From the **Connection Type** drop-down, select the required connection from which you wish to add the newly created Data Entity. For ML Model data entities, the **Connection Type** drop-down only lists **Oracle** as the option.
- The Connection drop-down is populated with the connections you have created with the associated connection type. From the **Connection** drop-down, select the server name where you wish to keep the ML model data entity.
- In the **Schema** drop-down, all schema corresponding to the selected connection are listed in two groups.
 - New Database Schema (ones that you've not imported from before) and
 - Existing Database Schema (ones that you've imported from before and are potentially replacing data entities).

From the **Schema** drop-down, select the required schema.

- In the **Tags** text box, enter a tag of your choice. You can use tags to filter the Data Entities displayed in the Data Entity Page.
- If you want to mark this data entity as a feature group, expand **Advanced Options** and click the **Treat as Feature Group** checkbox.
- Click **Next**.

Properties tab

- Select the **Type of Learning**, **Function**, and **Algorithm** you want to use to build this data entity. For more information about the options, see [ML Model Data Entity Properties](#).
- Based on the options selected, the **Parameters** section is populated with the list of parameters that are marked as "Importance" and "High". You can add other required



parameters using the  icon.

You must specify a value for each parameter so that the data flow can run successfully.

Columns tab



- Click the  Add Columns icon, to add new columns to the newly created Data Entity. A new column is added to the displayed table.
- The table displays the following columns:
 - Name
 - Data Type - Click the cell to configure the required Data Type.
 - Scale
 - Length
 - Actions - Click the cross icon to delete the created column.
- To delete the columns in bulk, select the columns and click the  Delete icon.
- To search for the required column details, in the Search text box enter the required column name and click enter. The details of the required column are displayed.

- Click **Next**.

Preview Data Entity tab

It displays a preview of all the created columns and their configured details. If the data entity belongs to an Oracle database, you can also view statistics of the table. See [View Statistics of Data Entities](#) for more information.

4. Click **Save** to save the configuration and exit the wizard.
5. Save and execute the data flow.
The new Data Entity is created. displayed in the Data Entities page.

ML Model Data Entity Properties

The **Properties** tab of the **Add Data Entity** wizard provides data mining options that you can use to define the ML Model data entity.

This topic assumes prior knowledge of Oracle Machine Learning concepts such as data mining functions and algorithms. For more information, see [Oracle Machine Learning for SQL API Guide](#).

You can use any of these options to configure the properties of the ML Model data entity:

- Type of Learning: Supervised
 - Function: Classification
 - * Decision Tree
 - * Explicit Semantic Analysis
 - * Generalized Linear Models
 - * Naive Bayes
 - * Random Forest
 - * Neural Network
 - * Support Vector Machines
 - Function: Regression
 - * Generalized Linear Models
 - * Neural Network
 - * Support Vector Machines
 - Function: Time Series
 - * Exponential Smoothing
 - Function: Attribute Importance
 - * Minimum Description Length
- Type of Learning: Unsupervised
 - Function: Association
 - * Apriori
 - Function: Attribute Importance
 - * CUR matrix decomposition
 - Function: Anomaly Detection

- * One Class Support Vector Machines
- Function: Clustering
 - * Expectation Maximization
 - * k-Means
 - * Orthogonal Partitioning Clustering
- Function: Feature Extraction
 - * Explicit Semantic Analysis
 - * Non-Negative Matrix Factorization
 - * Singular Value Decomposition

Use ML Model in a Data Flow

You can use the **Prediction Model** database function to run ML Model algorithms on source data and load the output to a target database.

Before you use an ML Model in a data flow, you need build the ML Model. For instructions on how to create an ML model, see [Create an ML Model Data Entity in the Data Flow editor](#).

To use an ML Model in a data flow:

1. Follow the instructions in [Create a Data Flow](#) to create a new data flow.
2. In the Data Flow Editor, drag the tables that you want to use as a source in the data flow and drop them on the design canvas.
3. From the Database Functions toolbar, click **Machine Learning** and drag the **Prediction Model** transformation component drop it on the design canvas.
4. Click the **Prediction Model** transformation component to view its properties.
5. In the **General** tab, specify the following:
 - **Connection** - The drop-down lists all the available Oracle connections. Select the Oracle connection that you want to use.
 - **Schema** - Select the schema.
 - **ML Model** - The drop-down lists all the available ML models. See [Create an ML Model Data Entity in the Data Flow editor](#) for instructions on how to build an ML Model.
6. In the **Column Mapping** tab, map the source column that you want to embed to the INPUT attribute of the operator. The only column available in the column mappings is `prediction parameters`. Drag a text column from the available columns to the Expression column.
7. Drag the table that you want to use as a target in the data flow and drop it on the design canvas.
8. Save and execute the data flow.
Data Transforms will run the prediction model on the source data and write the output to the target table.

Create and Manage Jobs

When you execute a data load, data flow or workflow Oracle Data Transforms creates a job to complete the process in the background.

A job is made up of multiple steps that corresponds to an execution task.

The **Jobs** page by default, lists the execution job sessions that are running and completed for the present day as `Date` parameter is set to `Today` by default. You can click on any of the instance IDs from the list to see its details.

To work with jobs in Data Transforms, you can do the following:

- [Create a Job from the Home page](#)
- [Create a Job from the Projects page](#)
- [View and Delete a Job](#)
- [Configure Purge Interval for Jobs](#)
- [Search for a Job](#)

Create a Job from the Home page

To create a new job,

- On the Home page, click **Jobs** on the left pane to access the Jobs page.
- Click **Create Job**.
The **Create Job** page slides-in.
- From the **Resource Type** drop-down, select the type of resource for which you wish to create a new job. It can be - Data Flow, Workflow or Schema.
- All the resources associated with the selected Resource Type get listed in the **Resource** field. From the **Resource** drop-down, select the required resource for which you wish to create a new job.
- Click **Create**.

A new job is created and is added to the existing list of jobs in the Jobs page.

For each session, you can see the Job Session **ID**, **Name**, **Status**, **Start Time** and **End Time** for each of the jobs that are executed.

Create a Job from the Projects page

To view the job sessions pertaining to a specific project:

- Click the Project tile displayed in the Projects page.
Project Details page appears.
- In the left pane, click **Jobs**.
Jobs page appears. It displays all the jobs and their details pertaining to the selected project.

Note

This page is project specific and displays only the jobs pertaining to the selected project.

- To create a new job, click **Create Job**.
- Enter all the required details and click **Create**.
A new job is created and is added to the existing list of jobs for the project.

View and Delete a Job

Click the Actions menu () to **View Details** of the jobs, **Delete** and **Rerun** the jobs, if required. If any of the steps failed, you can click on them to see the error details.

To delete a job, select the check box of the respective job session and click **Delete**. Upon confirmation the selected job gets deleted.

Configure Purge Interval for Jobs

You can configure the interval to purge older jobs to reduce the load on the sessions table. The default time interval is 30 days, which means that all jobs that are older than 30 days will be purged periodically. Note that only an ADMIN user (odiadmin) can configure the purge interval.

To configure the purge interval for jobs, click the gear icon () on the Jobs page. In the **Configurations** dialog box, set the required value in the **Delete jobs before (days)** field and click **Save**.

Search for a Job

You can also search for the required job session to know its details based on filters such as:

- **Name** - Name of the Job
- **Status** - Select the required session status from the list - All, Done, Error, Queued, Running, Waiting and Warning. Note that to use the Queued and Waiting filter options in the drop-down you must enable the **Show Queued/Waiting** Jobs option.
- **Show Queued/Waiting Jobs** – Enable this option to include the queued or waiting jobs in the list on the Jobs page. When this option is enabled, the **Status** drop-down includes the Queued and Waiting options that you can use to filter the list.
- **Date** - Select the date in which the required job session was executed - All, Today, Yesterday, Last Week, Last Month and Custom Range, which allows you to select specific From and To dates.

Export and Import Objects

You can move Data Transforms objects between environments. You require an Object Storage connection to store the exported file. You then need to use the same Object Storage connection to import the objects.

- [Export Objects](#)
You can export Data Transforms artifacts such as projects, connections, data loads, data flows, workflows, and schedules from one environment and import them to another.
- [Import Objects](#)
To import objects you need to provide the same Object Storage connection that you had used for the export operation.

Export Objects

You can export Data Transforms artifacts such as projects, connections, data loads, data flows, workflows, and schedules from one environment and import them to another.

The Export and Import options in Data Transforms allow you to do the following:

- Move code from development environments to production environments.
- Share code between development environments.
- Backup and restore objects.

To export objects from a Development environment you need to provide an Object Storage connection as an input. You need to use the same Object Storage connection when you want to run the import process in another environment. See [Import Objects](#). The export operation exports all the metadata of the repository objects along with all its dependent parent and child objects, and stores the information in an XML format in the Object Storage bucket.

You can use the Export wizard to export the entire project or export individual objects at the following levels:

- Projects
- Connections
- Data Load
- Data Flow
- Workflow
- Schedule

When you export an individual object, all the dependent objects are automatically included in the export.

Note

Individual variables export is not currently supported. However, when you export a Project, the variables within that project are also exported.

To export objects using the Export wizard:

1. From the left pane, click the Home tab. Click **Export**. The **Export** wizard appears.
2. In the **Export File Name** field, enter a name for the file. The exported data will be stored in the ZIP format with the *filename_timestamp_DTR* file naming convention.
3. Select an Object Storage connection from the drop-down to store the exported file. Note that you need to select the same Object Storage connection during the import operation.
4. Click **Next**.
5. Select the objects that you want to export. You can choose to export a single, multiple, or all the objects that are listed on the page.
6. Click **Next**.
7. Select the connections that you want to export. You may choose not to make any selection on this page. The connections that are associated with the objects that you selected in the previous screen are automatically included during the export operation.
8. Click **Next** to preview your selections.
9. Click **Export** to start the export operation. A confirmation prompt appears when the export operation starts. Click the link that is displayed in the prompt to monitor the progress of the export operation on the Job Details page.

To export individual objects:

1. Click the **Actions** menu of the project, connection, dataload, dataflow, workflow, or schedule that you want to export, and click **Export**.
2. In the **Export File Name** field, enter a name for the file. The exported data will be stored in the ZIP format with the *filename_timestamp_DTR* file naming convention.
3. Select an Object Storage connection from the drop-down to store the exported file. You need to select the same Object Storage connection during the import operation.
4. Click **Export** to start the export operation.
A confirmation prompt appears when the export operation starts. Click the link that is displayed in the prompt to monitor the progress of the export operation on the Job Details page.

Import Objects

To import objects you need to provide the same Object Storage connection that you had used for the export operation.

Note the following about the import process:

- **Projects** - To import projects, you can choose whether you want to overwrite the data in the repository or merge the matching container objects with the objects that are being imported.
 - **Overwrite** - If there is a match in the repository, all the dependent child objects will be overwritten by default.
If there is no match, the project along with the child objects will be copied into the repository.
 - **Merge** - If the project exists in the repository, but the dependent child objects have no match, all the child objects in the repository will be merged with the new child objects that are coming through the import process.
If the project exists in the repository, and the dependent child objects have a match, all the child objects in the repository will be overwritten with the new child objects that are coming through the import process.

If the project does not exist in the repository, the project along with the child objects will be copied into the repository.
- **Connections** - For connections that have a match in the repository, the import process updates only the advanced properties of the connection in the repository along with the connection name. Credentials such as the schema name, passwords, and connection URL are persisted as is in the new environment. You can use such connections without any modification in the target repository.
For connections that have no match in the repository, credentials such as the schema name, passwords, and connection URL are not carried over during the import operation. Post import, you need to update the credential information for such connections. You also need to make sure that you create a schema user in the database along with the required tables created for the schema user, which you may use as a data entity in dataloads, dataflows, workflows, and schedules.
- **Data Loads, Data Flows, and Workflows** - For these objects if there is a match in the repository, all the objects will be overwritten by default.
- **Schedule** - The status for schedules that have no match in the repository are set to Inactive. You need to update the connection information that is associated with the schedule, and then set the status of the schedule to Active.

To import objects:

1. From the left pane, click the Home tab. Click **Import**. The **Import** dialog box appears.
2. From the **Object Storage Connection** drop-down, select the connection that you had used for the export operation. See [Export Objects](#).
3. From the **Import File Name** drop-down, select the exported file.
4. Choose one of the following import options:
 - **Merge** - Choose this to merge the objects that are already in the repository with the objects that are coming through the import process. This option is only available for container objects such as Projects.
 - **Overwrite** - Choose this to replace the objects that are already in the repository with the objects that are coming through the import process.
5. Click **Import** to start the import operation. A confirmation prompt appears when the import operation starts. Click the link that is displayed in the prompt to monitor the progress of the import operation on the Job Details page.

Reference

Get additional information about using Oracle Data Transforms on an Autonomous Database.

Topics:

- [Enable Access to Private Data Sources from Autonomous Database](#)
- [Troubleshoot Mismatch with Server Cert DN Error](#)
- [Increase the Memory of ODI Agent](#)

Enable Access to Private Data Sources from Autonomous Database

APPLIES TO:  Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

The Autonomous Database from where you are accessing Data Transforms should be configured to use a private endpoint to be able to communicate with private database sources. Otherwise when you try to create and test such a connection, you might get a "failed to connect" error.

If your Autonomous Database is configured to use a Private Endpoint, then you can only access private data sources from clients in the same Virtual Cloud Network (VCN). See [Configuring Network Access with Private Endpoints](#) for detailed instructions.

After you have configured network access using private endpoints, you need to do the following additional configurations for Oracle Data Transforms to communicate with private data sources. This chapter includes the following topics:

- [Configuring DNS and Traffic Management](#)
- [Configuring RCE Proxy on the Private Endpoint](#)

Configuring DNS and Traffic Management

You need to set up a fully qualified domain name (FQDN) and specify this FQDN in the connection host field when you create the connection. Refer to <https://docs.oracle.com/en-us/iaas/Content/DNS/Concepts/views.htm> for more information.

The VCN DNS resolver should know about the FQDN and be able to resolve it to a Type A IP address.

If you are not able to test a connection in Data Transforms on an Autonomous Database that is configured to use private endpoints, then do the following:

- Check whether the query to set the `ROUTE_OUTBOUND_CONNECTIONS` database property to `PRIVATE_ENDPOINT` is run. See [Enhanced Security for Outbound Connections with Private Endpoints](#) for more information.
- Make sure that the FQDN is used as the host so that the framework resolves the FQDN to the required IP address.
Ensure that the data source VM has the FQDN set up with the required security rules.

If test connection fails even after using an FQDN, then do the following:

- From the OCI services menu, click **DNS Management** under **Networking**. Configure private DNS zones views and resolvers.
- Create and publish a record using the FQDN and the private IP (Type A) of the target database.

Note

Do not use `*.oraclecloud.com` as the domain name when you set up the FQDN because it is a reserved domain name.

- Retry test connection.
The test connection operation should complete successfully.

Configuring RCE Proxy on the Private Endpoint

For Autonomous Databases that are configured to use a private endpoint, the Autonomous Database service supports RCE to access resources in the customer subnet through a private route.

If you encounter a "failed to connect" error when you try to test a connection, contact your database administrator to check whether the database is RCE-enabled.

Note that RCE proxy can forward the reverse traffic only to the default OCI DNS resolver. If there is any overridden resolver, then connectivity will fail.

Troubleshoot Mismatch with Server Cert DN Error

APPLIES TO:  Data Transforms that is part of the suite of data tools built into Oracle Autonomous Database.

For a connection on an Autonomous Database that is configured to use private endpoints, you must specify the distinguished name (DN) of the database server in the JDBC URL so that the Oracle Connection Manager (CMAN) can accept the request. Otherwise when you try to create and test such a connection, you might get a "Mismatch with Server Cert DN" error.

First, specify `ssl_server_dn_match=yes` in the JDBC URL for the CMAN to accept the request. Next, use the `ssl_server_cert_dn` parameter to specify the DN of the database server. Note that the order in which the keys are placed in `ssl_server_cert_dn` is important. The sequence should be CN, O, L, ST, C.

For example:

```
jdbc:oracle:thin:@(description= (retry_count=20)(retry_delay=3)
(address=(protocol=tcps)(port=1521)(host=xxxxx.adb.us-phoenix-1.oraclecn.com))
(connect_data=(service_name=xxxxx.adb.oraclecloud.com))
(security=(ssl_server_dn_match=yes)(ssl_server_cert_dn="CN=adwc.uscom-
east-1.oraclecloud.com, O=Oracle Corporation, L=Redwood City, ST=California,
C=US")))
```

To get the values for `ssl_server_cert_dn`:

1. Login to the VM that is in the same subnet where your Autonomous Database lies.
2. Type the following command:
`openssl s_client -connect xxxxx.adb.us-phoenix-1.oraclecn.com:1521 -showcerts`

You will find the values for the following under Server Certificate details:

CN, O, L, ST, C

For example, "CN=adwc.uscom-east-1.oraclecloud.com, O=Oracle Corporation, L=Redwood City, ST=California, C=US"

Increase the Memory of ODI Agent

APPLIES TO:  Data Transforms that is available as a separate listing on Marketplace called Data Integrator: Web Edition.

If the data that you are loading from the source schema is huge, then you may want to increase the memory of the ODI Agent to avoid OutOfMemory exception errors.

To increase the memory of the ODI Agent:

1. Edit the `/u01/oracle/transforms_home/common/scripts/jettyServer.sh` file.
2. Add the `java -Xms1024m -Xmx4096m` parameter.
3. Restart the jetty server. Log in as OPC user and execute the following commands:

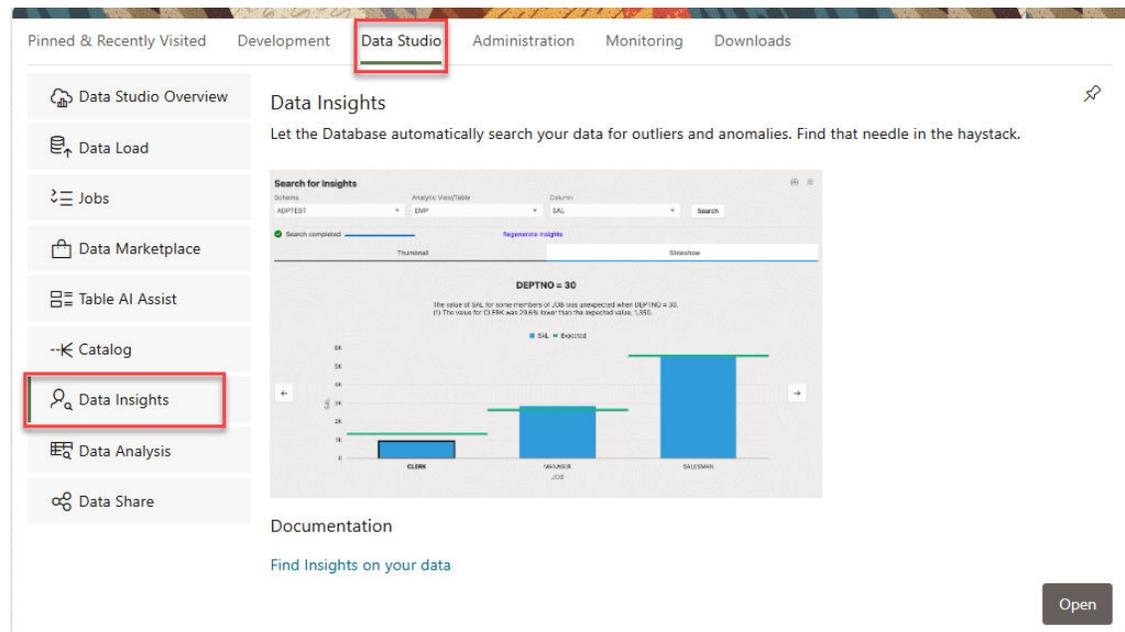
```
ssh -i <path to id_rsa> opc@<Instance IP>
sudo su
systemctl stop|start jettyserver.service
exit
```

15

The Data Insights Page

The Data Insights page displays information about patterns and anomalies in the data of entities in your Oracle Autonomous Database.

To reach the Data Insights page, click the **Data Studio** tab in the Database Actions page, and select the **Data Insights** menu



or click the Selector  icon and select **Data Insights** from the Data Tools menu in the navigation pane.

The following topics describe insights and how to generate and use them.

- [About Insights](#)
You can generate insights for a table or for the analytic view deployed for data analysis.
- [Generate Insights and View Reports](#)
Use these procedures to generate Insights and view reports about them.

About Insights

You can generate insights for a table or for the analytic view deployed for data analysis.

The insights that Data Insights generates for the analytic view of a business model can be more useful than those for a table because of the additional metadata that an analytic view provides.

Insights highlight data points as potentially anomalous if the actual value for a measure when filtering on pairs of analytic view hierarchy values or table column values is considerably higher

or lower than the expected value, calculated across all hierarchy or column values. Insights highlight unexpected patterns, which you may want to investigate.

Insights are automatically generated by various analytic functions built into the database. The results of the insight analysis appear as a series of bar charts in the Data Insights dashboard.

Data Insights uses the following steps to generate insights:

1. Finds the values of a measure, for example Sales, across all of the distinct pairs of the hierarchy or column values for the measure. If Sales has the hierarchies or columns Marital Status, Age Band, Income Level, and Gender, then the pairs would be the values of each distinct value of each hierarchy or column paired to each distinct value of each of the other hierarchies or columns. For example, if the values of Marital Status are Married and Single, and the values of Age Band are A, B, and C, then the pairs would be Married and A, Married and B, Married and C, Single and A, Single and B, and Single and C. Each distinct value of Marital Status would also be paired with each distinct value of Income Level and Gender, and so on.
2. Estimates an expected value for the measure for each hierarchy or column pair.
3. Calculates the actual value for the measure for each hierarchy or column pair, for example Marital Status = S, Age Band = C, and then the difference between the actual value and the expected value.
4. Scores all of the differences and selects the largest variations between the actual and expected values to highlight as potential insights.

The resulting insights highlight cases where the measure value is significantly larger or smaller for a given hierarchy or column value pair than expected, for example much higher Sales where Marital Status = S and Age Band = C.

Insights for analytic views tend to use the higher levels of a hierarchy because the differences between the estimated and actual values are generally larger than they are for lower level attributes. For example, the difference in dollars between the estimated and actual sales for the entire USA are generally larger than the difference between the estimated and actual sales for a town with a population under 1000. The difference is calculated in absolute values, not percentages.

Insights for tables categorize columns as dimension columns or measure columns based on their data types and cardinality. A VARCHAR2 column is always categorized as a dimension, but a NUMBER column may be either a dimension or a measure. For example, a NUMBER column for YEAR values that has only 10 distinct values in a table with 1 million rows is assumed to be a dimension.

Generate Insights and View Reports

Use these procedures to generate Insights and view reports about them.

Generate Insights

To generate insights for a table or business model, do the following:

1. In the **Schema** field, select a schema.
2. In the **Analytic View/Table** field, select an analytic view or a table.
3. In the **Column** field, select a column that contains data about which you want gain insights.
4. Click **Search**.

A confirmation notice announces that the request for insights has been successfully submitted. Dismiss the notice by clicking the Close (X) icon in the notice.

A progress bar indicates that the search is in progress and when it has completed. The insights appear in the Data Insights dashboard as a series of bar charts.

To refresh the display of the insights, click **Refresh**. To have refreshes occur automatically, click **Enable Auto Refresh**.

Click **Recent Searches** to view the list of previous insights search.

Select **View Errors** to see any log of error that occurs while its creation. The results appear in a new browser tab.

View the Report

The charts in the Data Insights dashboard show the data that contain anomalous results. The bars in a chart show the actual values. The expected values are indicated by green horizontal lines. The bars that are outlined in black contain the most significant differences between the expected and the actual values.

For example, if the fact table for the insights records values about an insurance program, and the measures of the fact table are AGE_CODE, GENDER_CODE, INCOME_CODE, NUM_INSURED, NUM_UNINSURED, and YEAR, then insights might be generated for the NUM_INSURED measure. In that case, the dashboard would have a series of charts labeled YEAR and INCOME_CODE. Each chart would have a value of the related dimension in the upper left corner. For example, an INCOME_CODE chart that has a related AGE_CODE might have the AGE_CODE value 2 in the upper left corner.

Clicking a chart displays more details about it. At the top of the expanded view of the chart is the dimension name and value and a short textual analysis of notable insights. Below the analysis is the chart showing the values and insights about them.

For example, a chart for INCOME_CODE might have at AGE_CODE = 2 at the top, plus the textual analysis. In the chart, the INCOME_CODE values would be on the x-axis and the NUM_INSURED values would be on the y-axis. Pointing to a bar on the expanded chart displays the actual and the expected NUM_INSURED value for that INCOME_CODE and AGE_CODE.

Click the **Back** button to return to the Data Insights dashboard.

View Previous Reports

To see the results of a previous search, click the Recent Searches icon at the upper right. In the Recent Searches panel, click anywhere in the box for the insights search that you want to see.

To filter the previous searches, enter a value in the search field at the top of the Recent Searches panel.

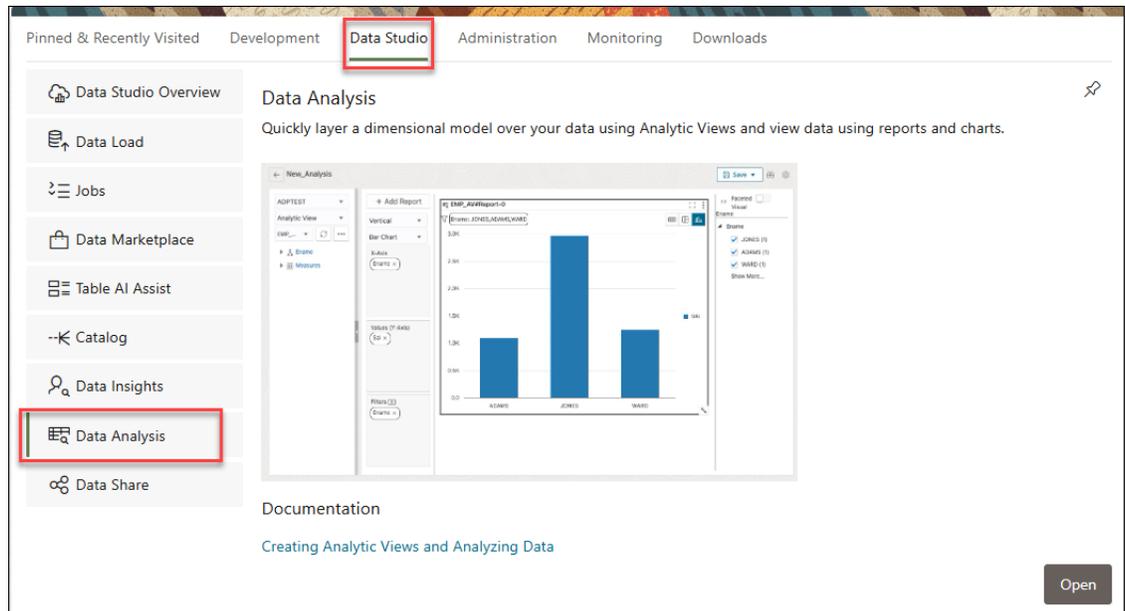
To close the Recent Searches panel without selecting a search, click the X at the upper right of the panel.

16

The Data Analysis Tool

The Data Analysis tool enables you to create Analytic Views with multidimensional metadata.

To reach the **Data Analysis** page, select the **Data Analysis** menu in the Data Studio tab of the Launchpad.



You create Analytic Views on top of a fact table with several dimensions and hierarchies. Analytic views refer to tables in the database and allow users to create hierarchies for dimensions. You can also create Analyses and reports using information from the Analytic Views. The Data Analysis homepage enables you to search for Analyses, view and perform tasks such as edit, delete, view or rename Analyses. You can also analyze, find errors, export, edit, compile and delete Analytic Views. You can analyze tables and generate SQL reports from them.

Select the **Data Analysis** card from the Data Studio suite to access this tool. You can also access it by clicking the Selector icon and selecting Data Analysis from the Data Tools menu in the navigation pane.

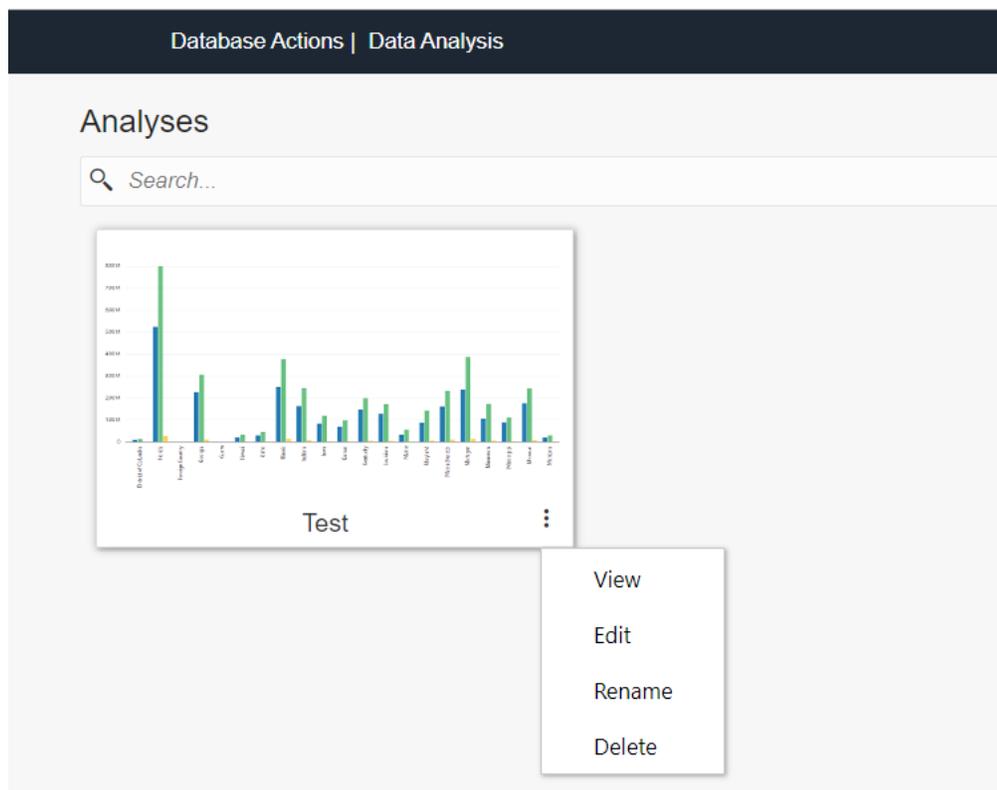
Note

If you do not see the Data Analysis card then your database user is missing the required DWROLE role.

The Data Analysis home page consists of three parts: Analyses, Analytic Views and Tables.



Analyses



The top section of the homepage comprises of a list of Analyses. Use the search field to search for Analyses you create. The top section of the homepage comprises of a list of Analyses. Use the search field to search for Analyses you create.

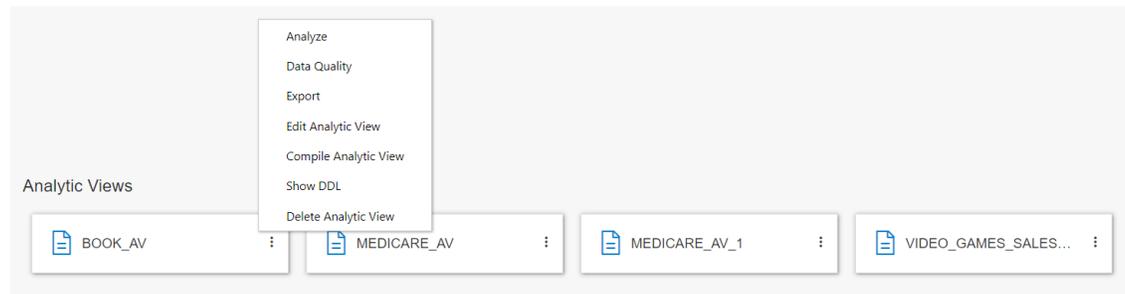
Analyses are analysis of multiple Analytic Views. The Analyses card displays the name of the analysis. Click **Actions** (three vertical dots) to open the context menu.

The actions available are:

- **View:** Opens the Analysis View page in a new window where you can view the analysis.
- **Edit:** Opens the selected Analysis page where you can edit the reports present in the analysis.

- **Rename:** Allows you to rename the Analysis. Click **save** to modify the new name.
- **Delete:** Opens the delete Analysis dialog where you can delete the analysis.

Analytic Views



The bottom section of the homepage displays list of existing Analytic Views. Each Analytic View card displays the name of the Analytic View. The Actions icon enables you to manage the Analytic View. Click **Actions** (three vertical dots) to open the context menu. The actions available are:

- **Analyze:** Opens the Analytic View browser and the Analysis View page in a new window where you can view the Analysis.
- **Data Quality:** Opens the Data Quality page where the tool validates the selected Analytic View for errors and lists them out.
- **Export:** Allows you to export the Analytic View to Tableau and PowerBI.
- **Edit Analytic View:** Opens the Edit Analytic View dialog box where you can edit the properties of the selected Analytic View.
- **Compile Analytic View:** This option compiles the Analytic View and returns compilation errors if there are any.
- **Show DDL:** Displays the DDL statements for the Analytic View.
- **Delete Analytic View:** Deletes the selected Analytic View.

The **+Create** button enables you to create **Analysis** and create **Analytic View** from the home page.

You can select both hierarchies and measures from Analytic Views. Hierarchies are DB objects that allow users to define relationships between various levels or generations of dimension members. As the name implies, hierarchies organize data using hierarchical relationships. With this tool you can analyze and visualize data in different Points of View (POV). You can export the metadata and visualize it with sophisticated tools like Oracle Analytics Cloud (OAC) and Tableau.

Advantages of Data Analysis tool

With Data Analysis tool you can:

- Visualize, analyze and inspect your data clearly and efficiently with pivot tables
- Calculate total number of errors present in the Analytic View you create and provide solutions to minimize the errors
- Automatically display meaningful insights to help you make better decisions
- Analyze your data across dimensions with support for hierarchical aggregation and drill-down
- Share your Analytic Views with the tool of your choice over various options of raw data consumption to draw meaningful insights and make them accessible to any user

By identifying relationships among tables and columns, Analytic Views enable your system to optimize queries. They also open new avenues for analyzing data. These avenues include data insights, improved hierarchy navigation, and the addition of hierarchy-aware calculations.

This tool runs complex and hierarchical SQL queries along with SQL extensions in the background, which simplifies real-time calculations. It makes complex data more accessible and easier to understand.

The Data Analysis Page

The following section describes searching and obtaining information about Analytic Views, creating Analytic Views, inspecting your data, discovering insights and visualizing data using tools like Oracle Analytics Cloud (OAC), Tableau, and Microsoft Power BI.

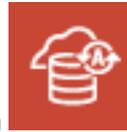
Note

- OAC has in-built tools to search and utilize Analytic Views.
- We have no direct support for Microsoft Power BI, yet its users can map their tool to the AV transparency views to avail some of the benefits of Analytic Views.

Read these topics for detailed descriptions on Analyses and Analytic Views:

- [Searching and obtaining information about Analytic Views](#)
When you first open the Data Analysis page, it displays the list of schemas and Analytic Views. With Select Schema, you can select a preferred Schema from a list of schemas available in the drop-down.
- [Creating Analytic Views](#)
You can create Analytic Views and view information about them. You can also edit and perform other actions on them.
- [Working with Analyses](#)
Analyses are a collection of multiple reports on a single page, which provides quick access to multiple data analyses collected from different Analytic Views.
- [Viewing Analyses](#)
Analyses provide you an insight into the performance of your data.
- [Workflow to build Analyses](#)
Here is the workflow to build an analyses.
- [Creating Analyses](#)
Use the Data Analysis tool to create and edit your Analyses. The analysis provides you customized view of Analytic View data. An analysis consists of one or more reports that displays the results of analysis.
- [Creating Reports](#)
A single report you generate analyzes an AV based on the Levels and measures you select.
- [Run Natural Language Query in the Data Analysis Tool](#)
You can query the Oracle Autonomous AI Database by using Natural Language Query rather than having to write SQL Query.
- [Generate Natural Language explanation from SQL Query using AI Explain](#)
You can translate SQL queries to natural language that is understood by you.

- [Vector Search in the Data Analysis Tool](#)
You can use Oracle AI Vector Search capabilities to search for relevant text from your source data on a specific column.
- [Format Numeric Columns in the Data Analysis Tool](#)
The custom formatting of numeric columns enables you to select your format type such as **Number**, **Currency** and **Percentage** using Mapping numeric column wizard in the Data Analysis tool.
- [Using Calculation Templates](#)
The Data Analysis tool provides templates for all of the calculations typically in demand for business intelligence applications.
- [Access the Spreadsheet Add-in by OCI IAM Domains](#)
Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) uses identity domains to provide identity and access management features such as authentication, Single Sign-On (SSO), and identity lifecycle management for OCI as well as for Oracle and non-Oracle applications, whether SaaS, cloud hosted, or on-premises.
- [Role Based Access Control IAM Connections](#)
This topic provides instructions for IT management on how to prepare a domain-integrated application that enables Role Based Access Control using JSON Web Token (JWT) custom claims. This will enable domain users to log in to the database using their domain credentials instead of the schema password.
- [Oracle Autonomous Database for Excel](#)
The Oracle Autonomous Database for Excel integrates Microsoft Excel spreadsheets with the Autonomous Database to retrieve and analyze data from Analytic Views in the database. You can also directly run SQL queries to view their results in the worksheet.
- [Oracle Autonomous Database for Google Sheets](#)



The Oracle Autonomous Database add-on enables you to query tables using SQL or Analytic Views using a wizard directly from Google Sheets for analysis.

Searching and obtaining information about Analytic Views

When you first open the Data Analysis page, it displays the list of schemas and Analytic Views. With Select Schema, you can select a preferred Schema from a list of schemas available in the drop-down.

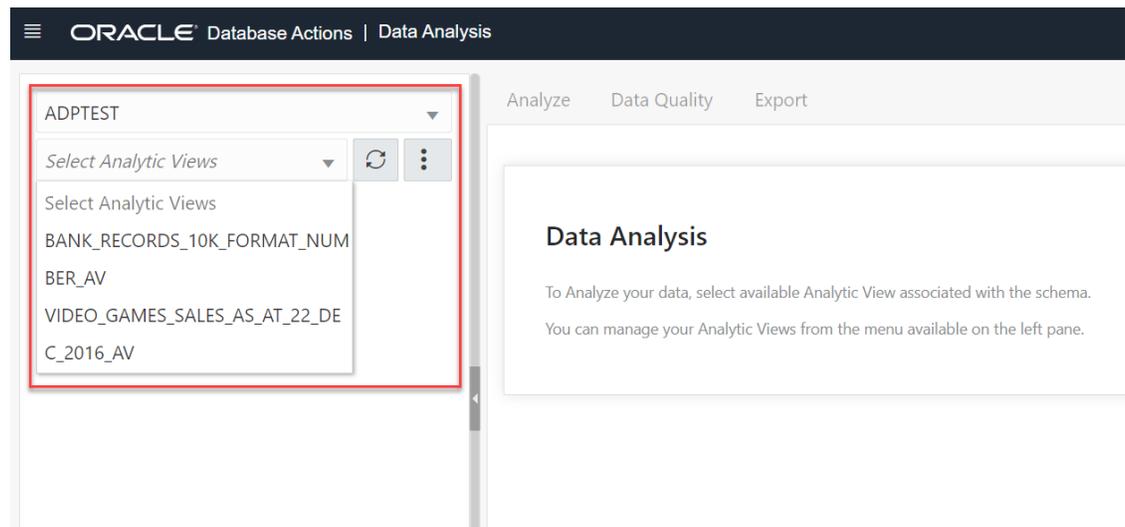
The **Select Analytic Views** drop-down enables you to select an available Analytic View associated with the schema. When you create an Analytic View, it appears in the drop-down option with your schema. The Refresh AV icon refreshes the contents of the selected Analytic View.

The **Action** icon next to the Refresh AV button enables you to manage Analytic Views. You can Create Analytic View, Edit Analytic View, Compile Analytic View, Show the Data Definition Language (DDL) that generates the Analytic View or Delete Analytic View from the menu.

Obtain information about Analytic Views

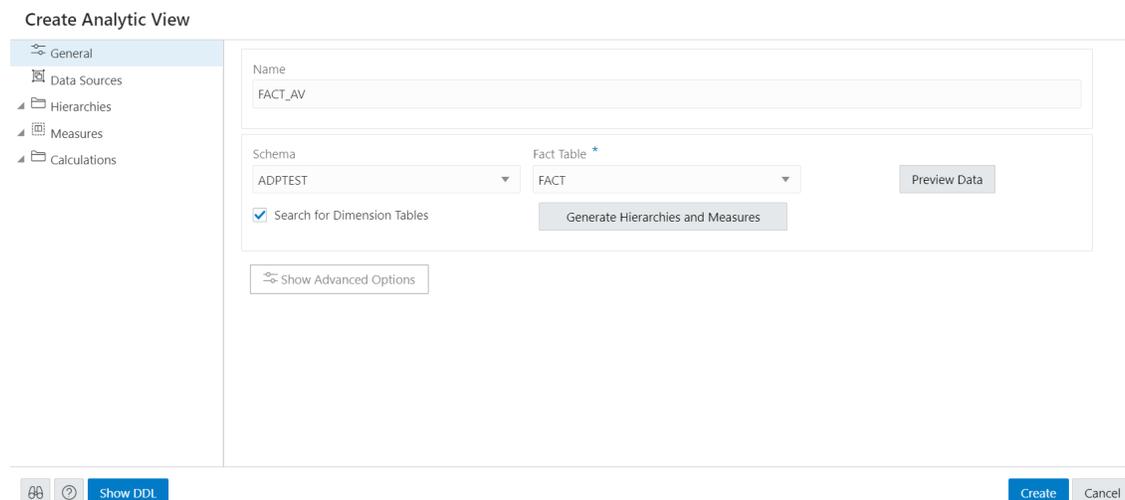
By default, Analytic Views are filtered by the current user's schema, as indicated by the schema list below the menu-bar. You can remove the selected schema filter by selecting another user's schema. To search for Analytic Views in other schemas, select one of the schemas from the drop-down.

If there is no Analytic View associated with the schema selected, the tool prompts you to create an Analytic View.



Creating Analytic Views

You can create Analytic Views and view information about them. You can also edit and perform other actions on them.



When you create an Analytic View, you identify a fact table that contains the data to inspect. The **Generate Hierarchies and Measures** button looks at the contents of that table, identifies any hierarchies in the fact table, and searches for other tables that may contain related hierarchies.

While creating an Analytic View, you can enable or disable the following advanced options:

- **Autonomous Aggregate Cache**, which uses the dimensional metadata of the Analytic View to manage a cache and that improves query response time.

- Analytic View Transparency Views, which presents Analytic Views as regular database views and enables you to use your analytic tools of choice while gaining the benefits of Analytic Views.
- Analytic View Base Table Query Transformation, which enables you to use your existing reports and tools without requiring changes to them.

Create Analytic View

To create Analytic View, click **Create** from the Data Analysis home page and select **Create Analytic View** to begin the process.

Click **Cancel** to cancel the creation of the Analytic View at any time.

Specify Attributes of the Analytic View

On the **General** tab of the **Create Analytic View** pane, specify the following:

- The name for the Analytic View
- The fact table for the view
- Advanced options

You can also preview the data of the fact table and see statistics about that data.

In the **Name** field, specify a name of your choice.

The **Schema** field has the current user's schema. You can only create an Analytic View in that schema.

In the **Fact Table** field, expand the drop-down list and click **More Sources**. The **Select Sources** dialog box has a list of the available tables and views. Select a table or view from the list.

To filter the list, begin typing characters in the Filter field. As you type, the list changes to show the tables or views that contain the characters. Clear the field to show the complete list again. After you select a table or view, click OK.

To enable or disable the advanced options, on the **Create Analytic View** pane, click the **Show Advanced Options** icon at the bottom left. Select or deselect options as desired.

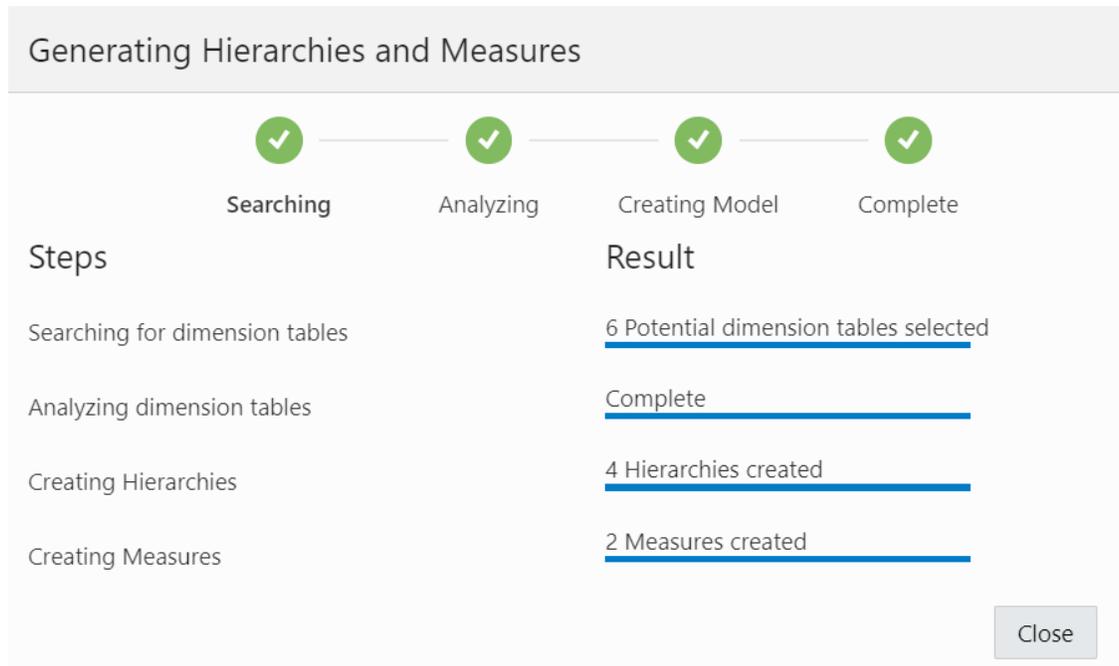
To view the data in the fact table and statistics about the data, click the **Preview Data** button. In the Preview and Statistics pane, the Preview tab shows the columns of the table and the data in the columns.

The Statistics tab shows the size of the table and the number of rows and columns. The statistics may take a few moments to appear, during which time the message, "No statistics available..." may appear. The statistics include the names of the columns, their data types, the number of distinct values and null values, the maximum and minimum values, and other information. The bar graph displays the top unique column values and the number of their occurrences for the selected column. Point to a bar in the graph to see the number of occurrences of the unique value.

Click **Close** to close the Preview and Statistics pane and return to the Create Analytic View pane.

Click on **Generate Hierarchies and Measures** icon.

The Generating Hierarchies and Measures dialog box displays the progress of searching for dimension tables, analyzing the dimension tables and identifying and creating the data sources, joins, hierarchies, and measures to use. When the process completes, click **Close**.



The **Search for Dimension Tables** check box when selected, enables you to search for dimension tables while generating hierarchies and measures.

After the hierarchies and measures are generated, they are displayed under their respective tabs. Review the hierarchies and measures created for you.

Specify the **Name**, **Fact Table** and select **Advanced Options** in the **General** tab of **Create Analytic View** pane. Click **Create** to generate an Analytic View.

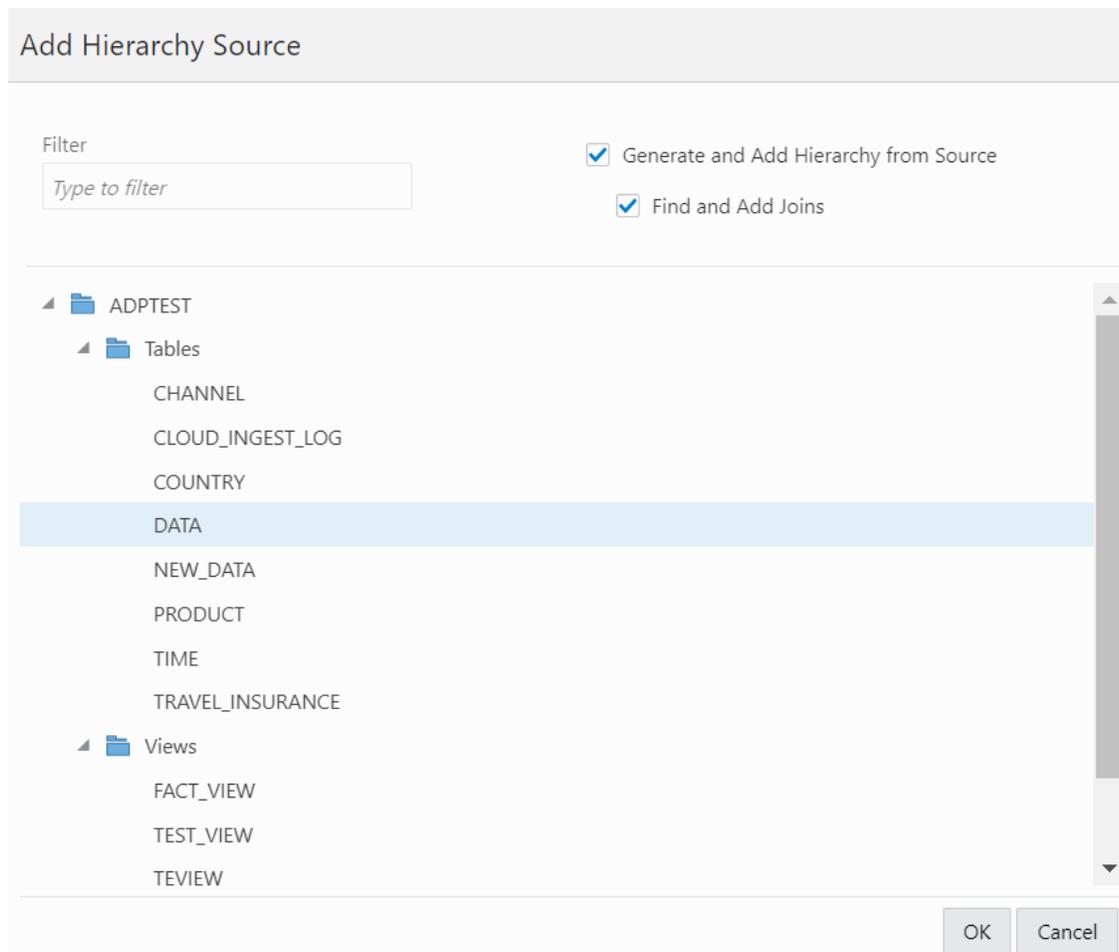
View Data Sources

The Data Sources tab displays the sources of the data and the relationships among them. It has a graphical display of the fact table and the related dimension tables. For example, a fact table of health insurance data might have columns for geography identifiers, income codes, and gender codes. The Data Sources tab would display items for the fact table and for the geography, income, and gender dimension tables.

You can add hierarchies from data sources even after generating hierarchies from the existing fact table. You can add one or more hierarchies to your new or existing analytic view. Multiple hierarchies can be defined and used in an analytical view, however only one will be used by default.

Right-click the Data Sources tab and select **Add Hierarchy Sources** or select **Add Hierarchy Sources**.

Selecting **Add Hierarchy Sources** launches an **Add Hierarchy Source** dialog box.



You can view all the fact tables and views associated with the analytic view.

In the filter field, you can either manually look for the source or start typing to search for the fact table or views from the list of available fact tables and views. After typing the full name of the source, the tool automatically matches the fact table or view.

Select **Generate and Add hierarchy from Source** to generate analysis and hierarchies associated with the source data you select.

Select **Find and Add Joins** to link all the data sources with the fact table. You can add multiple join entries for a single hierarchy.

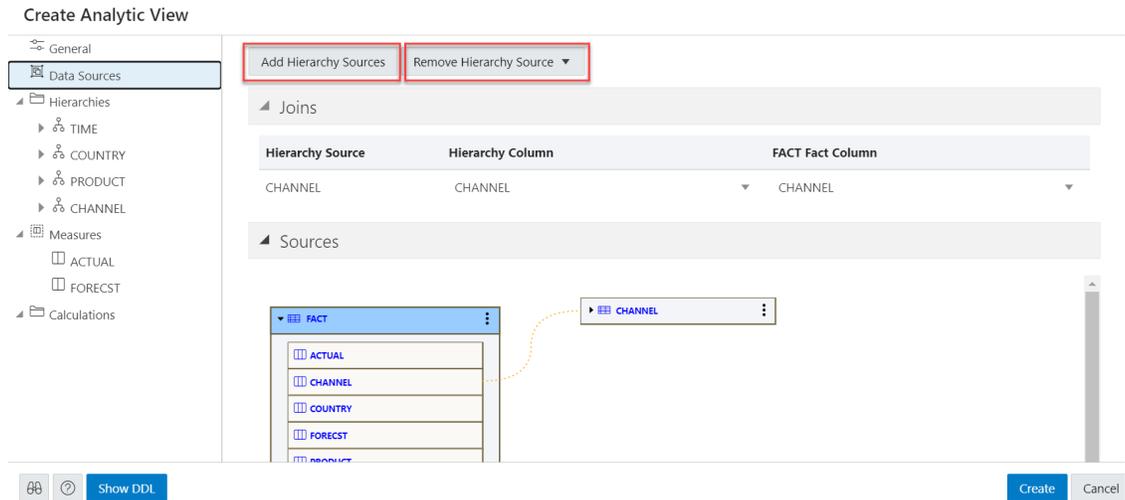
Click **OK** to select the source.

The Generating Hierarchies and Measures dialog box displays the progress of analyzing the dimension tables and creating the hierarchies. When the process completes, click **Close**.

Note

When you add a hierarchy from the data source, you see the new hierarchy in the list of hierarchies in the Hierarchies tab. You can navigate between the Data Sources tab, the Hierarchies tab, the Measures tab, the Calculations tab. You can add a hierarchy from a source that is not connected by navigating back to the Data Sources tab.

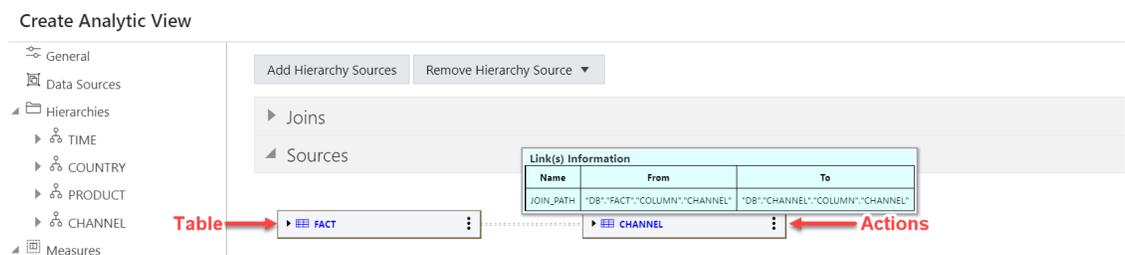
Select **Remove Hierarchy Source** to remove the hierarchies you create from the data sources. You cannot remove hierarchies generated from the fact table you select from this option.



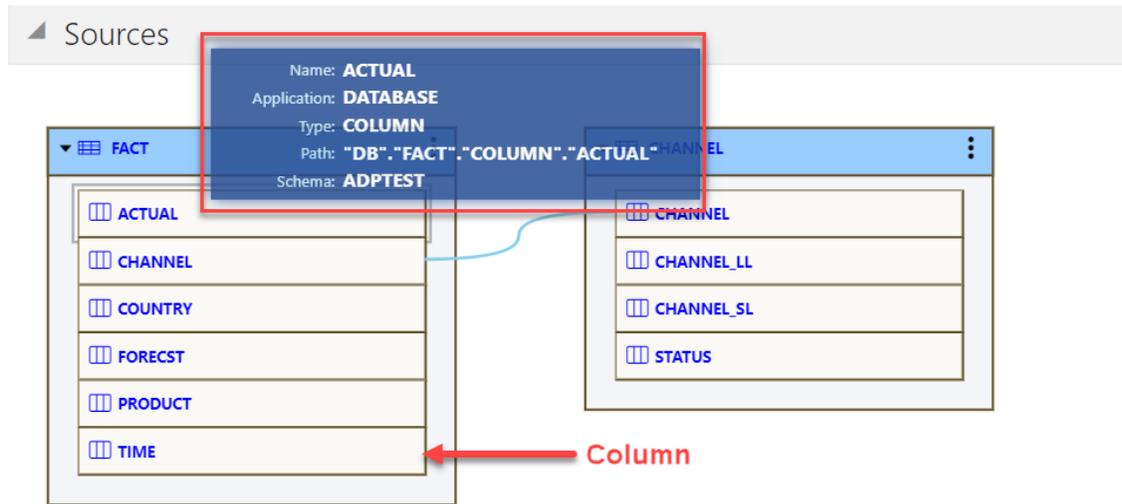
Expand **Joins** to view the **Hierarchy Source**, **Hierarchy Column** and the **Fact column** mapped with the Analytic View. The **Joins** is visible only when the hierarchy table differs from the fact table. You can add multiple join entries for a single hierarchy.

Expand **Sources** to view the fact table associated with the Analytic View. The data model expands to include the data from the source that you added.

Pointing to an item displays the name, application, type, path and the schema of the table. Click the **Actions** (three vertical dots) icon at the right of the item to display a menu to expand or collapse the view of the table.



An expanded item displays the columns of the table. Pointing to a column displays the name, application, type, path, and schema of the column.



The lines that connect the dimension tables to the fact table indicate the join paths between them. Pointing to a line displays information about the join paths of the links between the tables. If the line connects a table that is collapsed, then the line is dotted. If the line connects two expanded tables, then the line is solid and connects the column in the dimension table to the column in the fact table.

View and Manage Hierarchies

The Hierarchies tab displays the hierarchies generated by the Analytic View creation tool. The display includes the name of the hierarchy and the source table.

Create Analytic View

- General
- Data Sources
- Hierarchies**
 - COUNTRY
 - FORECST
- Measures
 - TIME
 - PRODUCT
 - FORECST
 - ACTUAL
- Calculations

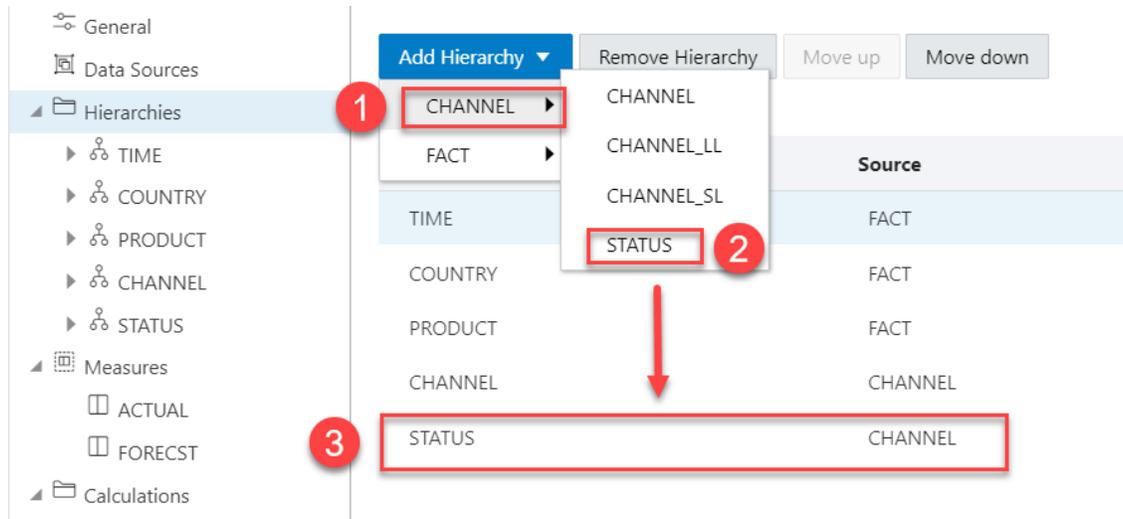
Add Hierarchy
Remove Hierarchy
Move up
Move down
Switch Hierarchy to Measure

Hierarchy Name	Source
COUNTRY	FACT
FORECST	FACT

Show DDL
Create
Cancel

An analytic view must include at least one hierarchy.

To add a Hierarchy, click **Add Hierarchy**. This results in a display as a list of column in that table. Select a column that operates as the detailed level of the hierarchy and be the join-key to the fact table.



To remove the hierarchy, select the hierarchy you want to remove from the list and click **Remove Hierarchy**

Select **Move Up** or **Move Down** to position the order of the Hierarchy in the resulting view.

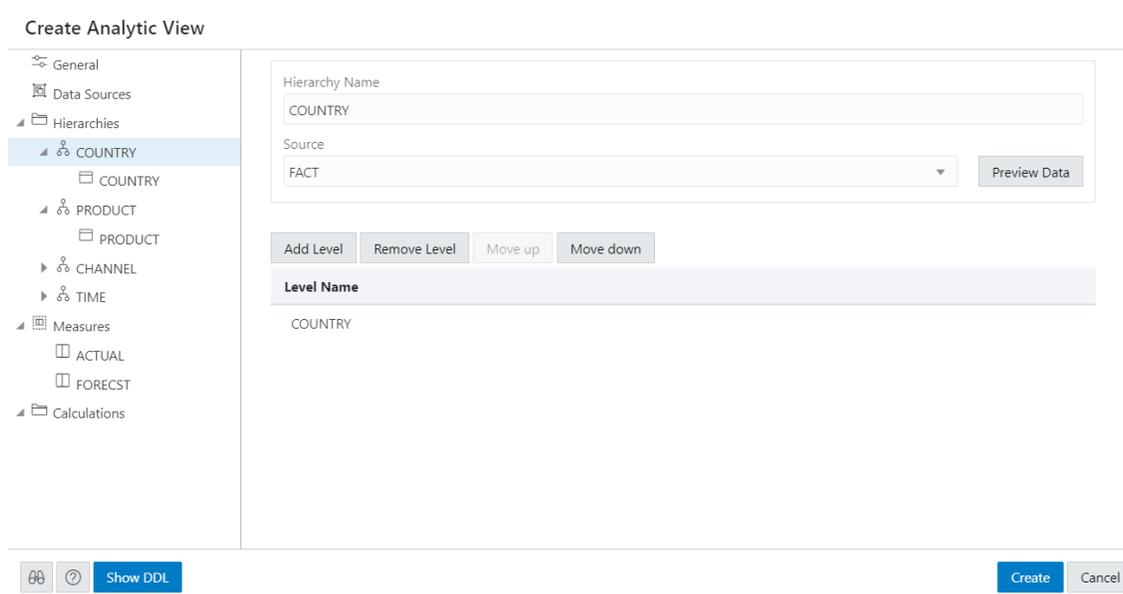
Click **Switch Hierarchy to Measure** to change the hierarchy you select to a measure in the Measures list.

You can also **Add Hierarchy** and **Add Hierarchy From Table** by right-clicking the Hierarchy tab.

If you click on a hierarchy name, a dialog box displays the Hierarchy Name and Source.

To change the source, select a different source from the drop-down list.

Select **Add Level** to add a level to the hierarchy. Click **Remove Level** to remove the selected level from the hierarchy.

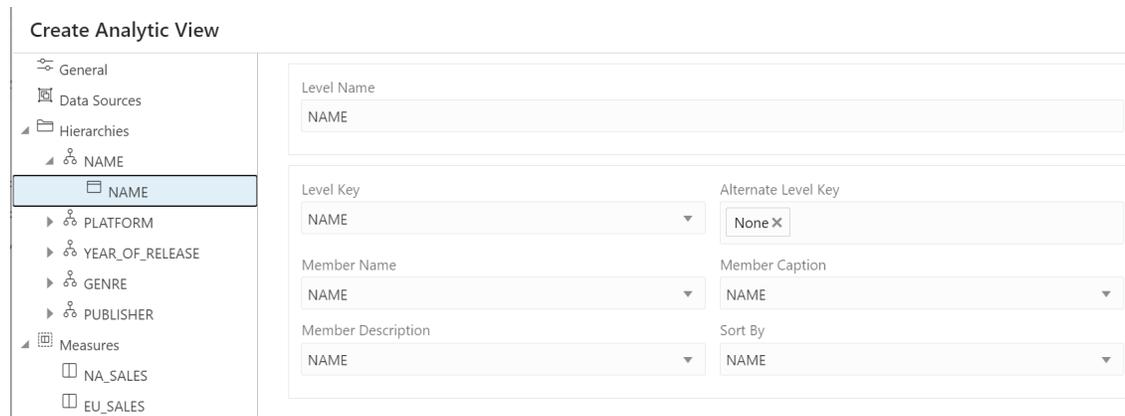


To view the data in the fact table and statistics about the data, click the **Preview Data** button. In the Preview and Statistics pane, the Preview tab shows the columns of the table and the data in the columns. The Statistics tab shows the size of the table and the number of rows and columns.

If you click on a particular level in the Hierarchy tab, a dialog box displays its respective Level Name, Level Key, Alternate Level Key, Member Name, Member Caption, Member Description, source, and Sort By drop-down. To change any of the field values, enter the value in the appropriate field.

Note

You can enter multiple level keys Member Name, Member Caption, Member Description and Sort By.



Create Analytic View

General

Data Sources

Hierarchies

- NAME
- PLATFORM
- YEAR_OF_RELEASE
- GENRE
- PUBLISHER

Measures

- NA_SALES
- EU_SALES

Level Name
NAME

Level Key
NAME

Alternate Level Key
None X

Member Name
NAME

Member Caption
NAME

Member Description
NAME

Sort By
NAME

Member Captions and Member Descriptions generally represent detailed labels for objects. These are typically end-user-friendly names. For example, you can caption a hierarchy representing geography areas named GEOGRAPHY_HIERARCHY as "Geography" and specify its description as "Geographic areas such as cities, states, and countries."

To see the measures for the Analytic View, click **Measures** tab. To immediately create the Analytic View, click **Create**. To cancel the creation, click **Cancel**.

View and Manage Measures

The Measures tab displays the measures suggested for the Analytic View. It displays the Measure Name, Column, and operator Expression for each measure.

The measures specify fact data and the calculations or other operations to perform on the data.

To add measures, click **Add Measure**. You can view a new measure at the bottom of the measures list. To remove the measure, select the measure you want to remove from the list and click **Remove Measure**.

Create Analytic View

General

Data Sources

Hierarchies

- COUNTRY
 - COUNTRY
 - FORECST

Measures

- TIME
- PRODUCT
- FORECST
- ACTUAL

Calculations

Default measure
TIME

Add Measure Remove Measure Switch Measure to Hierarchy

Measure Name	Column	Expression
TIME	TIME	SUM
PRODUCT	PRODUCT	SUM
FORECST	FORECST	SUM
ACTUAL	ACTUAL	SUM

Show DDL Create Cancel

To alternatively add a measure from the data source, right-click the Measures tab. This pops up a list of columns that can be used as measures. Select one measure from the list.

Create Analytic View

General

Data Sources

Hierarchies

- TIME
- COUNTRY
- PRODUCT
- CHANNEL

Measures

- ACTUAL
- FORECST

Calculations

Default measure
ACTUAL

Add Measure ▼ Re

Measure Name

- ACTUAL
- CHANNEL
- COUNTRY
- FORECST
- PRODUCT
- TIME

You can exclude a column from the measures on right-clicking the Measures tab and selecting Remove Measure.

Click **Switch Measure to Hierarchy** to change the measure you select to hierarchy in the Hierarchies list.

You must specify a measure as the default measure for the analytic view; otherwise, the first measure in the definition is the default. Select **Default Measure** from the drop-down.

To add a measure, right-click the Measures tab and select **Add Measure**. To remove a measure, select the particular measure you want to remove, right-click on it and select **Remove Measure**.

You can select a different column for a measure from the Column drop-down list. You can select a different operator from the Expression drop-down list.

In creating an analytic view, you must specify one or more hierarchies and a fact table that has at least one measure column and a column to join to each of the dimension tables outside of the fact table.

Create new calculated measures

You can add measure calculations to a query of an analytic view.

The measures and hierarchies associated with the analytic views enable us to create new calculated measures.

Calculated measures return values from data stored in one or more measures. You compute these measures at run time.

Note

You can create the measures without increasing the size of the database since the calculated measures do not store the data. However, they may slow performance. You need to decide which measures to calculate on demand.

The Analytic Views provides easy-to-use templates for creating calculated measures.

Once you create a calculated measure, it appears in the list of measures of the Analytic View. You can create a calculated measure at any time which is available for querying in SQL.

The Data Analysis tool provides easy-to-use templates for creating calculated measures.

The screenshot shows the 'Create Analytic View' dialog box. On the left, a tree view contains 'General', 'Data Sources', 'Hierarchies' (with sub-items TIME, COUNTRY, PRODUCT, CHANNEL), 'Measures', and 'Calculations' (which is selected). The main area features two buttons: 'Add Calculated Measure' and 'Remove Calculated Measure'. Below them is a text input field for 'Calculation Name'. A message states: 'No calculations defined. Please invoke the context menu on the Calculations folder and select the 'Add Calculated Measure' menu item to add a calculation'. At the bottom, there are three buttons: 'Show DDL', 'Create', and 'Cancel'.

Click **Add Calculated Measure** to add calculations to the measures. You can view the new calculation with system generated name in the **Calculations** tab.

Click the newly created calculated measure.

The screenshot shows the configuration interface for a calculated measure. On the left is a navigation tree with categories: General, Data Sources, Hierarchies, Measures, and Calculations. The 'CALCULATION1' item is selected under Calculations. The main configuration area includes:

- Measure Name:** CALCULATION1
- Calculation Category:** Prior and Future Period
- Calculation Template:** Future Period
- Measure:** NET_SALES
- Hierarchy:** DISTRIBUTION_CHANNEL
- Offset:** 1
- Expression:** LEAD(NET_SALES) OVER (HIERARCHY DISTRIBUTION_CHANNEL OFFSET 1)

In the **Measure Name** field, enter the name of the calculated measure.

You can select preferred category of calculation from a list of options such as Prior and Future Period, Cumulative Aggregates, Period To Date, Parallel Period, Moving Aggregates, Share, Qualified Data Reference, and Ranking using the **Calculation Category** drop-down.

Your choice of category of calculation dynamically changes the **Calculation Template**.

For more details on how to use Calculation templates, see [Using Calculation Templates](#).

Select the **Measure** and **Hierarchy** on which you want to base the calculated measures.

Select **Offset** value by clicking the up or the down arrow. The number specifies the number of members to move either forward or backward from the current member. The ordering of members within a level is dependent on the definition of the attribute dimension used by the hierarchy. The default value is 0 which represents POSITION FROM BEGINNING.

The Expression field lists the expressions which the calculated measure uses.

On the creation of the Analytic view, the calculated measure appears in the navigation tree in the Calculated Measures folder.

Click **Create**. A confirmation dialog box appears that asks for your confirmation. Select **Yes** to proceed with the creation of Analytic View.

After creating the Analytic View, you will view a success message informing you of its creation.

On editing the Analytic View you create, you can view the calculated measure in the navigation tree in the Calculations folder.

Click the **Tour** icon for a guided tour of the worksheet highlighting salient features and providing information if you are new to the interface.

Click the **help** icon to open the contextual or online help for the page you are viewing.

Click **Show DDL** to generate Data Definition Language statements for the analytic view.

Edit Analytic View

You might want to edit an Analytic View to make changes to the data sources, the hierarchies, or the measures.

To edit an Analytic View, click the **Action** icon on the Analytic View item, then click **Edit Analytic View**. On the Edit Analytic View screen, select a tab and make changes as desired.

When you have completed the changes, click **Update**.

Working with Analyses

Analyses are a collection of multiple reports on a single page, which provides quick access to multiple data analyses collected from different Analytic Views.

Analyses enable you to monitor performance, create reports and set estimates and targets for future work. It provides you a visual representation of performance with charts and graphs.

You can access the Analyses page by clicking the Analyses tile on the Data Analysis home page.

Viewing Analyses

Analyses provide you an insight into the performance of your data.

You can use the Analysis and the Analyze pane to search or browse Analytic Views, view their analysis, or reports you have access to. Clicking on the Analyses takes you to a page where you can view the Analyze pane. Here you can view default hierarchy level and measures selected. You can drag and drop any levels and measures from the Analytic View browser to rows/columns and Values in the drop area respectively. This defines your analysis criteria. Once the values are dropped, the Data Analysis tool generates a query internally. The tool displays the results of the analysis in the form of reports in the Analyses that matches your analysis criteria. You can add multiple reports to the Analysis. You can also examine and analyze the reports and save them as a new analysis. You can just save the Analysis and not a single report. Once you save all the reports, it will be part of that single Analysis. Reports are unnamed.

Workflow to build Analyses

Here is the workflow to build an analyses.

Following are few common tasks to start building Analyses:

- **Create a useful analysis:** Before creating your first analysis, you can construct a useful analysis over a single Analytic View. This way you can generate analyses on which you can create reports that you display on the Analyses.
- **Create Analysis:** Create an analysis to display data from analysis.
- **Create Reports:** An Analysis can have multiple reports that are independent of one another. This can be used to compare and analyze data generated from different Analytic Views.
- **Save Analysis:** Create customized Analyses that enable you to view reports and their analyses in the current state and save it for future reference.

Format Numeric Columns in the Data Analysis Tool

The custom formatting of numeric columns enables you to select your format type such as **Number**, **Currency** and **Percentage** using Mapping numeric column wizard in the Data Analysis tool.

Note

Mapping wizard can be viewed only in tabular mode and Chart mode of viewing output.

To map the numeric column:

1. From the Table browser column in the left, drag and drop the numeric column to the measure drop area.
2. Click the respective column on the drop area. The Formatting dialog box appears that enables you to perform changes to the numeric column.
3. The options available depends on the Format Type of the respective column you select. The various format types are **Number**, **Currency** and **Percentage**.
4. The following options are available from the Formatting dialog box when you select **Number** as format type:
 - **Precision:**
It represents the total number of digits. You can select **Default**, **0.0**, **0.00** or **0.000**.
For example: To view the number 1.2, map the precision to **0.0**.
 - **Thousand Split:** With this formatting you can insert a comma after every third significant figure.. For example, select **1,234** from the drop-down to view the number as 3,454 instead of 3454.
 - Select **Abbreviate** to view the abbreviated form of the number. For example, you will view 1000 as 1k.
 - **Aggregation:** This column performs aggregation operations such as **Sum**, **Min**, **Max**, **Avg**, **Count** and **Distinct Count** on the selected input column.
5. If your column is of **Currency** Type, you can view the following options:
 - **Select Currency:** You can select various currencies from the drop-down list.
Select **Highlight Negative Numbers** to highlight negative Numbers.
You can also view **Precision**, **Thousand Split**, **Abbreviate** and **Aggregation** options.
6. If your column is of **Percentage** Type, you can view **Precision**, **Thousand Split**, **Abbreviate** and **Aggregation** options.

Creating Analyses

Use the Data Analysis tool to create and edit your Analyses. The analysis provides you customized view of Analytic View data. An analysis consists of one or more reports that displays the results of analysis.

You can create a Basepay analysis and add content to track your team's pay. You can view the analysis in a pivot view or tabular view or in the form of charts. You can create an analysis that displays these three views.

In this example, you create a new analysis called New_Analysis.

1. On the Data Analysis home page, click the **Create Analysis** button.
2. You can make changes to existing Analyses by adding different values from hierarchies and measures.

3. The AV name you view on the report represents the AV used to create the report. With a different Analytic View you can create a different report.

Note

You must have at least one report to build an analysis.

4. To edit the existing analysis, In the Analytic View browser, select the objects to analyze in the navigation pane and drag and drop them to the drop area in Columns, Rows or Values and Filters of the Analyze tab.

5. The report updates based on the artifacts (levels and measures) you select.

The new analysis which contains the updated report will now be visible in the Data Analysis home page for further editing.

- [Saving Analyses](#)
You can save the personalized settings you made for the Analysis and use them on any other Analyses for future reference.

Saving Analyses

You can save the personalized settings you made for the Analysis and use them on any other Analyses for future reference.

You won't have to make these decisions manually every time you open the Analyses page if you save these preferences.

1. Open the analysis for editing from the Data Analysis home page. Select **Edit** from the Analysis tile you wish to edit.
2. Click on the **Save As** icon. Enter a descriptive name for your Analysis.
3. Click **Save**.

Creating Reports

A single report you generate analyzes an AV based on the Levels and measures you select.

You can add multiple reports to the newly created analysis. One report is independent of another report. To add a report in the Analysis.

1. Open the Analysis for editing from the Data Analysis home page. Select **Edit** from the Analysis tile you wish to edit.
2. Click on the **+ Report** icon to add one or more reports to the Analysis. You can use a report to add configured Analyses to the Analyses page.
3. Click on the report to select it. The resize arrow in the report resizes the report window.
4. Click the cross icon on the selected report to delete the report from the Analysis.
5. The header displays the name of the Analytic View you select.
6. You can expand or collapse the report with their respective arrows.

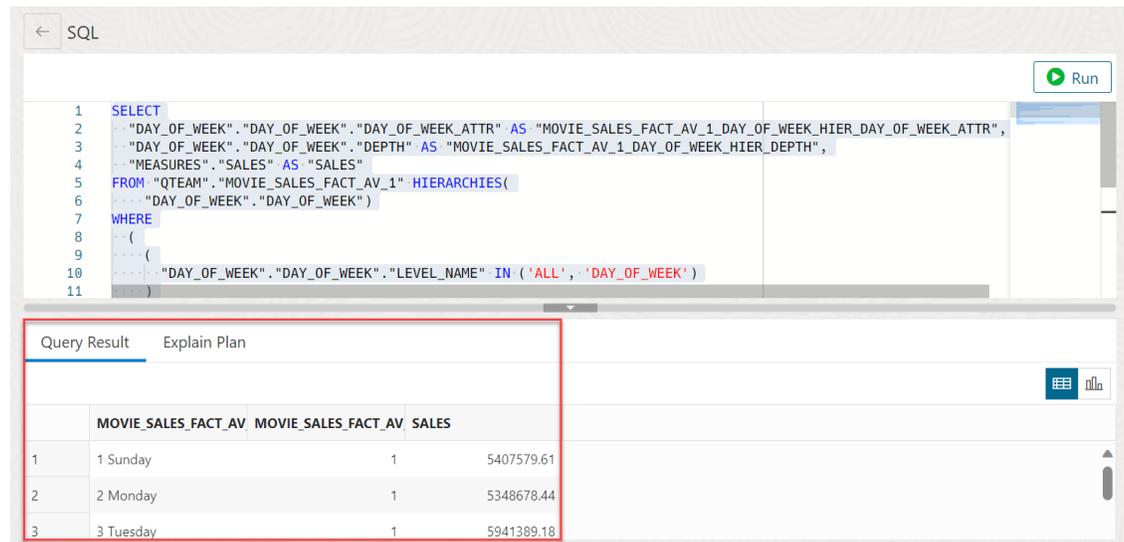
When you add a report to the Analysis, the report provides the following actions:

- Edit SQL
- Performance

- **Rename Report:** Click Rename Report to rename it. Click **Save Report** to save the current report.
- **Delete Report:** Click **Delete Report** to delete the report.

Edit SQL

You can view the SQL output when you click **Edit SQL**. The lower right pane in SQL displays the output of the operation executed in the SQL editor. The following figure shows the output pane on the SQL page.



The screenshot shows the Oracle SQL Developer interface. The top pane displays a SQL query:

```

1 SELECT
2   "DAY_OF_WEEK"."DAY_OF_WEEK"."DAY_OF_WEEK_ATTR" AS "MOVIE_SALES_FACT_AV_1_DAY_OF_WEEK_HIER_DAY_OF_WEEK_ATTR",
3   "DAY_OF_WEEK"."DAY_OF_WEEK"."DEPTH" AS "MOVIE_SALES_FACT_AV_1_DAY_OF_WEEK_HIER_DEPTH",
4   "MEASURES"."SALES" AS "SALES"
5 FROM "QTEAM"."MOVIE_SALES_FACT_AV_1" HIERARCHIES(
6   "DAY_OF_WEEK"."DAY_OF_WEEK")
7 WHERE
8   (
9     (
10    "DAY_OF_WEEK"."DAY_OF_WEEK"."LEVEL_NAME" IN ('ALL', 'DAY_OF_WEEK')
11    )
12  )

```

The bottom pane shows the "Query Result" tab with the following data:

	MOVIE_SALES_FACT_AV	MOVIE_SALES_FACT_AV	SALES	
1	1	Sunday	1	5407579.61
2	2	Monday	1	5348678.44
3	3	Tuesday	1	5941389.18

The output pane has the following tabs:

- **Query Result:** Displays the results of the most recent Run Statement operation in a display table.
- **Explain Plan:** Displays the plan for your query using the Explain Plan command. The default view is the diagram view. For more information, see the description of Explain Plan Diagram in subsequent sections.

Performance menu

The Performance menu displays the PL/SQL procedures in the worksheet area which describes the reports associated with the Analytic Views.

The top part of the performance output consists of the worksheet editor for running SQL statements and an output pane to view the results in different forms. You can view the results in a Diagram View, Chart View, Clear Output from the SQL editor, Show info about the SQL statements and Open the performance menu in a new tab.

The following figure shows the output pane of the performance menu:

The screenshot shows the Oracle SQL Developer interface. At the top, there's a window titled "MOVIE_SALES_FACT_AV#Report-0". Below the title bar is a "Run" button. The main area is a SQL editor with the following code:

```

1 SELECT
2   "DAY_OF_WEEK"."DAY_OF_WEEK", "DAY_OF_WEEK_ATTR" AS "MOVIE_SALES_FACT_AV_DAY_OF_WEEK_HIER_DA
3   "DAY_OF_WEEK"."DAY_OF_WEEK", "DEPTH" AS "MOVIE_SALES_FACT_AV_DAY_OF_WEEK_HIER_DEPTH",
4   "MEASURES"."SALES" AS "SALES"
5 FROM "QTEAM"."MOVIE_SALES_FACT_AV" "HIERARCHIES(
6   "DAY_OF_WEEK"."DAY_OF_WEEK")
7 WHERE
8   (
9   (

```

Below the editor is a toolbar with five icons: a diagram view icon, a list view icon, a trash icon, an info icon, and a new tab icon. This toolbar is highlighted with a red box. Below the toolbar is a table showing the execution plan:

OPERATION	OBJECT NAME	OBJECT TYPE	CARDINALITY
SELECT STATEMENT	null	(null)	2048
RESULT CACHE	74gfsgh0g9a17a8tt9kdjhzs	(null)	2048
VIEW	null	(null)	2048

The output pane has the following tabs:

- **Diagram View:** Displays the plan of your query in the diagram view.
- **Chart View:** Displays the plan of the query in a chart view.
- **Clear Output:** Clears the PL/SQL statements from the worksheet.
- **Show info:** Displays the SQL statement for which the output is displayed.
- **Open in new tab:** Opens the explain plan in a new window. An Explain Plan displays the plan for your query.

Explain Plan Diagram

The Explain Plain diagram view is a graphical representation of the contents of the insert row statements in the SQL Query. The plan depicts the hierarchical nature of the steps in the execution plan.

By default, three levels of steps are visible in the diagram. You can use the **+/-** signs at the

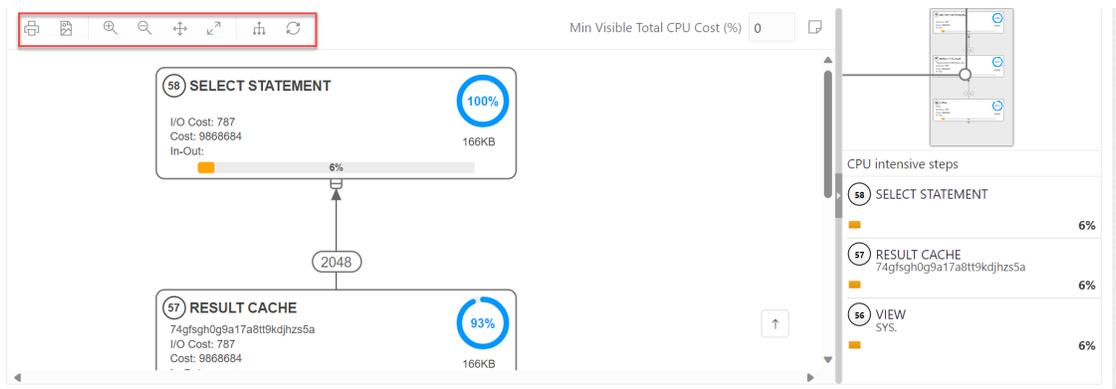
bottom of each step (available when the step has children) to expand or collapse. Use Expand All in the toolbar to view all steps in the diagram.

The diagram also provides the following details:

- Cardinality (number on the arrow to the parent step), which is the number of rows processed
- Operation and options applied in that step
- Execution order, which is the sequential number in the order of execution
- Access predicates CPU cost in percentage (orange bar)
- Total CPU cost for the step in percentage (blue circle)
- Estimated I/O Cost, Bytes processed, and Cost metrics

You can see a brief description pop-up when you hover over any of these statistics in a step.

The icons in the toolbar are:



- **Advanced View:** This is the default view of the query when you click Performance. Displays data from SQL Query in mixed tabular/tree view. There is a Diagram View icon that you can use to switch back to the diagram view.
- **Chart View:** Displays data from the SQL query in the form of charts.
- **Print Diagram:** Prints the diagram.
- **Save to SVG:** Saves the diagram to file
- **Zoom In, Zoom Out:** If a step is selected in the diagram, clicking the Zoom In icon ensures that it remains at the center of the screen.
- **Fit Screen:** Fits the entire diagram in the visible area.
- **Actual Size:** Sets the zoom factor to 1.
- **Expand All:** Displays all steps in the diagram.
- **Reset Diagram:** Resets the diagram to the initial status, that is, only three levels of steps are displayed.
- **Show Info:** Shows the SELECT statement used by the Explain Plan functionality.
- **Open in New Tab:** Opens the diagram view in a new tab for better viewing and navigation. The diagram is limited to the initial SELECT statement.
- **Min Visible Total CPU Cost(%):** Defines the threshold to filter steps with total CPU cost less than the provided value. Enter a value between 0 and 100. There is no filtering for 0.
- **Plan Notes:** Displays the Explain Plan notes.

Properties in Explain Plan Diagram

Double-click or press **Enter** on a selected step to open the Properties slider, which provides more information about that step. See `PLAN_TABLE` in [Oracle Database Reference](#) for a description of each property. The Properties slider shows:

- Displays information for that step extracted from `PLAN_TABLE` in a tabular format. Nulls are excluded. You can select **JSON** to view the properties in JSON format.
- Information from `OTHER_XML` column of `PLAN_TABLE`. The information is displayed in JSON format.
- [Working with Reports](#)
Reports help you in analyzing Analytic Views and Queries.
- [Creating Reports on a Query](#)
This section describes the steps to create reports on an SQL query.

- [Creating Reports on an Analytic View](#)
This section describes the steps to create reports on an Analytic View:

Working with Reports

Reports help you in analyzing Analytic Views and Queries.

The reports are based on the levels and measures you select for the Analytic View and the columns you select for a query.

Click **Analyze** in the Analytic View and click the Table you want to analyze to view the Analyses page.

The Analyses page comprises the following components:

1. **Analytic View Browser:** Select **Analytic View** from the drop-down if you choose to create reports on Analytic Views to view an Analytic View browser. The Analytic View browser displays the Hierarchies, Levels, and Measures associated with the selected Analytic View.

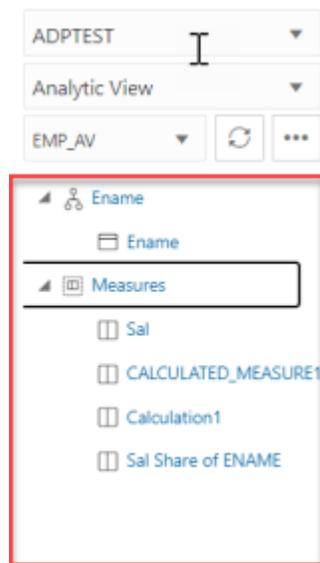
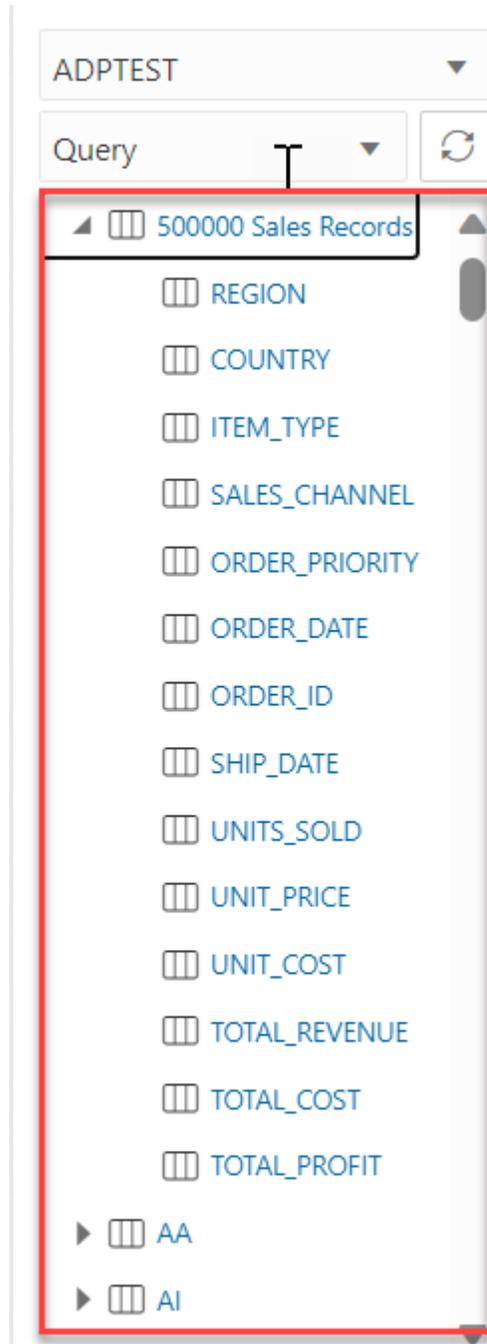


Table Browser: Select **Query** from the drop-down if you choose to create a report on SQL query to view a Table browser. If you select SQL Query, the Table browser displays the available tables and their corresponding columns. You can drill down tables to view their corresponding columns.



- SQL Worksheet editor with the Run icon:** You can view this component only when you generate a report on a SQL Query and not an Analytic View. The SQL editor area enables you to run SQL statements and PL/SQL scripts from the tables you want to query displayed on the Table browser. By default, the SQL editor displays the Select * statement to display all the columns from the first table. Click **Run** to run the statements in the editor.



3. **Output pane:** The output pane, when you view the results of a SQL Query, consists of the following tabs:
- **Query Result:** Displays the results of the most recent Run Statement operation in a display table.
 - **Explain Plan:** Displays the plan for your query using the Explain Plan command. For more information, refer to the **Explain Plan Diagram** in [Creating Reports](#) section.
 - **Autotrace:** Displays the session statistics and execution plan from `v$sql_plan` when running an SQL statement using the Autotrace feature.

REGION	COUNTRY	ITEM_TYPE	SALES_CHANNEL	ORDER_PRIORITY	ORDER_DATE	ORDER_ID	SHIP_DATE
Europe	Finland	Vegetables	Offline	C	2012-10-09T00:00:00Z	642134416	2012-11-2
Sub-Saharan Africa	Rwanda	Fruits	Online	L	2012-09-05T00:00:00Z	699160754	2012-09-11
Asia	Japan	Household	Offline	M	2014-07-16T00:00:00Z	747796285	2014-07-2
Europe	Romania	Beverages	Online	M	2012-02-19T00:00:00Z	756839835	2012-03-11
Central America and the Caribbean	Bolivia	Personal Care	Online	H	2015-06-14T00:00:00Z	315402734	2015-08-01

4. **Modes of visualization in the Query Result tab:** You can select any of the four modes to visualize the results of the SQL query report you generate.



The four modes of visualization, when you view the reports generated on a SQL query, are:

- **Base Query:** This type of view is by default. Query written in the SQL editor is the Base Query.
- **Table:** You can view the SQL results in tabular form. By selecting this view, a Column drop zone appears which enables you to drag and drop selected columns from the Table browser. By dropping the selected columns in the drop zone, you can view only those columns in the Query Result tab. Select the cross mark beside the Column name to remove it from the drop zone.
- **Pivot:** You can view the results of the SQL query in pivot format. By selecting this format, a Columns, Rows, and Values drop zone appears where you can drag and drop the selected columns, rows or values from the Tables browser.

Note

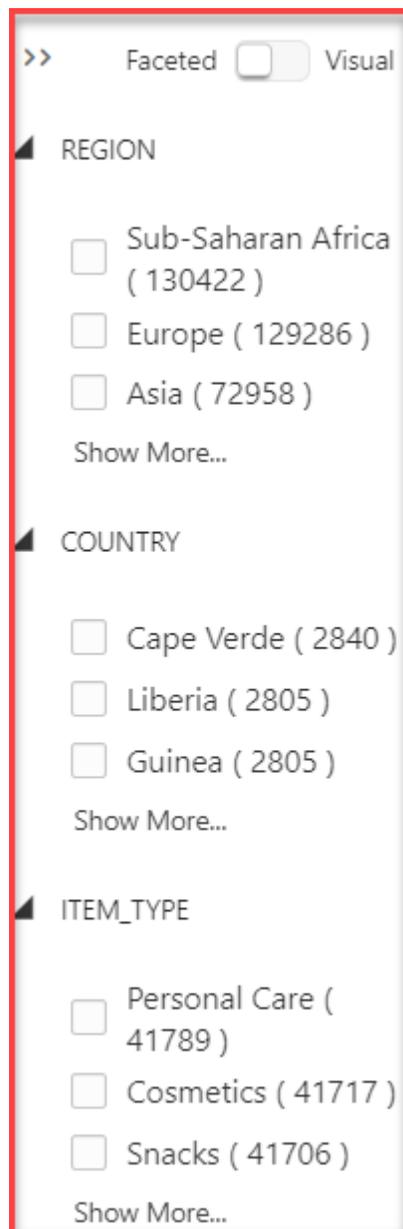
Values must be a NUMBER type.

- **Chart:** You can view the SQL results in the form of a chart. By selecting this view an X-axis and Y-axis drop zone appears. Drag and drop selected columns from the Table browser to the drop zone. You must ensure that only the columns with NUMERIC data type can be dropped in the Y axis. Otherwise, the display result would fail with a `Must be a NUMBER type` error. You can add multiple values to the Y-axis. To view the results in the chart view of only a particular y axis, select the Y axis value from the drop-down.
5. **Modes of visualization of reports generated on an Analytic View:** You can select any of the three modes to visualize the results of the report you generate on an Analytic View.

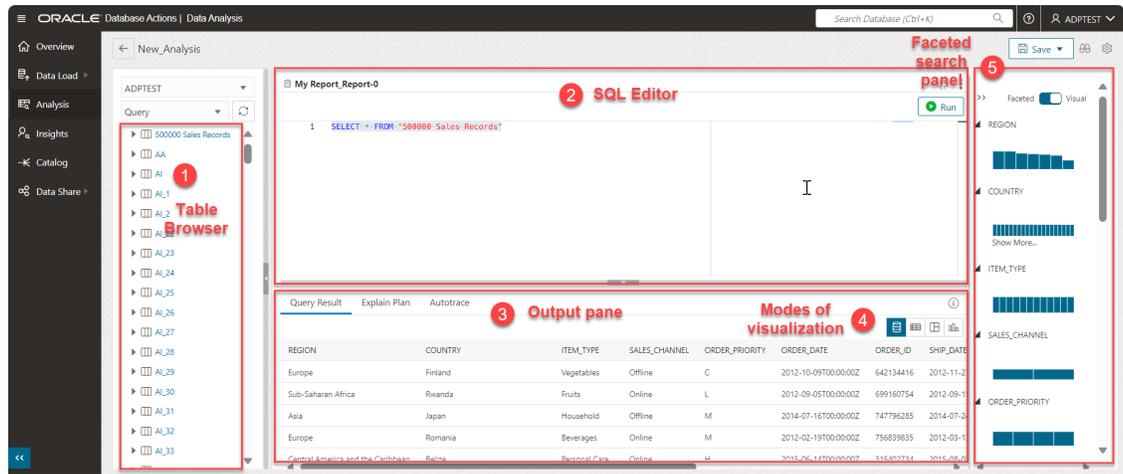


The three modes of visualization when you view the reports generated on an Analytic View are:

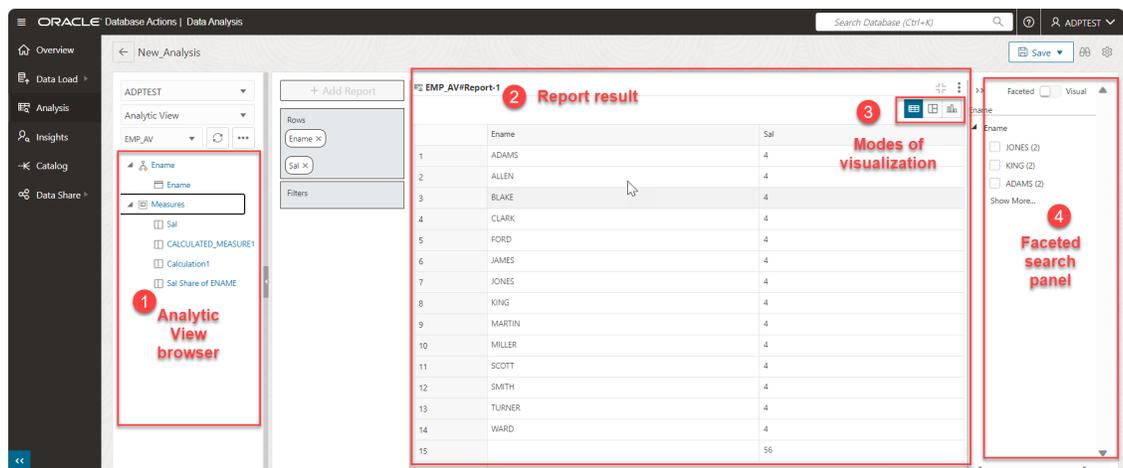
- **Table:** You can view the SQL results in tabular form. By selecting this view, a Rows and Filters drop zone appears which enables you to drag and drop selected Hierarchies and Measures from the Analytic View browser. This way you can view the report results that consist of the selected hierarchies and measures.
 - **Pivot:** You can view the results of the Analytic View report in the pivot format. By selecting this format, a Columns, Rows, Values and Filters drop zone appears where you can drag and drop the selected hierarchies and measures from the Analytic View browser. Note: Values must be a NUMBER type.
 - **Chart:** You can view the report you generate on an Analytic View in the form of a chart. By selecting this view an X-axis, Y-axis, and Filters drop zone appears. Drag and drop selected columns from the Table browser to the drop zone. You must ensure that only the columns with NUMERIC data type can be dropped in the Y axis. Otherwise, the display result would fail with a `Must be a NUMBER type` error. You can add multiple values to the Y-axis. You can select Horizontal and Vertical from the drop-down to view Horizontal and Vertical Charts respectively. You also have the option to select Area Chart, Bar Chart, Line Chart, and Pie Chart from the drop-down.
6. **Faceted search panel:** For the reports you generate on a SQL query and an Analytic View, you can view a Faceted search column. For a SQL report, this panel allows you to add filters to the report. The tool generates a filter for each value in the column that is retrieved from the query result. You can filter different columns on the faceted search panel and view the results in the Query result to get only the data you wish to view. You can view the data retrieved from the SQL query in either text or visual format. For reports you generate on an Analytic View, select **Faceted** from the radio button. This filter behaves differently than the Faceted search you generate on an SQL Query. See *Adding filters to a report you generate on an Analytic View*.



The Analyses page when you create a report on an SQL query looks like this:



The Analyses page when you create a report on an Analytic View looks like this:



The following topics describes how to create a report and access the Analyses page:

[Creating Reports](#)

- You can create Reports using either of the following ways:
 - [On a Query](#)
 - [On an Analytic View](#)
- Adding filters to a report you generate on an Analytic View

Creating Reports on a Query

This section describes the steps to create reports on an SQL query.

- From the Analysis home page, select any of the Tables you want to create a report on. You will view the Analyses page with a default query displayed on the SQL editor.

Note

By default, you will view "Select * from the <TableName> you select.

- Click **Run** to run the SQL statement.

The Query Results tab displays the result in whichever mode you select. The default view is Base Query.

- Add a filter to the report by displaying the Sales Records of only **Asia** region. Select **Asia** from the faceted search panel.

The screenshot shows the Oracle BI Reporting interface. On the left, there is a 'Columns' panel with various filters like REGION, COUNTRY, ITEM_TYPE, etc. The main area displays a report titled 'My Report_Report-2' with a SQL query: 'SELECT * FROM *500000:Sales_Records*'. Below the query, there are tabs for 'Query Result', 'Explain Plan', and 'Autotrace'. The 'Query Result' tab is active, showing a table with columns: REGION, COUNTRY, ITEM_TYPE, SALES_CHANNEL, ORDER_PRIORITY, ORDER_DATE, ORDER_ID, and SHIP_DATE. The table contains 10 rows of data, all with 'Asia' in the REGION column. On the right, there is a 'Faceted Search' panel with a 'REGION' section where 'Asia (72958)' is selected, and a 'COUNTRY' section with 'Cape Verde (2840)', 'Liberia (2805)', and 'Guinea (2805)' listed. There is also an 'ITEM_TYPE' section with 'Personal Care (41789)', 'Cosmetics (41717)', and 'Snacks (41706)' listed.

REGION	COUNTRY	ITEM_TYPE	SALES_CHANNEL	ORDER_PRIORITY	ORDER_DATE	ORDER_ID	SHIP_DATE
Asia	Bangladesh	Baby Food	Offline	C	2011-02-03T00:00:00Z	286749017	2011-02-20T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2011-02-23T00:00:00Z	338821637	2011-04-11T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2011-04-15T00:00:00Z	482129275	2011-04-30T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2011-07-13T00:00:00Z	816956937	2011-07-27T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2012-02-17T00:00:00Z	603382718	2012-04-07T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2012-09-23T00:00:00Z	389808499	2012-10-28T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2012-10-13T00:00:00Z	441881120	2012-10-28T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2013-02-07T00:00:00Z	139072167	2013-03-26T00:00:00Z
Asia	Bangladesh	Baby Food	Offline	C	2013-03-22T00:00:00Z	972089040	2013-04-03T00:00:00Z

- You will view a funnel icon in the Query Result tab which displays a filter with the **REGION** column as **Asia**. The Query Result will display only the records with **REGION** as **Asia**.

Creating Reports on an Analytic View

This section describes the steps to create reports on an Analytic View:

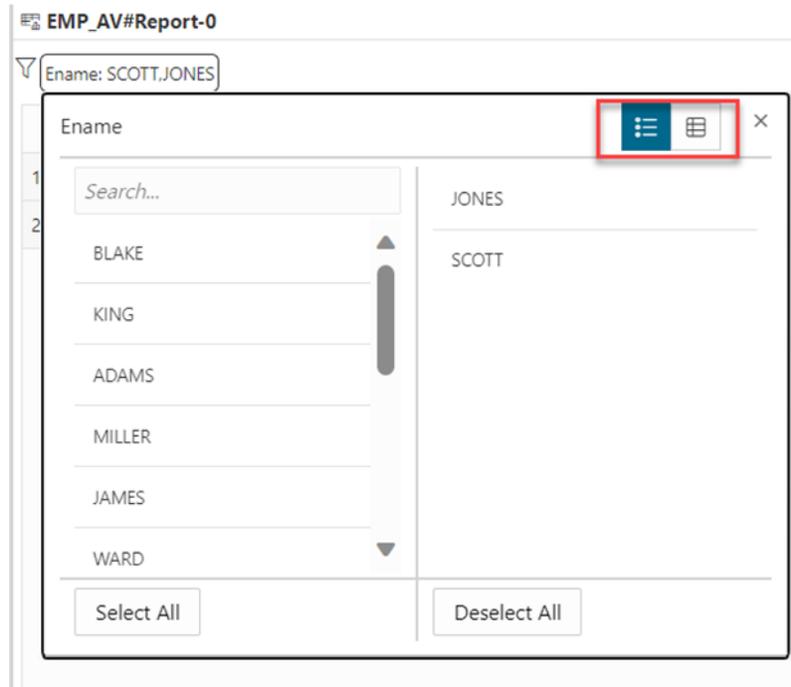
- From the Analysis home page, under the Analytic View section, select any of the Analytic Views you want to create a report on. You will view the Analyses page with a default report displayed as the output.
- Drag and drop hierarchies and measures from the Analytic View browser to edit the results you view in the output. For more information, refer to Working with Reports.
- Click **Expand Report** to expand the view of the report and click **Collapse Report** to minimize the view of the report. The default view of the report you generate on an Analytic View is Pivot.

Adding filters to a report you generate on an Analytic View

Let us add a filter to the report you generated on an Analytic View. Let's say you wish to view the salary of an employee named **SCOTT**.

- From the Analyses page which displays the report you generated on an **Employee Analytic View**, select "**Scott**" from the faceted search panel. You will view the report displaying the result of employees named **Scott**.

2. You can select more filters by selecting the values from the faceted search panel, or by clicking the funnel.
3. Clicking the funnel icon displays all the values of the Employee name column. Select **Jones** to filter the report results further displaying the salary of employees named **Scott** and **Jones**. You can view the values in a list view or a multi select view.



4. Select **Deselect All** to remove all filters.

You can now view the original report result that does not consist of any filters.

Run Natural Language Query in the Data Analysis Tool

You can query the Oracle Autonomous AI Database by using Natural Language Query rather than having to write SQL Query.

Prerequisites

To utilize the Natural Language Query, you must have:

- An OpenAI, or Cohere or An Azure OpenAI account service with credit
- An access to `DBMS_CLOUD_AI` package.

Note

- You must set your AI profile using the Natural Language Query feature. Follow the steps mentioned in this chapter to [Use Select AI to Generate SQL from Natural Language Prompts](#).
- After you have created and configured your AI profile, set your AI profile in [Data Studio Preferences](#) wizard on the Connections page to use AI features such as generating SQL from Natural Language in the Data Studio tool.

You can run Natural Language Queries on both Analytic Views and Tables.

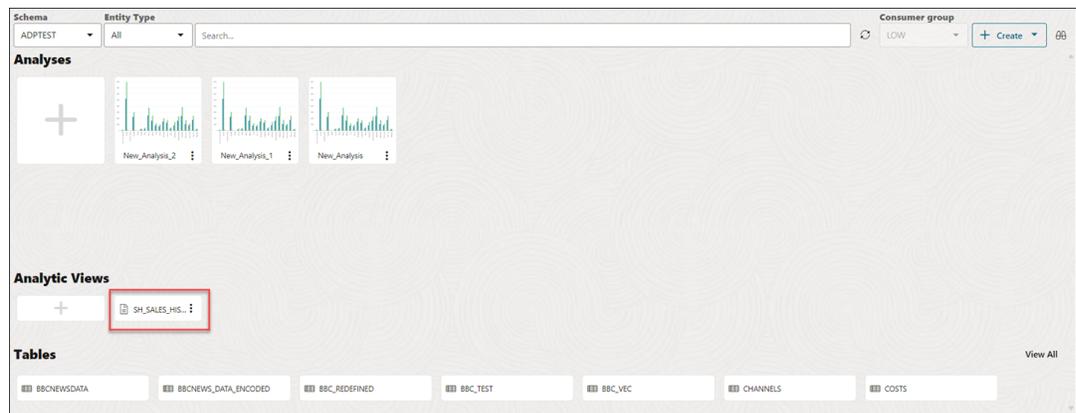
- [Generate SQL Queries From Natural Language on Analytic Views](#)
You can create SQL queries on Analytic Views.
- [Generate SQL Queries From Natural Language on Tables](#)
You can create SQL queries on Tables.
- [Generate SQL Queries From Natural Language on Analytic Views](#)
To run a natural language query from SQL query on Analytic Views, perform the following steps:
- [Generate SQL Queries from Natural Language on Tables](#)
Follow the procedure mentioned below to generate SQL queries from Natural Language on Tables:

Generate SQL Queries From Natural Language on Analytic Views

To run a natural language query from SQL query on Analytic Views, perform the following steps:

Let's say you wish to view sales amount in a categorized way.

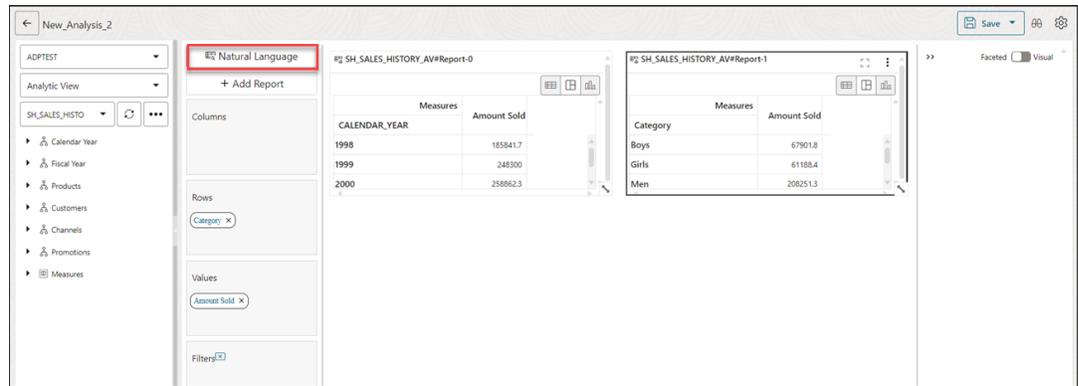
- On the Data Analysis home page, click on any of the Analytic Views you wish to query.



In this example, you will query the SH_SALES_HISTORY Analytic View.

This opens the Analyses page.

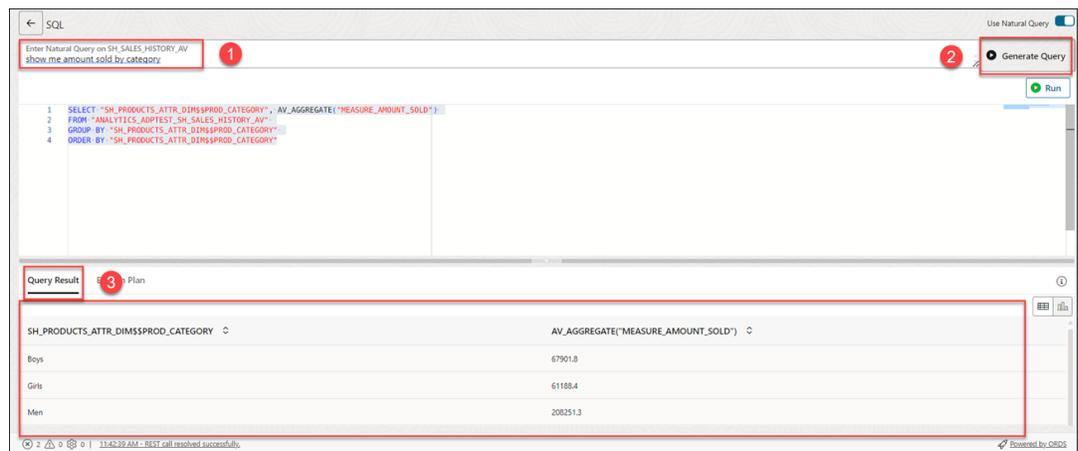
- Select **Natural Language**. Selecting Natural Language opens the SQL worksheet area with a predefined SQL query on the worksheet area.



Note

You can view **Natural Language** option only after you have configured and set AI profile using the [Use Select AI to Generate SQL from Natural Language Prompts](#) procedure and set the [Data Studio Settings](#) wizard on the Connections page.

- Enter the following natural Query you wish to run on SH_SALES_HISTORY in the Natural Query field: show me amount sold by category.
- Click **Generate Query**.



- After the tool generates the query, the **Query Result** tab displays the result of the query. You can also view the graphical representation of the contents of the PLAN_TABLE in the **Explain Plan** tab.

You can alternatively view the query result in chart view by switching the display mode in **Chart View** or **Diagram View**.

Generate SQL Queries from Natural Language on Tables

Follow the procedure mentioned below to generate SQL queries from Natural Language on Tables:

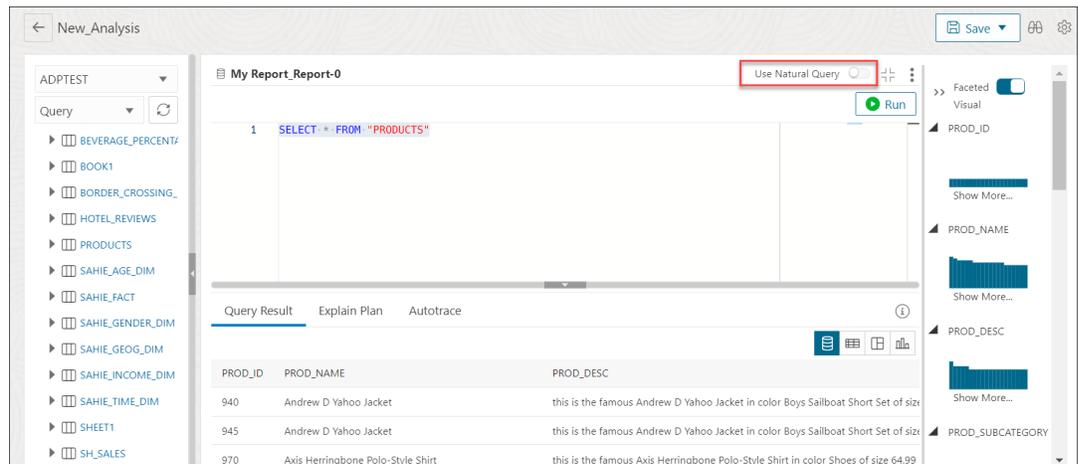
Let's say you want to view the product with the minimum product price.

1. On the Data Analysis home page, click on any of the Tables you wish to query. In this example, we select the `PRODUCTS` table.



This opens the Analyses page with a query that retrieves all the columns from the selected table `PRODUCTS`.

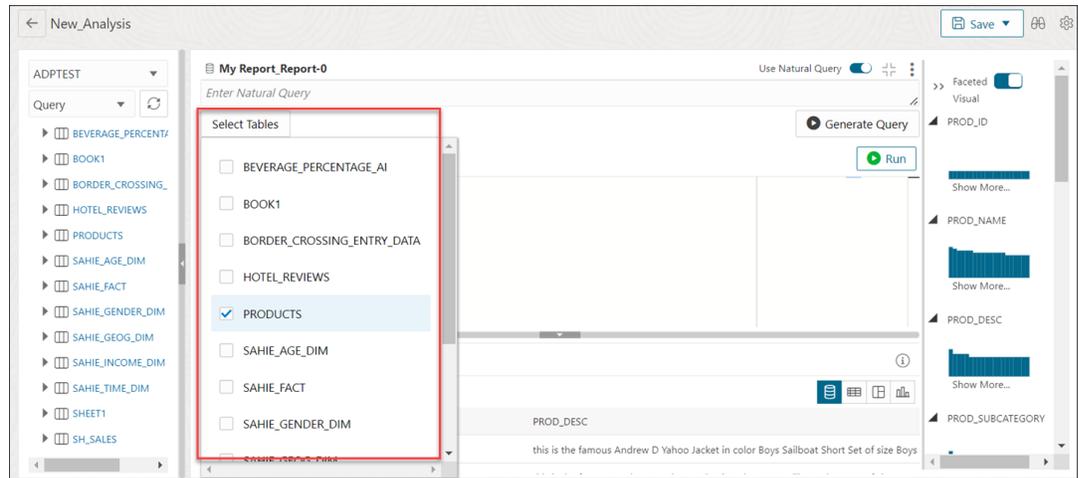
2. Select **Use Natural Query**.



3. Selecting **Use Natural Query** lets you select multiple tables from the **Select Tables** icon.

Note

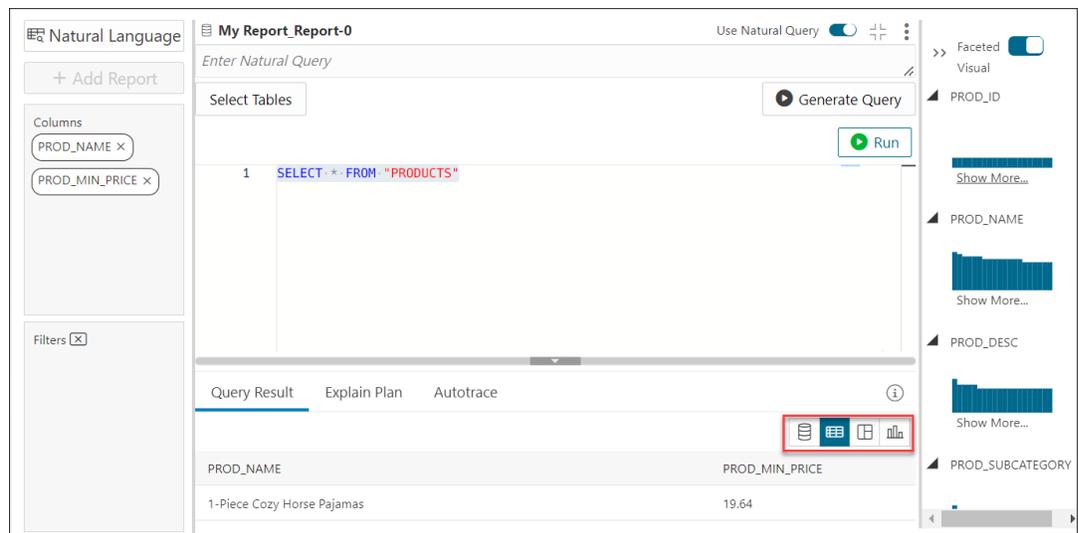
You can view **Natural Language** option only after you have configured and set AI profile using the [Use Select AI to Generate SQL from Natural Language Prompts](#) procedure and set the [Data Studio Settings](#) wizard on the Connections page.



Select the tables you wish to generate the SQL query from. You can use the columns from the selected table in the text field where you enter the Natural Query.

Note

You can click Tabular View, drag and drop columns from the navigator to the Columns and Filter drop area to select the intended columns to query.



- Click **Base Query** mode of visualization to enter Natural Language. Enter the following in the Natural Query Text field: show me the minimum product price with the product.

Click **Generate Query**.

Selecting **Generate Query** converts the Natural language to the equivalent SQL query and displays results under the Query Results tab.

You can alternatively view the query result in tabular view, pivot, and chart view. You can drag and drop rows, columns, and filters from the Tables Browser to the drop area.

Note

From the Chart view you can view the result in horizontal and vertical sheet.

Generate Natural Language explanation from SQL Query using AI Explain

You can translate SQL queries to natural language that is understood by you.

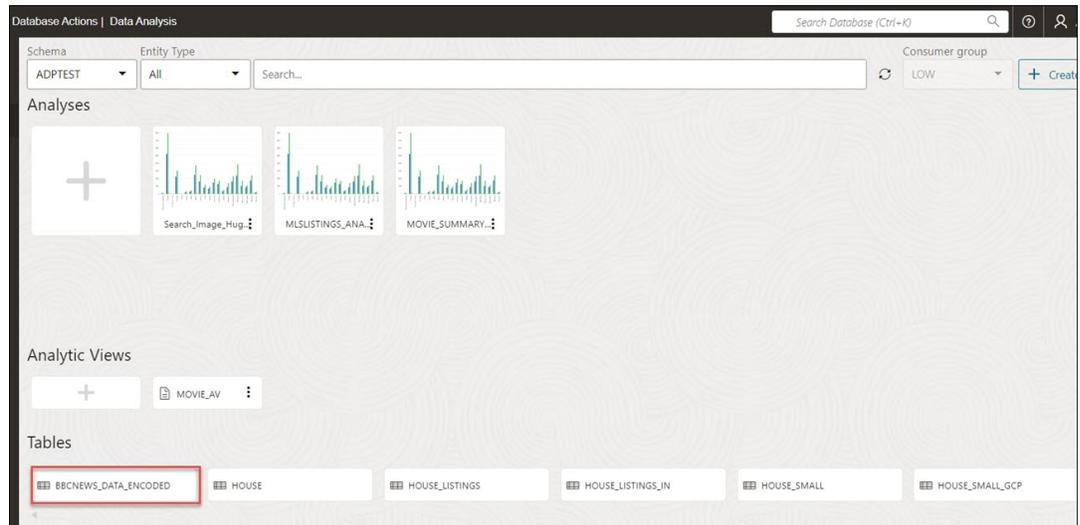
Note

This feature is available for SQL based reports on Queries and not on reports on Analytic Views.

To run an SQL query and view its Natural Language explanation, perform the following steps:

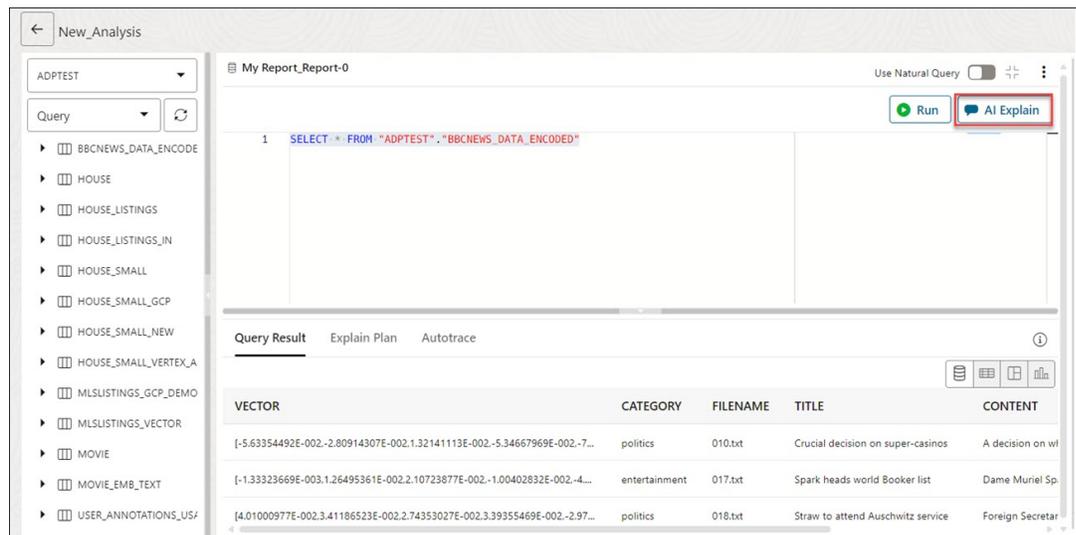
Let's say you wish to view the explanation of a query that retrieves all records of a table.

1. On the Data Analysis home page, click on any of the Tables you wish to query.

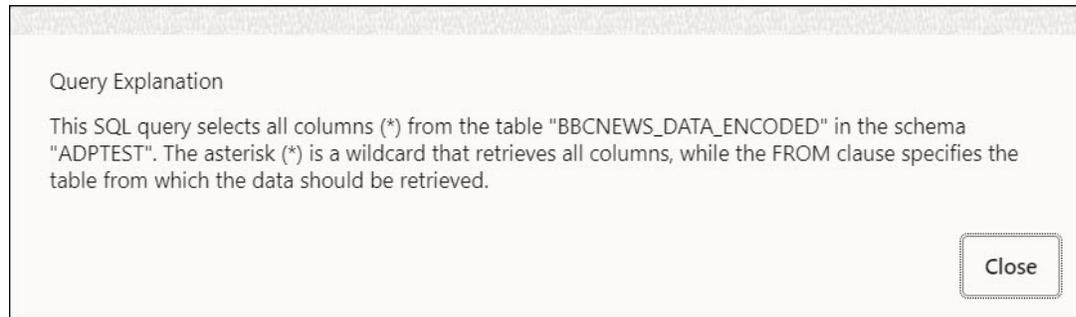


In this example, you will query the `BBCNEWS_DATA_ENCODED` table. This opens the Analyses page.

- You can view the default query displayed on the worksheet area that retrieves all the records of the table along with the columns displayed in the Query Results tab of the output pane.



- Click **AI Explain**.



You can view a Query Explanation dialog box that explains the query displayed on the worksheet area.

Click **Close** to close the dialog box.

Vector Search in the Data Analysis Tool

You can use Oracle AI Vector Search capabilities to search for relevant text from your source data on a specific column.

The Vector search helps quickly locate and match data that is similar to the text you provide as your search input.

Prerequisites

For using the vector search feature in the Data Analysis tool, you need the following:

- You must have 26ai Database with Vector support.
- The ORDS REST APIs must support VECTOR data type.
- You must use the Data Studio Settings page to select the same credential you use for vector search.
- The source data must already have a non-numeric column to be embedded first.
- To utilize the Vector Search in the Data Analysis tool, you must have:
 - An OpenAI, or Cohere or An Azure OpenAI account service with credit
 - An access to `DBMS_VECTOR` and `DBMS_VECTOR_CHAIN.UTL_TO_EMBEDDING` package.

Note

- You must set your AI profile. Follow the steps mentioned in this chapter to [Use Select AI to Generate SQL from Natural Language Prompts](#).
- After you have created and configured your AI profile, set your AI profile in the [Data Studio Settings](#) wizard on the Data Studio navigation menu to use AI features such as generating SQL from Natural Language and Vector Search in the Data Studio tool.

The Data Analysis tool utilizes the Vector Utility PL/SQL package `DBMS_VECTOR` and `DBMS_VECTOR_CHAIN.UTL_TO_EMBEDDING` to provide the third-party REST APIs that let you interact with external embedding models such as Cohere, Google AI, Hugging Face, Oracle

Cloud Infrastructure (OCI) Generative AI, OpenAI, or Vertex AI. You must understand the terms of using third-party embedding models.

Note

Certain features of the database may allow you to access services offered separately by third-parties, for example, through the use of JSON specifications that facilitate your access to REST APIs.

Your use of these features is solely at your own risk, and you are solely responsible for complying with any terms and conditions related to use of any such third-party services. Notwithstanding any other terms and conditions related to the third-party services, your use of such database features constitutes your acceptance of that risk and express exclusion of Oracle's responsibility or liability for any damages resulting from such access.

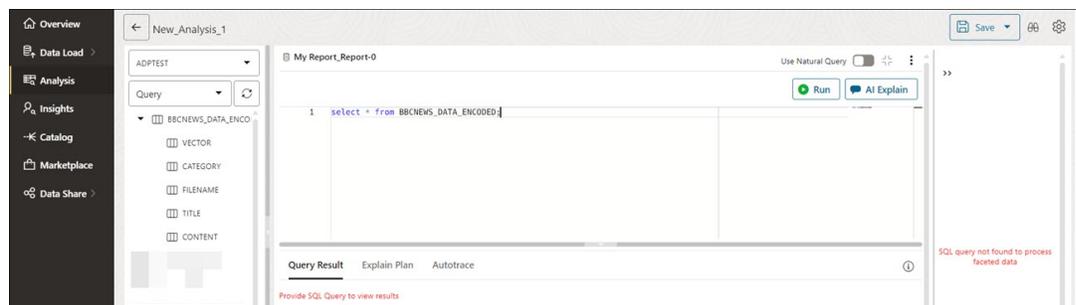
Suppose you are a finance enthusiast and you want to learn about finance through news to gain better sense of financial products and investments. In this example, you can use the Vector Search feature to find news related to the word `Bank`.

Consider a table that has the columns `CATEGORY`, `FILENAME`, `TITLE`, `CONTENT` and a different column labeled `VECTOR`, that contains the vector representation of the `CONTENT` column. Since we can use vector search on source columns with non-numeric data (image or text), any of the non-numeric column can be embedded and stored as a vector column. But in this example, we need to search for relevant text from `CONTENT` column.

You can use the Data Analysis tool to perform vector search.

To indicate which column's embedding is kept in the vector column, we must first map the appropriate column with the appropriate vector using the Data Analysis tool, search for similar text, and then run the query. The Query Result tab will display the content similar to `Bank`.

1. From the Data Analysis tool, click **+** under Analyses to create a new Analysis.
2. Select the Schema from **Schema** drop-down and Query from the **Type** drop-down.
3. Select the table you want to query from the table browser. In this example, you will select `BBCNEWS_DATA_ENCODE`.



4. Enter the Select statement in the SQL Worksheet edit area to retrieve all the columns from the `BBCNEWS_DATA_ENCODE` table.

Specify the following command:

```
Select * from BBCNEWS_DATA_ENCODE;
```

5. Click **Run**.

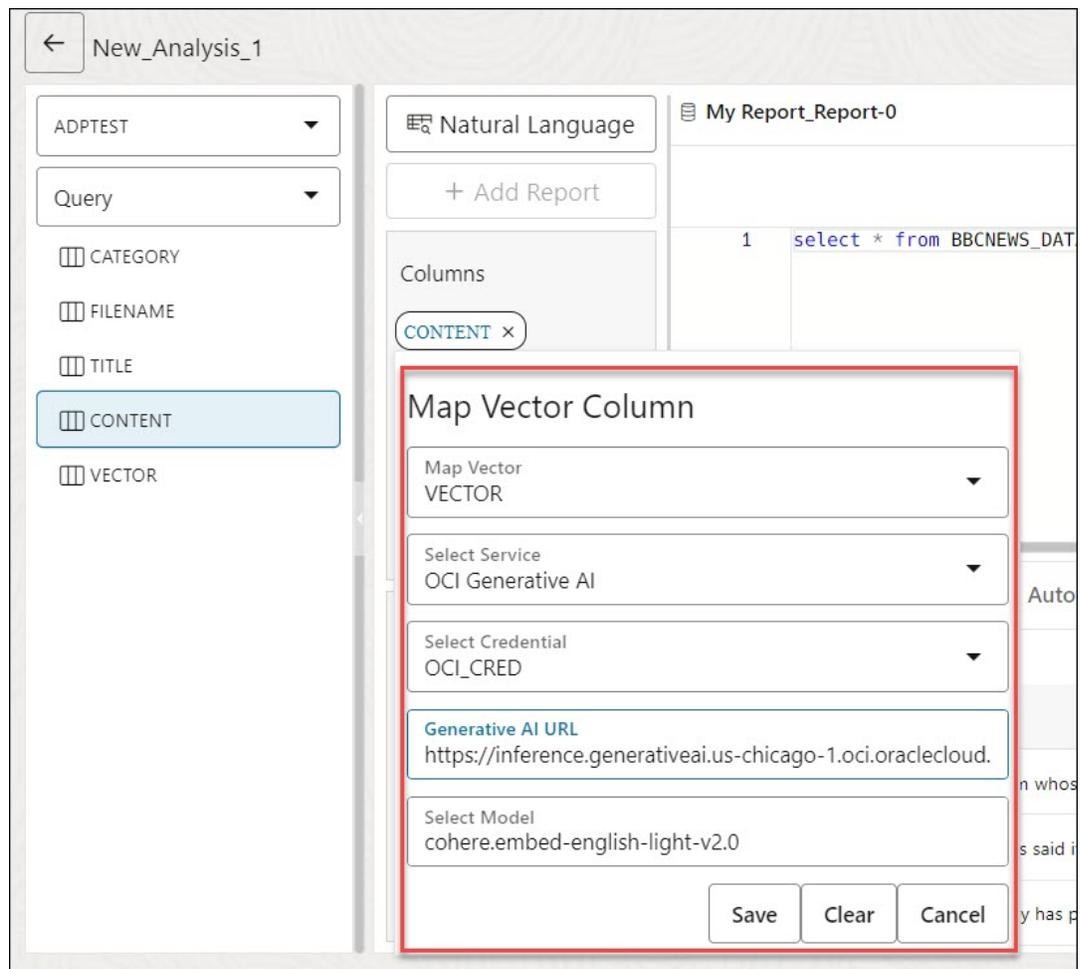
VECTOR	CATEGORY	FILENAME	TITLE	CONTENT
[-5.63354492E-002,-2.80914307E-002,1.32141113E-002,-5.34667969E-002,-7...	politics	010.txt	Crucial decision on super-casinos	A decision on w
[-1.33323669E-003,1.26495361E-002,2.10723877E-002,-1.00402832E-002,-4...	entertainment	017.txt	Spark heads world Booker list	Dame Muriel Sp
[4.01000977E-002,3.41186523E-002,2.74353027E-002,3.39355469E-002,-2.97...	politics	018.txt	Straw to attend Auschwitz service	Foreign Secretar

You can view all the columns of the BBCNEWS_DATA_ENCODE table in the Query Results tab.

6. Select tabular view of mode from the icons that display various modes of visualizing the query result.

CATEGORY
business
entertainment
politics

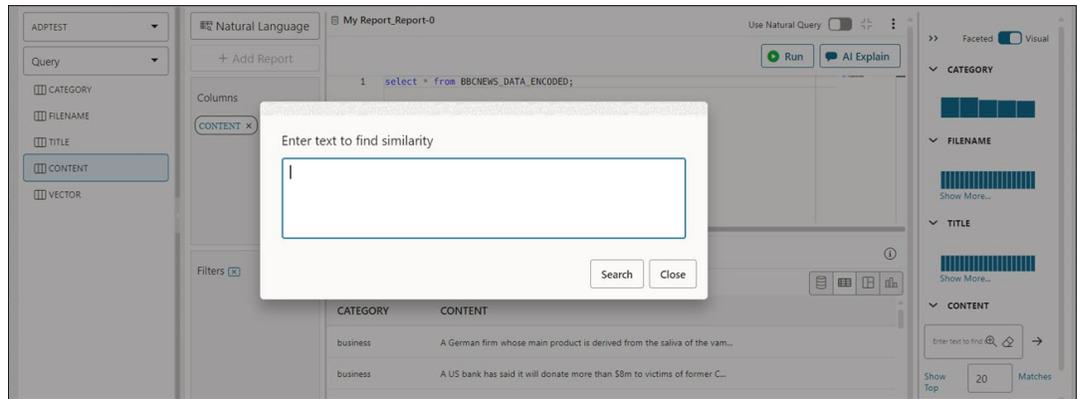
7. Drag and drop the **CONTENT** column to the Column drop area and click it to map the vector column.



8. On the Map Vector Column wizard, specify the following field values:
 - **Map Vector:** Select the column that consists of the vector embeddings of the selected column from the drop-down list. In this example, it is *VECTOR*.
 - **Select Service:** Select the AI service you will use to utilize this feature of the tool. In this example, it is *OCI Generative AI*.
 - **Select Credential:** Enter the credential you create from the Data Studio Settings to access the third party service provider. You require these credentials to enable access during REST API calls to your chosen third-party service provider.
 - **Generative AI URL:** Specify the Generative AI URL you will use to call the third-party service provider, such as Cohere, Google AI, Hugging Face, Oracle Cloud Infrastructure (OCI) Generative AI, OpenAI, or Vertex AI.
 - **Select Model:** Specify the embed model you will use to generate embeddings from the text. In this example, you will enter *cohere.embed-english-light-v2.0*.

Click **Save** to save the mapping. Select **Clear** to clear the field values you mention in this step.

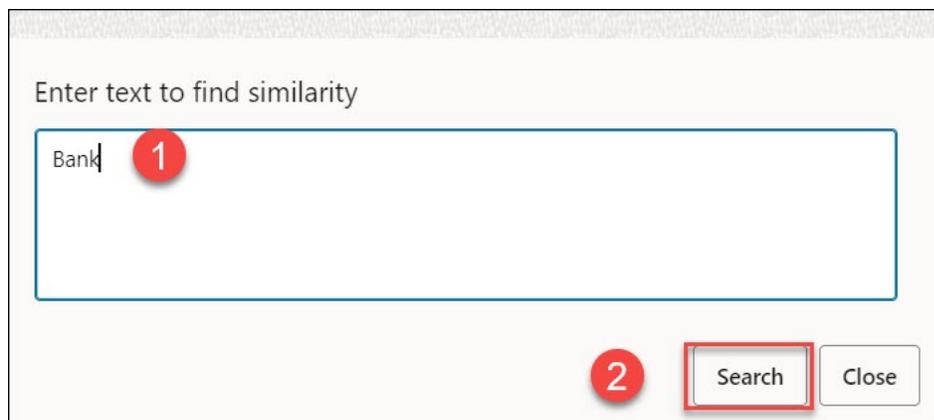
9. On the Faceted Filters, you can view text box under the Content column. Click the + sign on it to expand the text box.



10. Enter the following text in the text box to content similar to the word `Bank` in the `CONTENT` column:

`BANK`

Click **Search**.



11. The Query Results tab of the Output panel displays the content column that consists of content related to `Bank`.

The screenshot shows the Oracle Data Analysis tool interface. At the top, there's a 'Natural Language' section with a 'Run' button and an 'AI Explain' button. Below that, a query is displayed: `1 select * from BBCNEWS_DATA_ENCODED;`. The main area shows the 'Query Result' tab, which displays a table with one column named 'CONTENT'. The first row of data is highlighted with a red box and contains the text: "Former Scotland international Finlay Calder fears civil war at the SRU...". To the right of the table, there are several faceted views: 'CATEGORY', 'FILENAME', 'TITLE', and 'CONTENT'. The 'CONTENT' facet has a search box with the text 'Bank' and a 'Show More...' button. The 'Show Top' button is set to '20' matches.

You can hover over each content to view its expanded form.

This screenshot shows the same interface as the previous one, but with the 'CONTENT' row expanded. The expanded text is displayed in a larger font and is enclosed in a red box. The text reads: "Former Scotland international Finlay Calder fears civil war at the SRU could seriously hamper his country's RBS Six Nations campaign. Four members of the executive board, including the chairman, David Mackay, have resigned after a simmering row. And Calder said: "This is terrible news for every level of Scottish rugby. "David is a successful businessman and I thought that if anybody could transform the negative atmosphere and rising debt level, it was him." Mackay's executive board has been in a power struggle with the general committee, which contains members elected by Scotland's club sides. "He has been driven out by people who seem happier waging civil war than addressing the central issue that professional rugby can't be run by amateurs," said Calder. "In fact, I don't understand why we are still having this argument 10 years after professionalism arrived. "But I don't believe the rest of the SRU will take this lying down. "I think the banks will be dismayed at this decision and, ultimately, it is them who pull the strings." "So I wouldn't be surprised if they reviewed their position. But, in the wider picture, what message does this send out?" He thought the". Below this expanded text, the truncated text from the previous screenshot is visible: "Former Scotland international Finlay Calder fears civil war at the SRU...".

Using Calculation Templates

The Data Analysis tool provides templates for all of the calculations typically in demand for business intelligence applications.

The following topics describe the types of calculations available as calculation templates in the tool.

- [Cumulative Aggregates](#)
- [Prior and Future Period](#)
- [Period to Date](#)
- [Parallel Period](#)
- [Moving Aggregates](#)
- [Share](#)
- [Rank](#)

- [Cumulative Aggregates](#)
Cumulative calculations start with the first time period and calculate up to the current member, or start with the last time period and calculate back to the current member.
- [Prior and Future Period](#)
The Data Analysis tool provides several calculations for prior or future time periods.
- [Period to Date](#)
Period-to-date functions perform a calculation over time periods with the same parent up to the current period.
- [Parallel Period](#)
Parallel periods are at the same level as the current time period, but have different parents in an earlier period. For example, you may want to compare current sales with sales for the prior year at the quarter and month levels.
- [Moving Aggregates](#)
Moving aggregates are performed over the time periods surrounding the current period.
- [Share](#)
Share calculates the ratio of a measure's value for the current dimension member to the value for a related member of the same dimension.
- [Rank](#)
Rank orders the values of a dimension based on the values of the selected measure. When defining a rank calculation, you choose the dimension, a hierarchy, and the measure.

Cumulative Aggregates

Cumulative calculations start with the first time period and calculate up to the current member, or start with the last time period and calculate back to the current member.

The tool provides several aggregation methods for cumulative calculations:

- **Cumulative Average:** Calculates a running average across time periods.
- **Cumulative Maximum:** Calculates the maximum value across time periods.
- **Cumulative Minimum:** Calculates the minimum value across time periods.
- **Cumulative Total:** Calculates a running total across time periods.

You can choose the measure, the time dimension, and the hierarchy. For selecting the time range see "Choosing a Range of Time Periods" in *Oracle OLAP User's Guide*.

Cumulative Calculation Example

This template defines a calculated measure using Cumulative Minimum.

Cumulative minimum of `SALES` in the `TIME` dimension and `TIME.CALENDAR` hierarchy within ancestor at level `TIME.CALENDAR.YEAR`. Total from `beginning to current` member.

These are the results of a query against the calculated measure, which displays values for the descendants of calendar year 2021. The minimum value for quarters begins with Q1-21 and ends with Q4-21, and for months begins with Jan-21 and ends with Dec-21.

TIME	TIME_LEVEL	SALES	MIN_SALES
Q1.21	CALENDAR_QUARTER	32977874	32977874
Q2.21	CALENDAR_QUARTER	35797921	32977874
Q3.21	CALENDAR_QUARTER	33526203	32977874
Q4.21	CALENDAR_QUARTER	41988687	32977874

JAN-21	MONTH	11477898	11477898
FEB-21	MONTH	10982016	10982016
MAR-21	MONTH	10517960	10517960
APR-21	MONTH	11032057	10517960
MAY-21	MONTH	11432616	10517960
JUN-21	MONTH	13333248	10517960
JUL-21	MONTH	12070352	10517960
AUG-21	MONTH	11108893	10517960
SEP-21	MONTH	10346958	10346958
OCT-21	MONTH	14358605	10346958
NOV-21	MONTH	12757560	10346958
DEC-21	MONTH	14872522	10346958

Prior and Future Period

The Data Analysis tool provides several calculations for prior or future time periods.

Here are the calculations used for for prior or future time periods:

- **Prior Period:** Returns the value of a measure at an earlier time period.
- **Difference From Prior Period:** Calculates the difference between values for the current time period and an earlier period.
- **Percent Difference From Prior Period:** Calculates the percent difference between the values for the current time period and an earlier period.
- **Future Period:** Returns the value of a measure at a later time period.
- **Difference From Future Period:** Calculates the difference between the values for the current time period and a later period.
- **Percent Difference From Future Period:** Calculates the percent difference between the values for the current time period and a later period.

When creating a calculation for prior or future time periods, you choose the measure, the time dimension, the hierarchy, and the number of periods from the current period.

Prior Period Example

This template defines a calculated measure using Prior Period:

Prior period for measure `SALES` in `TIME` dimension and `TIME.CALENDAR` hierarchy `1` period ago.

These are the results of a query against the calculated measure. The `PRIOR_PERIOD` column shows the value of Sales for the preceding period at the same level in the Calendar hierarchy.

TIME	TIME_LEVEL	SALES	PRIOR_PERIOD
2020	CALENDAR_YEAR	136986572	144290686
2021	CALENDAR_YEAR	140138317	136986572
Q1.20	CALENDAR_QUARTER	31381338	41988687
Q2.20	CALENDAR_QUARTER	37642741	31381338
Q3.20	CALENDAR_QUARTER	32617249	37642741
Q4.20	CALENDAR_QUARTER	35345244	32617249
Q1.21	CALENDAR_QUARTER	36154815	35345244
Q2.21	CALENDAR_QUARTER	36815657	36154815
Q3.21	CALENDAR_QUARTER	32318935	36815657
Q4.21	CALENDAR_QUARTER	34848911	32318935

Period to Date

Period-to-date functions perform a calculation over time periods with the same parent up to the current period.

These functions calculate period-to-date:

- **Period to Date:** Calculates the values up to the current time period.
- **Period to Date Period Ago:** Calculates the data values up to a prior time period.
- **Difference From Period to Date Period Ago:** Calculates the difference in data values up to the current time period compared to the same calculation up to a prior period.
- **Percent Difference From Period To Date Period Ago:** Calculates the percent difference in data values up to the current time period compared to the same calculation up to a prior period.

When creating a period-to-date calculation, you can choose from these aggregation methods:

- Sum
- Average
- Maximum
- Minimum

You also choose the measure, the time dimension, and the hierarchy.

Period to Date Example

This template defines a calculated measure using Period to Date.

Gregorian Year to date for SALES in the TIME dimension and TIME.CALENDAR hierarchy. Aggregate using MINIMUM from the beginning of the period.

These are the results of a query against the calculated measure. The MIN_TO_DATE column displays the current minimum SALES value within the current level and year.

TIME	TIME_LEVEL	SALES	MIN_TO_DATE
Q1.21	CALENDAR_QUARTER	36154815	36154815
Q2.21	CALENDAR_QUARTER	36815657	36154815
Q3.21	CALENDAR_QUARTER	32318935	32318935
Q4.21	CALENDAR_QUARTER	34848911	32318935
JAN-21	MONTH	13119235	13119235
FEB-21	MONTH	11441738	11441738
MAR-21	MONTH	11593842	11441738
APR-21	MONTH	11356940	11356940
MAY-21	MONTH	13820218	11356940
JUN-21	MONTH	11638499	11356940
JUL-21	MONTH	9417316	9417316
AUG-21	MONTH	11596052	9417316
SEP-21	MONTH	11305567	9417316
OCT-21	MONTH	11780401	9417316
NOV-21	MONTH	10653184	9417316
DEC-21	MONTH	12415325	9417316

Parallel Period

Parallel periods are at the same level as the current time period, but have different parents in an earlier period. For example, you may want to compare current sales with sales for the prior year at the quarter and month levels.

The Data Analysis tool provides several functions for parallel periods:

- **Parallel Period:** Calculates the value of the parallel period.
- **Difference From Parallel Period:** Calculates the difference in values between the current period and the parallel period.
- **Percent Difference From Parallel Period:** Calculates the percent difference in values between the current period and the parallel period.

To identify the parallel period, you specify a level and the number of periods before the current period. You can also decide what happens when two periods do not exactly match, such as comparing daily sales for February (28 days) with January (31 days).

You also choose the measure, the time dimension, and the hierarchy.

Parallel Period Example

This template defines a calculated measure using Parallel Period.

Parallel period for SALES in the TIME dimension and TIME.CALENDAR hierarchy 1 TIME.CALENDAR.QUARTER ago based on position from beginning to ending of period.

These are the results of a query against the calculated measure, which lists the months for two calendar quarters. The parallel month has the same position within the previous quarter. The prior period for JUL-21 is APR-21, for AUG-21 is MAY-21, and for SEP-21 is JUN-21.

TIME	PARENT	SALES	LAST_QTR
APR-21	CY2006.Q2	11356940	13119235
MAY-21	CY2006.Q2	13820218	11441738
JUN-21	CY2006.Q2	11638499	11593842
JUL-21	CY2006.Q3	9417316	11356940
AUG-21	CY2006.Q3	11596052	13820218
SEP-21	CY2006.Q3	11305567	11638499

Moving Aggregates

Moving aggregates are performed over the time periods surrounding the current period.

The Data Analysis tool provides several aggregation methods for moving calculations:

- **Moving Average:** Calculates the average value for a measure over a fixed number of time periods.
- **Moving Maximum:** Calculates the maximum value for a measure over a fixed number of time periods.
- **Moving Minimum:** Calculates the minimum value for a measure over a fixed number of time periods.
- **Moving Total:** Returns the total value for a measure over a fixed number of time periods.

You can choose the measure, the time dimension, and the hierarchy. You can also select the range, as described in "Choosing a range of time periods" in *Oracle OLAP User's Guide*, and the number of time periods before and after the current period to include in the calculation.

Moving Aggregates Example

This template defines a calculated measure using Moving Minimum.

Moving minimum of `SALES` in the `TIME` dimension and `TIME.CALENDAR` hierarchy. Include `1` preceding and `1` following members within `level`.

These are the results of a query against the calculated measure, which displays values for the descendants of calendar year 2021. Each value of Minimum Sales is the smallest among the current value and the values immediately before and after it. The calculation is performed over all members of a level in the cube.

TIME	TIME_LEVEL	SALES	MIN_SALES
Q1.21	CALENDAR_QUARTER	32977874	32977874
Q2.21	CALENDAR_QUARTER	35797921	32977874
Q3.21	CALENDAR_QUARTER	33526203	33526203
Q4.21	CALENDAR_QUARTER	41988687	31381338
JAN-21	MONTH	11477898	10982016
FEB-21	MONTH	10982016	10517960
MAR-21	MONTH	10517960	10517960
APR-21	MONTH	11032057	10517960
MAY-21	MONTH	11432616	11032057
JUN-21	MONTH	13333248	11432616
JUL-21	MONTH	12070352	11108893
AUG-21	MONTH	11108893	10346958
SEP-21	MONTH	10346958	10346958
OCT-21	MONTH	14358605	10346958
NOV-21	MONTH	12757560	12757560
DEC-21	MONTH	14872522	12093518

Share

Share calculates the ratio of a measure's value for the current dimension member to the value for a related member of the same dimension.

You can choose whether the related member is:

- **Top of hierarchy:** Calculates the ratio of each member to the total.
- **Member's parent:** Calculates the ratio of each member to its parent.
- **Member's ancestor at level:** Calculates the ratio of each member to its ancestor, that is, a member at a specified level higher in the hierarchy.

When creating a share calculation, you can choose the measure, dimension, and hierarchy. You also have the option of multiplying the results by 100 to get percentages instead of fractions.

Share Example

This template defines a calculated measure using SHARE:

Share of measure `SALES` in `PRODUCT.PRIMARY` hierarchy of the `PRODUCT` dimension as a ratio of `top of hierarchy`.

These are the results of a query against the calculated measure. The `TOTAL_SHARE` column displays the percent share of the total for the selected products.

PRODUCT	PROD_LEVEL	SALES	TOTAL_SHARE
Total Product	TOTAL	144290686	100
Hardware	CLASS	130145388	90
Desktop PCs	FAMILY	78770152	55
Portable PCs	FAMILY	19066575	13
CD/DVD	FAMILY	16559860	11
Software/Other	CLASS	14145298	10
Accessories	FAMILY	6475353	4
Operating Systems	FAMILY	5738775	4
Memory	FAMILY	5430466	4
Modems/Fax	FAMILY	5844185	4
Monitors	FAMILY	4474150	3
Documentation	FAMILY	1931170	1

Rank

Rank orders the values of a dimension based on the values of the selected measure. When defining a rank calculation, you choose the dimension, a hierarchy, and the measure.

You can choose a method for handling identical values:

- **Rank:** Assigns the same rank to identical values, so there may be fewer ranks than there are members. For example, it may return 1, 2, 3, 3, 4 for a series of five dimension members.
- **Dense Rank:** Assigns the same minimum rank to identical values. For example, it may return 1, 2, 3, 3, 5 for a series of five dimension members.
- **Average Rank:** Assigns the same average rank to identical values. For example, it may return 1, 2, 3.5, 3.5, 5 for a series of five dimension members.

You can also choose the group in which the dimension members are ranked:

- **Member's level:** Ranks members at the same level.
- **Member's parent:** Ranks members with the same parent.
- **Member's ancestor at level:** Ranks members with the same ancestor at a specified level higher in the hierarchy.

Rank Example

This template defines a calculated measure using Rank:

Rank members of the `PRODUCT` dimension and `PRODUCT.PRIMARY` hierarchy based on measure `SALES`. Calculate rank using `RANK` method with `member's parent` in order `lowest to highest`. Rank NA (null) values `nulls last`.

These are the results of a query against the calculated measure in which the products are ordered by RANK:

PRODUCT	SALES	RANK
Monitors	4474150	1
Memory	5430466	2
Modems/Fax	5844185	3
CD/DVD	16559860	4
Portable PCs	19066575	5
Desktop PCs	78770152	6

Access the Spreadsheet Add-in by OCI IAM Domains

Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) uses identity domains to provide identity and access management features such as authentication, Single Sign-On (SSO), and identity lifecycle management for OCI as well as for Oracle and non-Oracle applications, whether SaaS, cloud hosted, or on-premises.

You can enable use of the spreadsheet add-ins with Oracle Identity and Access Management (IAM) connection. You must:

- [Create a domain integrated application](#)
- [Enable IAM Login for Autonomous Database Schema](#)
- [Create a connection file](#)

Note

The integrated application creation, OAuth configuration, and creation of the connection file should be performed by an `admin` user.

After the connection file is created, it can be shared with any user to enable domain login.

Prerequisites

- Users in OCI must have the necessary policies to manage Oracle Autonomous Databases and OCI IAM domains. For more information on policy reference of all the services, see [Policy Reference](#).
- An Oracle Autonomous Database must be available. For more information, see [Provision an Autonomous Database Instance](#).

Create or use a domain integrated application

To use a domain integrated application in Oracle Cloud Infrastructure (OCI), you follow these steps to create and configure an identity domain and then set up policies and integrations:

1. Use an Identity Domain:
 - Sign in to the OCI Console with an administrator account into the *Default* identity domain.
 - On the OCI console navigation menu, click **Integrated applications**.
 - On the **Integrated applications** list page, select **Add application**.
 - In the Add application window, select **Confidential Application**, and then **Launch workflow**.

Add Confidential Application

- Add application details
- Configure OAuth
- Configure policy

Name
Spreadsheet Addins

Description *Optional*
Allow user to login to the database with the spreadsheet addins

Application icon ⓘ



URLs

Application URL *Optional*
Enter the URL where users access your enterprise application after successful sign in. Use the host name and port number of the App Gateway. If you have multiple instances of App Gateway, then use the host name and port number of the load balancer. The preferred format is HTTPS. Use HTTP only for testing purposes.

Custom sign-in URL *Optional*
Enter the URL where the user is redirected to sign in. Leave this field blank if you're using a default sign-in page provided by Oracle.

Custom sign-out URL *Optional*
Enter the URL where the user is directed after the sign-out process. Leave this field blank if you're using a default sign-in page provided by Oracle.

Custom error URL *Optional*
Enter the URL to which a user is redirected after an error. If you leave this field blank, the domain-specific error page URL specified in Session settings will be used. If no error URLs are configured, then the user is redirected to the identity domain error page (/uiv1/error).
Enter the URL to redirect to after linking a user between social providers and an identity domain is complete. If you leave this field blank, the Social linking callback URL, specified in Session settings will be used.

Display settings

Display in My Apps
Select if you want the application to be listed for users on their My Apps page.

User can request access
Select if you want to allow users to request access to the application from the Catalog.

Authentication and authorization

Define a more detailed authentication and authorization configuration.

Enforce grants as authorization
Select if you want to allow access to this app only if the user has been granted this app.

[Hide advanced options](#)

Tags

Add tags to organize your resources. [What can I do with tagging?](#)

Tag namespace	Tag key	Tag value
None (add a free-form tag)		

In the Add application details page, use the following table to configure application details and the display settings.

Table 16-1 Application Details and their description

Option	Description
Name	Enter a name for the confidential application. You can enter up to 125 characters. Consider keeping your application names as short as possible. In this example use, <i>Spreadsheet addins</i> .

Table 16-1 (Cont.) Application Details and their description

Option	Description
Description	Enter a description for the confidential application. You can enter up to 250 characters.
Application Icon	This field is optional. You can skip this field. Click Upload to add an icon that represents the application. This icon appears next to the name of the application on the My Apps page and the Applications page
Application URL	This field is optional. You can skip this field. Enter the URL (HTTP or HTTPS) where the user is redirected after a successful login.
Custom Login URL	This field is optional. You can skip this field. In the Custom Login URL field, you can specify a custom login URL. However, if you are using a default login page provided by Oracle Identity Cloud Service, then leave this field blank.
Custom Logout URL	This field is optional. You can skip this field. In the Custom Logout URL field, you can specify a custom logout URL. However, if you are using a default login page provided by Oracle Identity Cloud Service, then leave this field blank.
Custom Error URL	This field is optional. You can skip this field. You can enter the error page URL to which a user has to be redirected, only in case of a failure. If not specified, the tenant specific Error page URL will be used.
Display in My Apps	Do not select this check box. You select the check box only if you want the confidential application to be listed for users on their My Apps pages. In this case you need to configure the application as a resource server.
User can request access	Do not select this check-box. You select the check box only if you want end users to be able to request access to the app from their My Apps page.
Tags	Skip this field. Click Add Tag only if you want to add tags to your confidential applications to organize and identify them.

- Click **Next** to proceed to the **Configure OAuth** tab.
- On the Resource Server Configuration wizard of the **Configure OAuth** tab:

Select **Configure this application as a resource server now** to protect resources for your application now, and to make the application visible on the **My Apps** page.

Add Confidential Application

- 1 Add application details
- 2 **Configure OAuth**
- 3 Configure policy

Resource server configuration

Configure this application as a resource server now Skip for later

Configure application APIs that need to be OAuth protected

Access token expiration (seconds)

Allow token refresh
Select if you want to use the refresh token that you obtain when using the Resource Owner, Authorization Code, or Assertion grant types.

Primary audience

Enter the primary recipient where the access token of your application is processed.

Add secondary audience
Enter the secondary recipients where the access token of your application is processed.

Add scopes
Add scopes to specify which of the application's resources are available to other applications.

Previous Next Cancel

Use the following table to fill in the information in the **Configure application APIs that need to be OAuth protected** section that opens.

Table 16-2 Options and their descriptions to configure application APIs

Option	Description
Access Token Expiration	Keep the default value of 3600 seconds. Define how long (in seconds) the access token associated with your confidential application remains valid.
Is Refresh Token Allowed	Do not select this check-box. Select this check box only if you want to use the refresh token that you obtain when using the Resource Owner, Authorization Code, or Assertion grant types.
Refresh Token Expiration	Do not select this option. You can define how long (in seconds) the refresh token, which is returned with your access token and is associated with your confidential application, remains valid.
Primary Audience	Enter "ords" . This the primary recipient where the access token of your confidential application is processed.

Table 16-2 (Cont.) Options and their descriptions to configure application APIs

Option	Description
Secondary Audiences	Skip this field. You must enter the secondary recipients where the access token of your confidential application is processed, and click Add . In this example, you do not have any secondary recipients.
Add (Allowed Scopes)	To specify which parts of other applications that you want your application to access, click this button to add those scopes to your confidential application. Applications must interact securely with external partner or confidential applications. Also, applications from one Oracle Cloud service must interact securely with applications in another Oracle Cloud service. Each application has application scopes that determine which of its resources are available to other applications.

- Click **Add Scopes** and select **Add**.

Add scope

Scope
oracle.dbtools.auth.privileges.builtin.ResourceModules

Display name *Optional*

Description *Optional*

Requires user consent
Require consent for this scope configured for the application.

Add [Cancel](#)

On the Add scope wizard, specify the following field value:

- **Scope:** `oracle.dbtools.auth.privileges.builtin.ResourceModules`
- **Display name:** This is optional field.
- **Description:** This is optional field.

Click **Add**.

You have added `oracle.dbtools.auth.privileges.builtin.ResourceModules` as a scope.

Similarly add the following scopes to the confidential application:

- oracle.dbtools.sdw.user
- oracle.dbtools.ords.db-api.developer
- adp_lmd_privilege
- adp_analytics_privilege

Note

Only one confidential application for a domain can use the each scope, so if you want more than one application to use a scope the second application can be a mobile application that will refer to the confidential application scope.

Add Confidential Application

1 Add application details

2 Configure OAuth

3 Configure policy

Add scopes
Add scopes to specify which of the application's resources are available to other applications.

Scopes

<input type="checkbox"/>	Scope	Protected	Display name	Description	Requires user consent
<input type="checkbox"/>	adp_analytics_privilege	No			No
<input type="checkbox"/>	adp_lmd_privilege	No			No
<input type="checkbox"/>	oracle.dbtools.ords.db-api.developer	No			No
<input type="checkbox"/>	oracle.dbtools.auth.privileges.builtin.ResourceModules	No			No
<input type="checkbox"/>	oracle.dbtools.sdw.user	No			No

0 selected Showing 5 items

Client configuration

Configure this application as a client now Skip for later

- On the **Add Confidential Application** wizard's Client Configuration dialog, Click **Configure this application as a client now** to configure authorization information for your application now.

Add Confidential Application

- Add application details
- Configure OAuth**
- Configure policy

Client configuration

Configure this application as a client now Skip for later

Authorization

Allowed grant types ⓘ

Resource owner
 Client credentials
 JWT assertion
 Refresh token
 Device code

Authorization code
 Implicit
 SAML2 assertion
 TLS client authentication

Allow non-HTTPS URLs ⓘ

Redirect URL ⓘ

×

+ Another redirect URL

Post-logout redirect URL *Optional* ⓘ

×

+ Another post-logout redirect URL

Logout URL *Optional*

Enter the URL to be called during the logout process. When this URL is called, the resource owner session is terminated.

Client type ⓘ

Trusted Confidential

Certificate *Optional*

Allowed operations ⓘ

Introspect
 On behalf of

ID token encryption algorithm

▾

Select one of the available content encryption algorithms so that ID tokens passed through third parties, such as browsers, are encrypted.

Bypass consent

Turn on Bypass consent to overwrite the Require consent attribute for all the scopes configured for the application. Turning this option on means that no scope will require consent.

Client IP address

Anywhere Restrict by network perimeter

Token issuance policy

Authorized resources ⓘ

All Specific

Add resources

Add resources if you want your application to access the APIs of other applications.

Add app roles

Add the application roles to assign to this application. For example, add the Identity Domain Administrator role so that all REST API tasks available to the identity domain administrator will be available to the application.

- In the **Authorization** and **Token Issuance Policy** sections that open, use the following table to fill in the information.

Table 16-3 Client Configuration options and their description

Option	Description
Resource Owner	Do not select this field. Use only when the resource owner has a trust relationship with the confidential application, such as a computer operating system or a highly privileged application, because the confidential application must discard the password after using it to obtain the access token.
Client Credentials	Do not select this field. Use only when the authorization scope is limited to the protected resources under the control of the client or to the protected resources registered with the authorization server.
JWT Assertion	Do not select this field. Use only when you want to use an existing trust relationship expressed as an assertion and without a direct user approval step at the authorization server.
SAML2 Assertion	Do not select this field. Use only when you want to use an existing trust relationship expressed as a SAML2 assertion and without a direct user approval step at the authorization server.
Refresh Token	Do not select this field. Select this grant type only when you want a refresh token supplied by the authorization server, and then use it to obtain a new access token.
Authorization Code	Do not select this field. Select this grant type only when you want to obtain an authorization code by using an authorization server as an intermediary between the client application and resource owner.
Implicit	Select this field. If the application can't keep client credentials confidential for use in authenticating with the authorization server, then select this check box. For example, your application is implemented in a web browser using a scripting language such as JavaScript. An access token is returned to the client through a browser redirect in response to the resource owner authorization request (rather than an intermediate authorization).
Device Code	Do not select this field. Select the Device Code grant type only if the client doesn't have the capability to receive requests from the OAuth Authorization Server, for example, it cannot act as an HTTP server such as game consoles, streaming media players, digital picture frames, and others.

Table 16-3 (Cont.) Client Configuration options and their description

Option	Description
TLS Client Authentication	Do not select this field. Select the TLS Client Authentication grant type only to use the client certificate to authenticate with the client. If a token request comes with an X.509 client certificate and the requested client is configured with the TLS Client Authentication grant type, the OAuth service uses the Client_ID in the request to identify the client and validate the client certificate with the certificate in the client configuration. The client is successfully authenticated only if the two values match.
Allow non-HTTPS URLs	Do not select this field. Select this check box only if you want to use HTTP URLs for the Redirect URL , Logout URL , or Post Logout Redirect URL fields. For example, if you are sending requests internally, want a non-encrypted communication, or want to be backward-compatible with OAuth 1.0, then you can use an HTTP URL.
Redirect URL	Enter the following application URL where the user is redirected after authentication, <i>https://static.oracle.com/cdn/spreadsheet/red-4/redirector.html</i>
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>Provide an absolute URL. Relative URLs are not supported</p> </div>
Logout URL	Skip this field. You will enter the URL where you will be redirected after logging out of the confidential application.
Post Logout Redirect URL	Enter the following URL where you want to redirect the user after logging out of the application. <i>https://static.oracle.com/cdn/spreadsheet/red-4/redirector.html</i>
Client Type	Select Confidential . The available client types are Trusted and Confidential . Choose Trusted only if the client can generate self signed user assertions.

Table 16-3 (Cont.) Client Configuration options and their description

Option	Description
Allowed Operations	<p>Skip this field. It is optional.</p> <ul style="list-style-type: none"> – Select the Introspect check box only if you want to allow access to a token introspection end point for your application. – Select the On behalf Of check box only if you want to ensure that access privileges can be generated from the user's privileges alone. This allows the client application to access endpoints to which the user has access, even if the client application by itself would not normally have access.
ID Token Encryption Algorithm	The default is <code>none</code> .
Allowed Client IP Address	Skip this field. It is optional.
Authorized Resources	<p>Select All.</p> <p>You can select any one of the following options to allow a client application to access authorized resources:</p> <ul style="list-style-type: none"> – All – Access any resource within a domain (All). See Accessing All Resources. – Tagged – Access any resource with matching tags (Tagged). See Accessing Resources With Matching Tags. – Specific – Access only those resources where an explicit association between the client and the resource (Specific) exists. See Accessing Resources With Specific Scopes. <div data-bbox="1003 1266 1120 1297" data-label="Section-Header">Note</div> <p>The option to define an authorized resource is available to only confidential applications. Mobile applications don't have the option to define a trust scope.</p>
Resources	<p>Skip this field. It is optional.</p> <p>Only if you want your application to access APIs from other applications, then click Add in the Token Issuance Policy section of the Add Confidential Application page.</p>

Table 16-3 (Cont.) Client Configuration options and their description

Option	Description
Grant the client access to Identity Cloud Service Admin APIs	<p>Skip this field. It is optional. Click Add to enable your confidential application to access Oracle Identity Cloud Service APIs.</p> <p>In the Add App Role window, select the application roles that you want to assign to this application. This enables your application to access the REST APIs that each of the assigned application roles can access.</p>

- Click **Next** to proceed to the Configure policy tab of the **Add Confidential Application** wizard.
- On the **Add Confidential Application** wizard's **Web Tier Policy** page, click **Skip and do later**.

- Click **Finish**.
The application has been added in a deactivated state. It must be activated to function.

Record the **Client ID** and **Client Secret** that appear in the **Application Added** dialog box.

Use this ID and secret as part of your connection settings to integrate with your confidential application. The **Client ID** and **Client Secret** are equivalent to a credential (for example, an ID and password) that your application uses to communicate with Oracle Identity Cloud Service.

You have created a Confidential Application of Type Client which is Assigned the Desired Scopes.

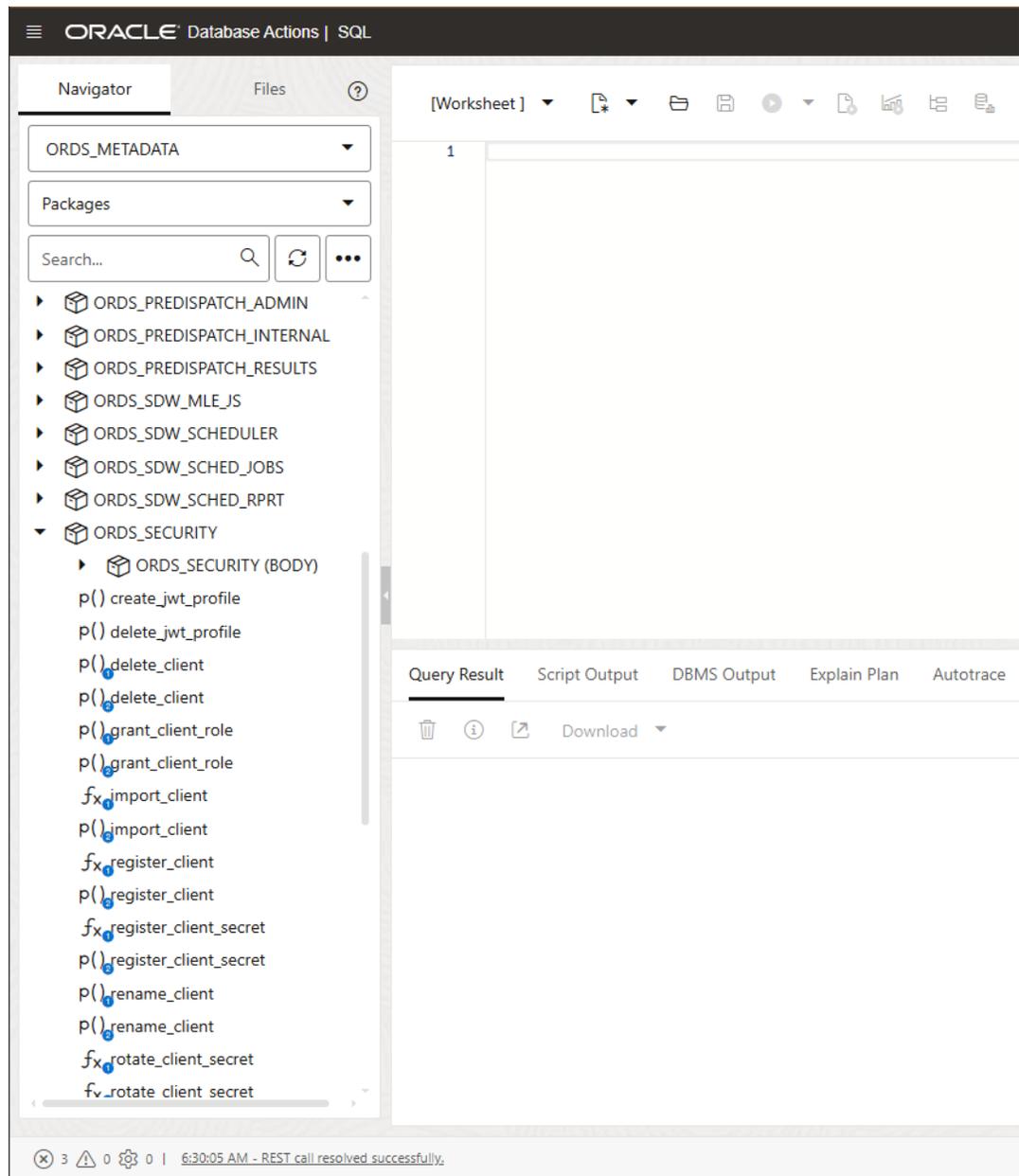
Enable IAM Login for Autonomous Database Schema

Prerequisites:

- Create an ORDS JWT profile. See <https://docs.oracle.com/en/learn/secure-ords-oci-iam/index.html#task-2-create-an-ords-jwt-profile>.
- Ensure you have ORDS version 23.3 or higher, which supports JWTs.

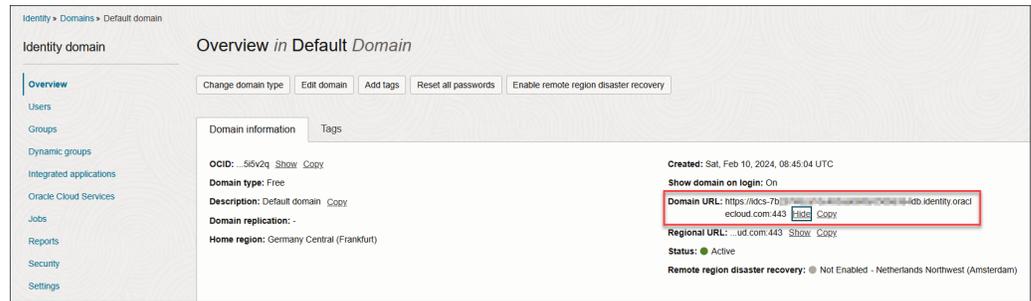
The following instructions allows ORDS to validate JWT bearer tokens and grant access to protected resources. You can set the JWT profile details, ensuring that the issuer, audience, and JWK URL are correctly configured.

- On the SQL Worksheet, from the Navigator tab, select `ORDS_METADATA` from the Schema drop-down list.
- Select `Packages` from the Object type drop-down list.
- Type `ORDS_SECURITY` in the Search field. The search function retrieves all the entries that start with `ORDS_SECURITY`.
- Expand the `ORDS_SECURITY` package.



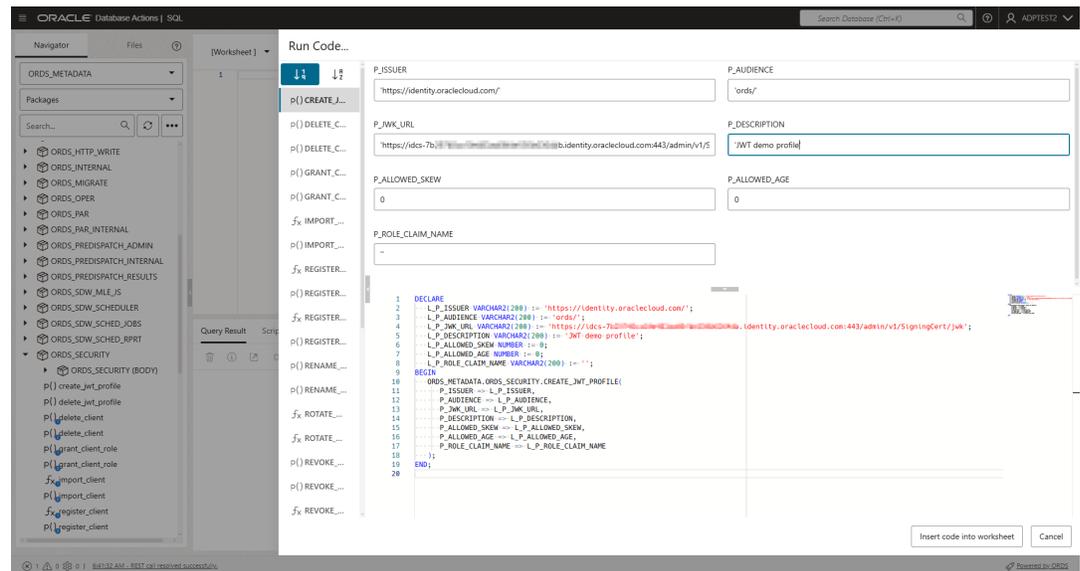
- Right click `CREATE_JWT_PROFILE` and click `RUN`. This opens a `RUN CODE` dialog. On the `Run Code...` dialog, specify the following field values:

- **P_ISSUER**- <https://identity.oraclecloud.com/>. This field must be a non-null value and must be filled within a single comma.
- **P_AUDIENCE**- *ords/*. This field must be a non-null value.
- **P_JWK_URL**- Append the *DOMAIN URL* with */admin/v1/SigningCert/jwk*. It must be a non-null value starting with *https://* and identify the public verification key provided by the authorization server in a JSON Web Key (JWK) format. You can view the **Domain URL** in the Domain information tab present in the **Domains** menu of **Identity & Security** navigation menu of the OCI console.



See: <https://docs.oracle.com/en/cloud/paas/iam-domains-rest-api/api-security-signing-certificates-jwk.html> for more details.

- **P_DESCRIPTION**- Enter the description for this profile. For example, *"JWT Demo confluence"*.
- **P_ALLOWED_AGE**- *"0"*
- **P_ALLOWED_SKEW**- *"0"*



Click **Run Script**.

Once the JWT profile is configured, end users can access ORDS protected resources by presenting JWT tokens specified in the JWT profile.

Create a connection file

1. Click on the **Add** button on the header of the Connections pane to add a connection. This opens an Add new connection dialog box.
2. Specify the following fields on the Add new connection dialog box:
 - **Connection name:** Enter the name of the connection.
 - **Autonomous Database URL:** Enter the URL of the Autonomous Database you wish to connect to. Copy the entire URL from the web UI of the Autonomous Database. For example, enter or copy the following link "https://<hostname>-<databasename>.adb.<region>.oraclecloudapps.com/" to connect to the database.
 - Select the connection type: **OCI IAM**
 - **Domain URL:** Enter the **Domain URL** from the domain information tab.
 - **Client ID:** Enter the **Client ID** you recorded from [Create a domain integrated application](#).
 - **OAuth Client ID:** Enter the **Client Secret** you recorded from [Create a domain integrated application](#).
 - **Schema:** Enter the same schema you use to [Enable IAM Login for Autonomous Database Schema](#).

After the connection is created you can share it with other users of this domain.

Role Based Access Control IAM Connections

This topic provides instructions for IT management on how to prepare a domain-integrated application that enables Role Based Access Control using JSON Web Token (JWT) custom claims. This will enable domain users to log in to the database using their domain credentials instead of the schema password.

This chapter enables domain users to authenticate to the database with their domain credentials rather than using the schema specific password.

To create domain integrated application that allows Role Based Access Control with JWT custom claims, you must:

- [Create custom attributes for users](#)
- [Assign custom role](#)
- [Create a domain Integrated application](#)
- [Define the Custom Claim](#)
 - [Create an Identity domain integrated application](#)
 - [Create new claim for the JWT token](#)
- Enable IAM Login for Autonomous Database Schema
- Create a connection file

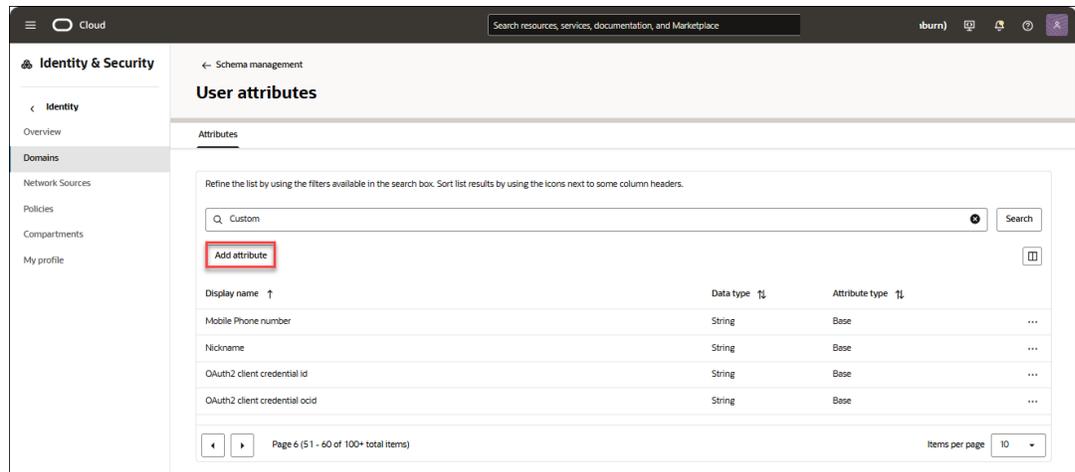
Note

We will use the `Default` domain throughout this chapter, but you can select any domain of your preference.

Create custom attributes for users

Entity attributes are properties of the entity. The information about the user entity is stored in the form of attributes, such as first name, last name, user login, and password. You can create custom attributes using OCI. Let's begin by defining a custom user attribute, which will serve to store the roles assigned to each user within the domain.

1. Navigate to: **Identity** → **Domains** → **Default**. Click the **Schema management** tab and select **User attributes**.
2. Click **Add Attribute** under **User attributes**.



3. On the Add Attributes dialog, specify the following fields:
 - **Display name:** *ORDS RBAC*
 - **Name:** *rbac_ords*
 - **Description:** *Role Based Access Control for ORDS*
 - **Data type:** *String Array*

You can leave the rest of the fields to their default state.

Add attribute

Display name
ORDS RBAC

Name
rbac_orcls

Description
Role Based Access Control for ORDS

Data type
String array

Minimum length

Maximum length

Searchable

End-user permissions
Read-Write

Cancel Add

Click **Add**. You have added the custom attribute.

- Search for the newly added attribute by typing *Custom* in the search field. You will view *ORDS RBAC* attribute in the list.

← Schema management

User attributes

Attributes

Refine the list by using the filters available in the search box. !

Q Custom

Display name ↑	Data type ↑↓	Attribute type ↑↓	
ORDS RBAC	String array	Custom	...
Organization Name	String	Base	Edit attribute
Other Address Country	String	Base	Remove

Click **Edit attribute**.

- Copy and note the value of **FQN** (Fully Qualified Name).

Edit attribute

Display name
ORDS RBAQ

Name
rbac_ords

FQN
urn:ietf:params:scim:schemas:ids:extension:custom:User:rbac_ords

Description
Role Based access control for ORDS

Data type
String array

Minimum length

Maximum length

Searchable

Administrator permissions
Read-Write

End-user permissions
Read-Write

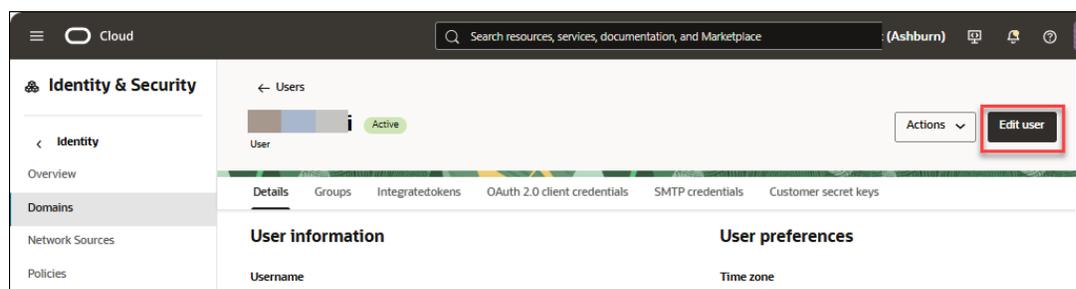
Cancel Save

In this example, **FQN** is
urn:ietf:params:scim:schemas:ids:extension:custom:User:rbac_ords.

Assign custom role

Once you have created a custom user attribute, you can proceed to assign custom roles within that attribute field.

1. Navigate to: **Identity** → **Domains** → **Default**. Click the **User Management** tab and select your username from the list of users displayed.



Click **Edit user**. Scroll down to the **Other Information** field and enter **SQL Developer, SODA Developer** under ORDS RBAC.

For more information on ORDS roles, refer to this chapter on [About Oracle REST Data Services User Roles](#).

Edit user

Country

Time zone

Preferred language

Organization name

Other information

ORDS RBAC
SQL Developer, SODA Developer

Cancel Save changes

After you have added the roles, click **Save changes**.

Create a domain Integrated application

We will develop an integrated application within the domain that will issue a JWT token upon sign-in.

1. Navigate to: **Identity** → **Domains** → **Default**. Click the **Integrated applications** tab and select **Add application**.
2. Click **Confidential Application** and select **Launch Workflow**.
3. On the **Add Confidential Application** dialog, specify the following fields:
 - **Name:** Enter the name of the confidential application. For example, *Spreadsheet-Addin RBAC*.
 - **Description:** Enter the description. For example, *Integrated Application for the spreadsheet add-in role based access control*.

Add Confidential Application

Name
Spreadsheet-Addin RBAC

Description
Integrated Application for the spreadsheet add-in role based access control

Application icon Default

URLs

Application URL

Custom sign-in URL

Custom sign-out URL

Custom error URL

Custom social linking callback URL

Display settings

Display in My Apps

Select if you want the application to be listed for users on their My Apps page.

Authentication and authorization

Define a more detailed authentication and authorization configuration.

Enforce grants as authorization

Select if you want to allow access to this app only if the user has been granted this app.

▼ **Tags**

Add tags to organize your resources. [What can I do with tagging?](#)

No items to display

Add tag

What's next
Configure OAuth and Web tier policy on the Application details page

Cancel **Submit**

Click **Submit**.

You will view the newly added **Spreadsheet-Addin RBAC** application page.

4. Click **Edit OAuth configuration** in the **OAuth configuration** tab of the **Spreadsheet-Addin RBAC** page.

← Integrated applications

Spreadsheet-Addin RBAC inactive Actions

Next steps
Configure OAuth and Web tier policy

Details **OAuth configuration** Web tier policy Users Groups Tags

Resource server configuration Resource server configuration for this application is disabled. **Edit OAuth configuration** **Client configuration** Client configuration for this application is disabled.

5. Under Resource Server Configuration, select **Configure this application as a resource server now**.
6. In **Configure application APIs that need to be OAuth protected**, select **3600** as Access token expiration (seconds).
7. In the **Primary audience** field, enter **ssaddin/**.
8. Click **Add scopes** and add **rbac Scope**.

Edit OAuth configuration

Resource server configuration

Configure this application as a resource server now
 No resource server configuration

Configure application APIs that need to be OAuth protected

Access token expiration (seconds):

Allow token refresh:

Select if you want to use the refresh token that you obtain when using the Resource Owner, Authorization Code, or Assertion grant types.

Primary audience

Primary audience:

Add secondary audience:

Enter the secondary recipients where the access token of your application is processed.

Add scopes:

Add scopes to specify which of the application's resources are available to other applications.

Scopes

Scope	Protected	Display name	Description	Requires user consent
<input type="checkbox"/> rbac	No	Role Based Access Control		No

Page 1 of 1 (1 - 1 of 1 total items) Items per page: 10

9. Under **Client Configuration**, specify the following fields:
 - Select **Configure this application as a client now**.
 - Select **Implicit** under **Allowed grant types**.
 - Enter the following value in the **Redirect URL** field: `https://static.oracle.com/cdn/spreadsheet/red-4/redirector.html`
 - Enter the following value in the **Post-logout URL** field: `https://static.oracle.com/cdn/spreadsheet/red-4/redirector.html`

Edit OAuth configuration

▼ Client configuration

Configure this application as a client now
 No client configuration

Authorization

Allowed grant types
Select the grant types that this application is allowed to use when requesting validation.

Resource owner
 Client credentials
 JWT assertion
 Refresh token
 Device code
 Authorization code
 Implicit
 SAML2 assertion
 TLS client authentication

Allow non-HTTPS URLs

Select this check box if you want to use non-HTTPS URLs for the Redirect URL, Logout URL, or Post-logout redirect URL fields. For example, if you are sending requests internally, want a non-encrypted communication, or want to be backward-compatible with OAuth 1.0, then you can use a non-HTTPS URL. Also, select this check box when you are developing or testing your application and you might not have configured SSL. This option is not recommended for production deployments.

Redirect URL

Redirect URL
 ×

Post-logout redirect URL

Post-logout redirect URL
 ×

Logout URL

Logout URL

Client type
Select the client type. Choose Trusted if the client can generate self-signed user assertions, and then import your signing certificate.

Trusted
 Confidential

Certificate

Alias

Allowed operations
Select 'Introspect' if you want to allow access to a token introspection end point for your application. Select 'On behalf of' if you want to ensure that access privileges can be generated from the user's privileges alone. Using this option means that a client application can access endpoints to which the user has access, even if the client application by itself wouldn't normally have access.

Introspect
 On behalf of

ID token encryption algorithm

Bypass consent

Turn on Bypass consent to overwrite the Require consent attribute for all the scopes configured for the application. Turning this option on means that no scope will require consent.

Client IP address
 Anywhere
 Restrict by network perimeter

Token issuance policy

Authorized resources
Choose which authorized resources a client application can access. Select 'All' to access any resource within a domain. Select 'Specific' to access only those resources where an explicit association between the client and the resource exists.

All
 Confidential

Add resources

Add resources if you want your application to access the APIs of other applications.

Add app roles

Add the application roles to assign to this application. For example, add the Identity Domain Administrator role so that all REST API tasks available to the Identity domain administrator will be available to the application.

Cancel

Click **Submit**.

Note

Ensure you **Activate** the application.

Define Custom Claim

To include user roles within the JWT, we must map the custom attribute established in the first section into the JWT using a Custom Claim.

To achieve this, we need to create an additional temporary Integrated Application and attach the custom claim through it.

Create Identity domain integrated application

You will create an additional temporary Integrated Application:

1. Navigate to: **Identity** → **Domains** → **Default**.
2. Click the **Integrated applications** tab and select **Add application**.
3. Click **Confidential application** and select **Launch workflow**.
 - Enter the name of the application: *Identity domain integrated application*
 - Enter Description: This is a domain integrated application.

Click **Submit**.

4. On the **Identity domain integrated application** page, select the **OAuth configuration** tab.
5. Click **Edit OAuth configuration** on the **Resource server configuration**. On the **Edit OAuth configuration** dialog, specify the following fields:
 - Under **Client configuration**, select the default value **Configure this application as a client now**.
 - Under **Authorization**, select **Client credentials**. Keep the rest of the configurations to its default state.
 - Select **Add app roles**.

Edit OAuth configuration

Resource server configuration

Configure this application as a resource server now

No resource server configuration

Client configuration

Configure this application as a client now

No client configuration

Authorization

Allowed grant types
Select the grant types that this application is allowed to use when requesting validation.

Resource owner

Client credentials

JWT assertion

Refresh token

Device code

Authorization code

Implicit

SAML2 assertion

TLS client authentication

Allow non-HTTPS URLs

Select this check box if you want to use non-HTTPS URLs for the Redirect URL, Logout URL, or Post-logout redirect URL fields. For example, if you are sending requests internally, want a non-encrypted communication, or want to be backward-compatible with OAuth 1.0, then you can use a non-HTTPS URL. Also, select this check box when you are developing or testing your application and you might not have configured SSL. This option is not recommended for production deployments.

Redirect URL

Redirect URL ×

Post-logout redirect URL

Post-logout redirect URL ×

Logout URL

Logout URL

Client type
Select the client type. Choose Trusted if the client can generate self-signed user assertions, and then import your signing certificate.

Trusted

Confidential

Certificate

Alias —

Allowed operations
Select 'Introspect' if you want to allow access to a token introspection endpoint for your application. Select 'On behalf of' if you want to ensure that access privileges can be generated from the user's privileges alone. Using this option means that a client application can access endpoints to which the user has access, even if the client application by itself wouldn't normally have access.

ID token encryption algorithm
None

Bypass consent

Turn on Bypass consent to overwrite the Require consent attribute for all the scopes configured for the application. Turning this option on means that no scope will require consent.

Client IP address

Anywhere

Restrict by network perimeter

Token issuance policy

Authorized resources
Choose which authorized resources a client application can access. Select 'All' to access any resource within a domain. Select 'Specific' to access only those resources where an explicit association between the client and the resource exists.

All

Confidential

Add resources

Add resources if you want your application to access the APIs of other applications.

Add app roles

Add the application roles to assign to this application. For example, add the Identity Domain Administrator role so that all REST API tasks available to the identity domain administrator will be available to the application.

- On the **Add app roles** dialog, click **Identity Domain Administrator** and select **Add**.

Add app roles

Q Identity Domain Administrator Search

Identity Domain Administrator

Signin

Me Password Validator

Security Administrator

User Administrator

User Manager

Help Desk Administrator

Application Administrator

Audit Administrator

Change Password

Page 1 of 3 (1 - 10 of 26 total items) Items per page 10

Cancel **Add**

- After adding App roles, click **Submit**.
- Activate** the application.

Note

Ensure you make a note of the **Client ID** and **Client Secret** from the **General Information** section of the Integrated Application.

Identity & Security ← Integrated applications

Identity domain integrated application Inactive Actions

Application

Next steps
Configure Web tier policy

Details OAuth configuration Web tier policy Consent information Users Groups Tags

Resource server configuration Resource server configuration for this application is disabled. Edit OAuth configuration

General Information

Client ID 127a2905e9114574ba58877a6ebfcd3

Client secret

Show secret

Regenerate Client secret available for regeneration Regenerate secret

Client configuration Client configuration for this application is enabled.

Authorization

Grant Types --

Resource owner Disabled

Client credentials Enabled

JWT assertion Disabled

Allow non-HTTPS URLs Disabled

Redirect URL --

Post-logout redirect URL --

Logout URL -

Bypass consent Disabled

ID token encryption algorithm None

Client type Confidential

Certificate -

Allowed operations -

Client IP address Restrict by network perimeter

Token issuance policy

Authorized resources Explicit

Create new claim for the JWT token

We require `CLIENT ID` , `CLIENT SECRET` and `DOMAIN URL` to retrieve a new access token for the Identity Domain Administrator Application.

We have the `CLIENT ID` and `CLIENT SECRET` from the domain integrated application you created in the previous step.

The `DOMAIN URL` can be found in the domain page under Domain information.

The screenshot shows the 'Domains' page in Oracle Identity Cloud Service. The domain 'Default' is active. The details section includes the following information:

Field	Value	Action
OCID	ocid1.domain.oc1..aaaaaaanxvvdnorkqwhohu5myn754dxijiatm3xpws6yecg2z76yhd4k4a	Copy
Domain type	Free	
Description	Default domain	Copy
Domain replication	-	
Home region	US East (Ashburn)	
Created	Dec 09, 2024, 18:22:06 UTC	
Show domain on login	On	
Domain URL	https://idcs-c276...oraclecloud.com:443	Copy
Regional URL	...s-ashburn-idcs-1.identity.us-ashburn-1.oci.oraclecloud.com:443	Copy
Status	Active	
Remote region disaster recovery	US West (Phoenix)	
Remote region disaster recovery status	Not Enabled	

Run the following command and replace the (`CLIENT ID`) , (`CLIENT SECRET`) and (`DOMAIN URL`) values with your values.

```
export ACCESS_TOKEN=$(curl -s -i -u"(CLIENT ID):(CLIENT SECRET)" -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" --request POST (DOMAIN URL)/oauth2/v1/token -d "grant_type=client_credentials&scope=urn:opc:idm:__myscopes__" | tail -n +1 | grep -o '"access_token": "[^"]*' | cut -d'"' -f4)
```

In this example you will use the following values:

- Client ID: 123a1234e1234567aa12345a1abcdefg1
- Client Secret: idcscs-12a1a123-a123-1234-1234-e1a05aabc123

- Domain URL: `https://idcs-a123ab1ab12a4bb99a9aa9ab99aabb99.identity.oraclecloud.com:443`

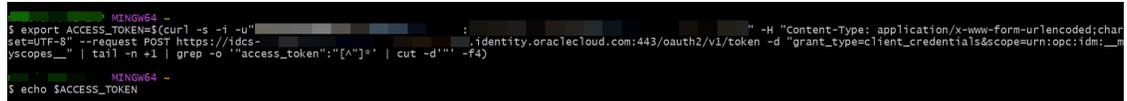
Run the following code in the Bash shell, replacing the placeholder values accordingly.:

```
export ACCESS_TOKEN=$(curl -s -i -u"123a1234e1234567aa12345a1abcdeffg1:
idscs-a123a123-a123-1234-1234-e1a05aabc123" -H "Content-Type: application/x-
www-form-urlencoded;charset=UTF-8" --request POST https://idcs-
a123ablab12a4bb99a9aa9ab99aabb99.identity.oraclecloud.com:443/oauth2/v1/token
-d "grant_type=client_credentials&scope=urn:opc:idm:__myscopes__" |
    tail -n +1 | grep -o "access_token":"[^"]*" | cut -d'"' -f4)
```

To verify that you have successfully retrieved the token, run the following command and review the displayed token.

```
echo $ACCESS_TOKEN
```

The image below displays how the above commands appear in the Bash shell.



```
MINGW64 ~
$ export ACCESS_TOKEN=$(curl -s -i -u"123a1234e1234567aa12345a1abcdeffg1:
idscs-a123a123-a123-1234-1234-e1a05aabc123" -H "Content-Type: application/x-
www-form-urlencoded;charset=UTF-8" --request POST https://idcs-
a123ablab12a4bb99a9aa9ab99aabb99.identity.oraclecloud.com:443/oauth2/v1/token
-d "grant_type=client_credentials&scope=urn:opc:idm:__myscopes__" |
    tail -n +1 | grep -o "access_token":"[^"]*" | cut -d'"' -f4)
MINGW64 ~
$ echo $ACCESS_TOKEN
```

It displays output consisting of multiple lines (over 10) containing alphanumeric characters.

Run the following command with (ROLE CLAIM NAME) as `ssaddin.role` and (MODIFIED FQN) as `urn:ietf:params:scim:schemas:idcs:extension:custom:User.rbac_ords`:

Note

Use the FQN from step 1 and replace the final ":" with a ".".

For example, the FQN from step 1 is:

```
urn:ietf:params:scim:schemas:idcs:extension:custom:User.rbac_ords
```

So the MODIFIED FQN should be:

```
urn:ietf:params:scim:schemas:idcs:extension:custom:User.rbac_ords
```

```
curl -i -X POST (DOMAIN URL)/admin/v1/CustomClaims -H"Cache-Control: no-
cache" -H"Accept:application/json" -H"Content-Type:application/json" -
H"Authorization: Bearer $ACCESS_TOKEN" -d '{
  "schemas": [
    "urn:ietf:params:scim:schemas:oracle:idcs:CustomClaim"
  ],
  "name": "(ROLE CLAIM NAME)",
```

```
"value": "$user.(MODIFIED FQN).*",  
"expression": true,  
"mode": "always",  
"tokenType": "AT",  
"allScopes": false,  
"scopes": [  
  "ssaddin/rbac"  
]  
'
```

Note

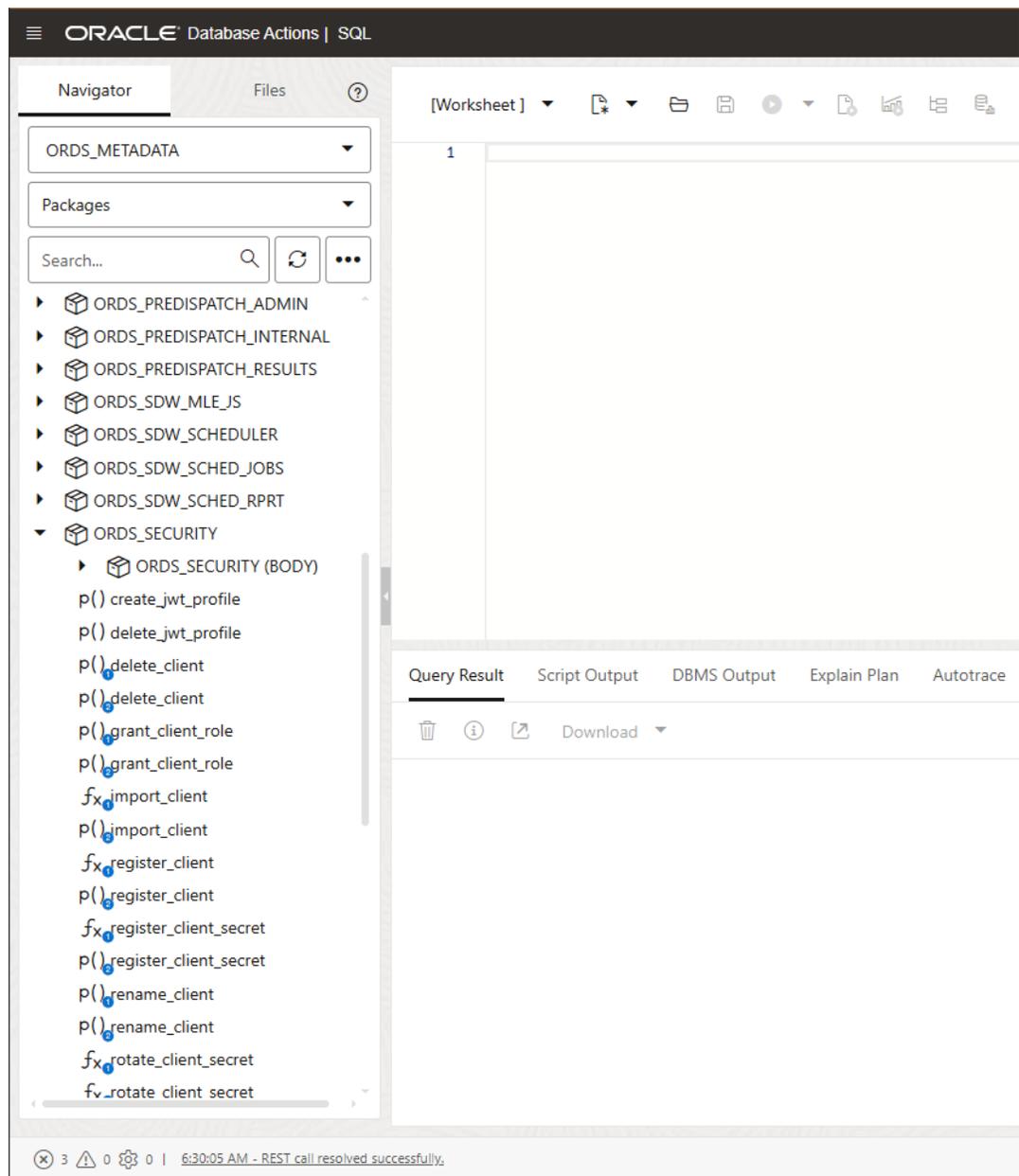
Replace the `Domain URL` with its actual value in the code above.

You will view the following as output: `HTTP/1.1 201 Created`.

Enable RBAC IAM Login for Autonomous Database Schema

The following instructions allow ORDS to validate JWT bearer tokens and grant access to protected resources.

1. On the SQL Worksheet, from the Navigator tab, select `ORDS_METADATA` from the Schema drop-down list.
2. Select `Packages` from the Object type drop-down list.
3. Type `ORDS_SECURITY` in the Search field. The search function retrieves all the entries that start with `ORDS_SECURITY`.
4. Expand the `ORDS_SECURITY` package.



5. Right click `CREATE_JWT_PROFILE` and click `RUN`. This opens a `RUN CODE` dialog. On the `Run Code...` dialog, specify the following field values:
 -
 - **P_ISSUER**- `https://identity.oraclecloud.com/`. This field must be a non-null value and must be filled within a single comma.
 - **P_AUDIENCE**-`ssaddin/`. This field must be a non-null value.
 - **P_JWK_URL**- Append the `DOMAIN URL` with `/admin/v1/SigningCert/jwk`. It must be a non-null value starting with `https://` and identify the public verification key provided by the authorization server in a JSON Web Key (JWK) format. You can view the **Domain URL** in the Domain information tab present in the **Domains** menu of **Identity & Security** navigation menu of the OCI console.
 - **P_DESCRIPTION**- Enter the description for this profile. For example, `"RBAC JWT Demo confluence"`.

- **P_ALLOWED_AGE**-"0"
- **P_ALLOWED_SKEW**-"0"
- **P_ROLE_CLAIM_NAME**- "ssaddin.role"

Click **Insert code into worksheet**.

Run the procedure.

You will view "PL/SQL procedure successfully completed." in the output panel.

Create a connection file

1. Click on the **Add** button on the header of the Connections pane to add a connection. This opens an Add new connection dialog box.
2. Specify the following fields on the Add new connection dialog box:
 - **Connection name:** Enter the name of the connection.
 - **Autonomous Database URL:** Enter the URL of the Autonomous Database you wish to connect to. Copy the entire URL from the web UI of the Autonomous Database. For example, enter or copy the following link "https://<hostname>-<datasenname>.adb.<region>.oraclecloudapps.com/" to connect to the database.
 - **Schema Name:** Enter the same schema you use to [Enable IAM Login for Autonomous Database Schema](#).
 - Select the connection type: **OCI IAM**
 - **Domain URL:** Enter the **Domain URL** from the domain information tab.
 - Select **RBAC IAM** type.

- **IAM Scope:** ssaddin/rbac

The screenshot shows the 'Edit Connection' dialog box in Oracle Cloud. The dialog is titled 'Connections' and 'Edit Connection'. It contains several fields:

- Connection Name:** RBAC Connection
- Autonomous Database URL:** https://g76d4bc07617374-t0rpt8cc5mixvhd8.adb.us-ashburn-1.oracleclou
- Schema Name:** .
- OAuth Client ID:** ffdbff4ba1ac488c96c7afe52ff8615e
- Connection type:** OCI IAM
- Domain URL:** https://idcs-c276ef6ec34f4bb99f8eb6df86aebcb8.identity.oraclecloud.com
- Select IAM type:** RBAC (selected), SBAC (unselected). Below this, it says: Access control based on the roles defined on a custom claim (Role Based Access Control)
- IAM scope:** ssaddin/rbac

At the bottom, there are 'Cancel' and 'Save' buttons.

After the connection is created you can share it with other users of this domain.

Oracle Autonomous Database for Excel

The Oracle Autonomous Database for Excel integrates Microsoft Excel spreadsheets with the Autonomous Database to retrieve and analyze data from Analytic Views in the database. You can also directly run SQL queries to view their results in the worksheet.

Note

- The **Download Microsoft Excel/Google Sheets add-in** is available to you under the **Downloads** menu of your Database Actions instance only if you have the `ADPUSER` role.
- You require `ADMIN` privileges to grant roles.
- To grant yourself an `ADPUSER` role, you must log into your Database Actions instance and enter the following command in the SQL worksheet area displayed in the **SQL** tab under **Development** tools present in the Launchpad:

```
Grant ADPUSER to USER;
```

- Use the Microsoft Edge browser for the best compatibility and performance.

- [Install the add-in from Microsoft AppSource](#)
The **Oracle Autonomous Database for Excel** is available in **Microsoft AppSource**.
- [Install the add-in from Database Actions](#)
You can install the add-in from Database Actions. This is accessed by users with Oracle Account.
- [Uninstall the add-in](#)
The following section describes the steps to uninstall the Oracle Autonomous Database add-in.
- [Using Oracle Autonomous Database for Excel](#)
After you install the add-in, a new ribbon tab, **Oracle Autonomous Database** appears in MS Excel.
- [Connection management](#)
Each time you start the add-in for Excel, you must create a connection.
- [Log Settings](#)
The Settings icon enables you to set the log levels using the Excel add-in.
- [Run Direct SQL queries in an Excel worksheet](#)
The Oracle Autonomous Database for Excel lets you run Direct SQL queries to work with your data in an Excel worksheet.
- [Read and Access Data Using Table Hyperlinks in Excel Sheet](#)
Table Hyperlinks in Oracle Autonomous Database provide secure, preauthenticated URLs that allow read-only access to data stored in tables, views, or the result of SQL queries.
- [Reporting and Analysis in Excel](#)
You can view Reports and Analytic Views or visualizing data for analysis purpose using the **Analysis and Reports** menu.
- [Query an Analytic View in an Excel worksheet](#)
The Query Wizard menu enables you to query an Analytic View and retrieve the results in an Excel Worksheet. Once the wizard retrieves the data, it becomes local to Excel. You can further edit the data in Excel but not write back to the Autonomous AI Database.
- [Data Analysis in Excel Sheet](#)
The Data Analysis tool enables you to analyze data in the Autonomous Database by running SQL queries or querying an Analytical View using an intuitive drag-and-drop interface. You can also write custom queries to be run. You now can dynamically apply filters to the result set retrieved using the new faceted search capability on specific columns.

- [Natural Language in Excel Sheet](#)
You can use Natural Language Query to query the Oracle Autonomous AI Database using the **Natural Language** menu in the **Oracle Autonomous Database for Excel**.
- [FAQs for Troubleshooting errors with Excel Add-in](#)
If you experience issues with Oracle Autonomous Database for Excel, refer to frequently asked questions in this section to identify and resolve issues.
- [Oracle Autonomous Database for Excel Privacy Policy Details](#)
This topic covers details for writing policies to control access to Autonomous Database resources.

Install the add-in from Microsoft AppSource

The **Oracle Autonomous Database for Excel** is available in **Microsoft AppSource**.

Prerequisites

- You must have an active Microsoft account or an account with access to Microsoft 365.
- You can access the web version of Excel or the (Licensed version) from desktop.
- You must verify within your organization whether external apps from AppSource are permitted to be installed in your Microsoft 365 accounts. If not, you can [Install the add-in from Database Actions](#).

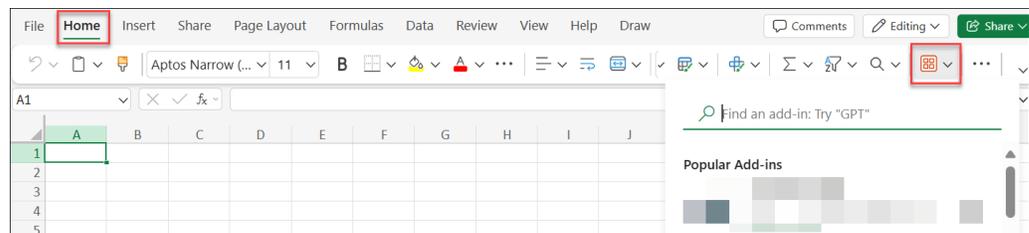
Follow these steps to install the add-in:

1. Start Excel and open a new or existing workbook:

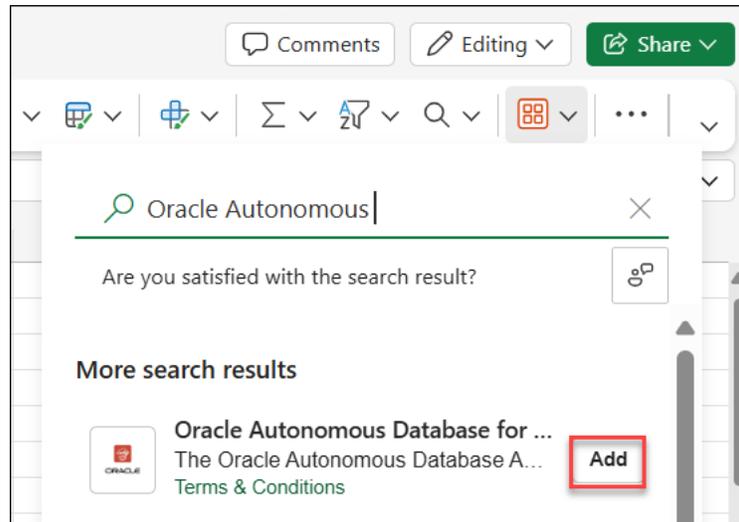
- You can access the web version of Microsoft Excel with your Microsoft AppSource account:
 - Go to your Microsoft 365 Copilot (earlier known as MS Office) app link: <https://m365.cloud.microsoft>
 - Login with your account.
 - Select Excel Workbook from the Microsoft 365 Copilot home page.

2. Install the Add-in from Office Add-ins:

- From the Home menu, go to **Add-ins** and search for "*Oracle Autonomous*" in the search bar.



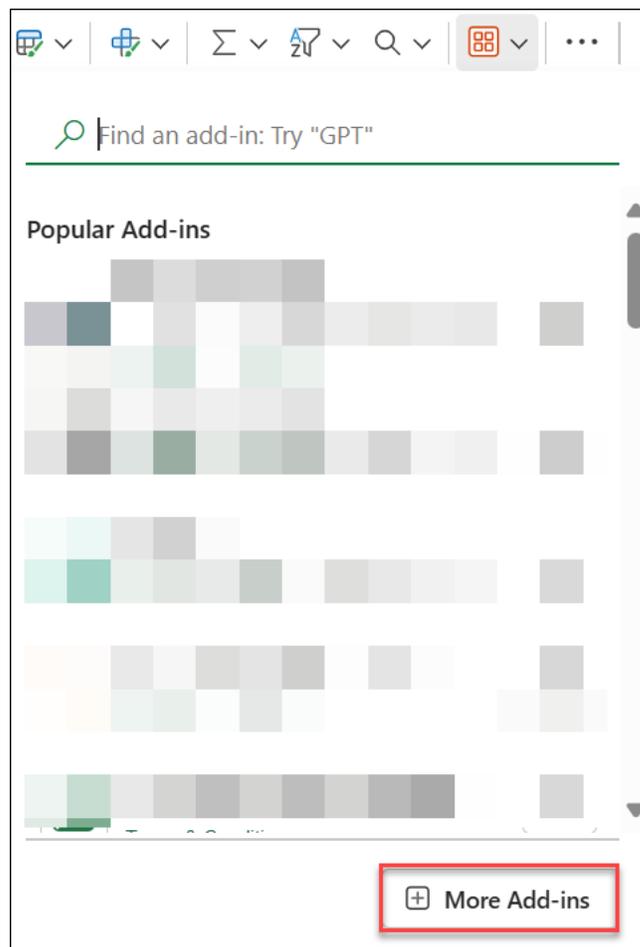
- You can view the Oracle Autonomous Database for Excel in the search results.



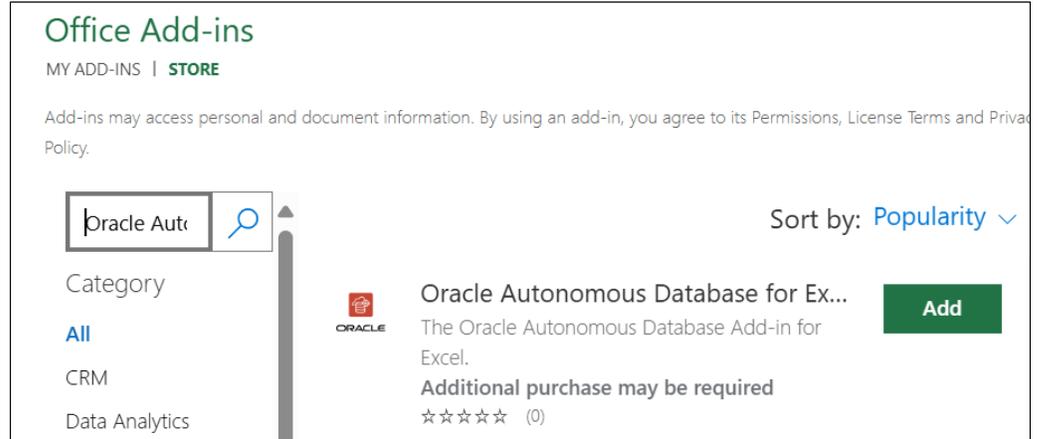
- Click **Add** to install the add-in.

OR

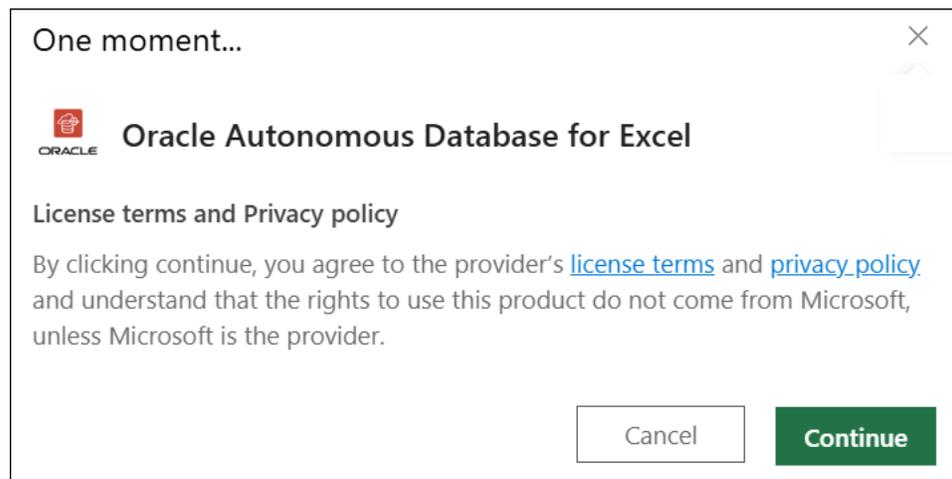
- You can alternatively click the **Add-ins** icon from the **Home** menu and select **+ More Add-ins** from the bottom of the Add-ins drop-down list.



- Enter "Oracle Autonomous" in the search field of the **STORE** tab of **Office Add-ins** and click the **search** icon.

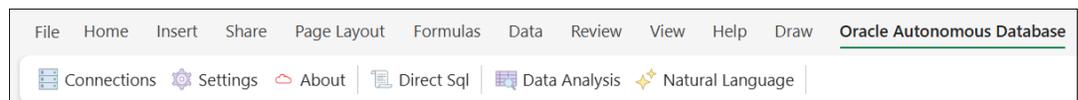


- You can view the Oracle Autonomous Database for Excel in the search results.
- Click **Add** to install the add-in in Excel.
- You will view a dialog to accept Oracle's [license terms](#) and [Oracle Autonomous Database for Excel Privacy Policy Details](#). Click **Continue** to agree to it.



You will view a message confirming the add-in loaded successfully.

A new **Oracle Autonomous Database** ribbon tab appears in MS Excel.



After you have installed the add-in, you can now connect to the Autonomous Database to analyze and query the data. Refer to the [Connection management](#) chapter for more details on this topic.

Refer to the [Uninstall the add-in](#) chapter to uninstall the add-in you installed from Microsoft AppSource.

Install the add-in from Database Actions

You can install the add-in from Database Actions. This is accessed by users with Oracle Account.

Prerequisites:

- You must have an Oracle Cloud Account.
- You can install the add-in only from the licensed desktop version of Microsoft Excel.

The following instructions outline how to install the add-in depending on your operating system, specifically for Mac and Windows.

- [Install the add-in on Mac](#)
- [Install the add-in on Windows](#)

- [Install the add-in on Mac](#)

The Oracle Autonomous Database for Excel is supported on Mac OS running Microsoft Office 365.

- [Install the add-in on Windows](#)

The Oracle Autonomous Database for Excel is supported on Windows 10 and Windows 11 operating systems running Microsoft Excel 365.

Install the add-in on Mac

The Oracle Autonomous Database for Excel is supported on Mac OS running Microsoft Office 365.

To install the Autonomous Database for Excel, run the installer script file from your Autonomous Database instance by following the steps below:

- Open the Database Actions Launchpad.
- On the **Downloads** tab of the Database Actions page, click the **Download Microsoft Excel/Google Sheets Add-in** pane.

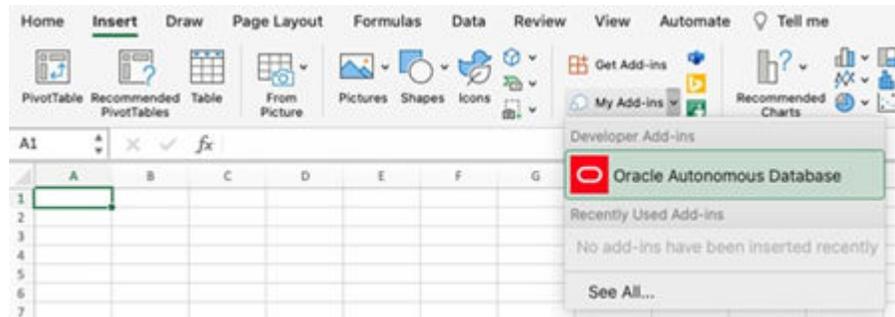


- Click **Download**.
- Click the Microsoft Excel tab and select the **Download Add-in** button to download the **oracleplugin.zip** file.
- You can now view the zip file in the Downloads folder.
- Create a new folder named **Add-in** on your system.
- Extract the contents of the zip file in the **Add-in** folder.

Follow these steps to install the add-in.

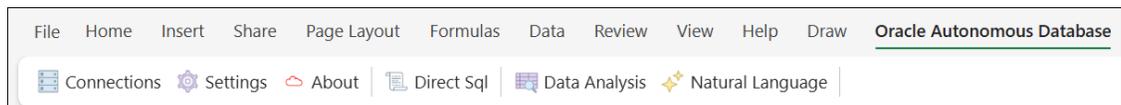
1. Quit Excel before you run the installer.

2. Navigate to the *install.sh* file in the **Add-in** folder.
3. Right-click *install.sh* and select the following options as shown : **Open With -> Other... -> Enable: All Applications ->Utilities->Terminal**
4. On completion, close the Terminal window.
5. Start Excel and open a new or existing workbook.
6. From the **Insert** menu in the Excel ribbon, select the drop-down menu of **My Add-ins**. A new **Oracle Autonomous Database** entry appears under the **Developer Add-Ins** dialog box.



7. Select **Oracle Autonomous Database**.

A new **Oracle Autonomous Database** ribbon tab appears in MS Excel.



Install the add-in on Windows

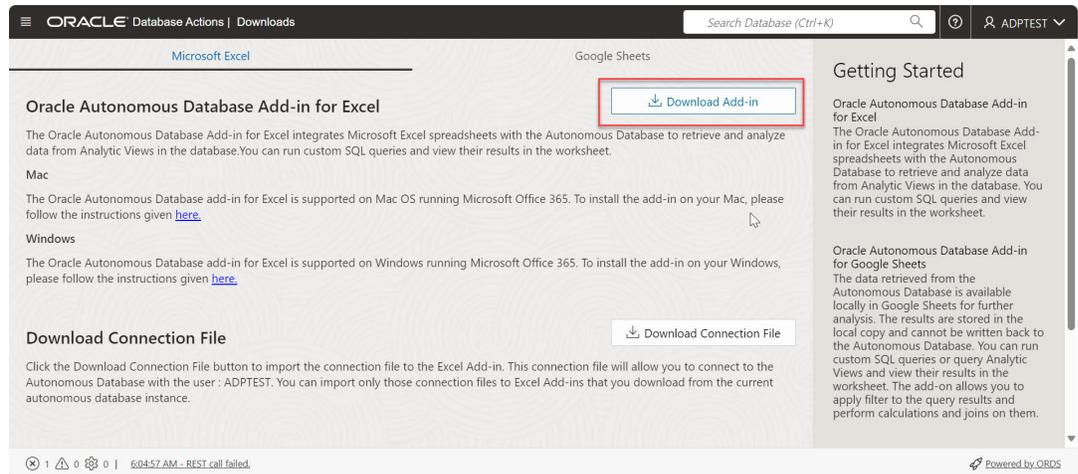
The Oracle Autonomous Database for Excel is supported on Windows 10 and Windows 11 operating systems running Microsoft Excel 365.

To install the Oracle Autonomous Database for Excel, download the *oracleplugin.zip* file and extract it to get the *install.cmd* script file from your Database Actions instance.

- Open the Database Actions Launchpad.
- On the **Downloads** tab of the Database Actions page, click the **Download Microsoft Excel/Google Sheets Add-in** pane.



- Click **Download**.
- Click the **Download Add-in** icon in the Microsoft Excel tab to download the Oracle Autonomous Database Add-in for Excel.



- Extract the *oracleplugin.zip* file to a new folder in the Downloads of your system. The extracted folder consists of an installer (install.cmd file), a manifest.xml file and a readme.txt file.

Follow these steps to install the add-in.

1. Quit Excel before you run the installer.
2. Right-click the *install.cmd* file that you downloaded.

Note

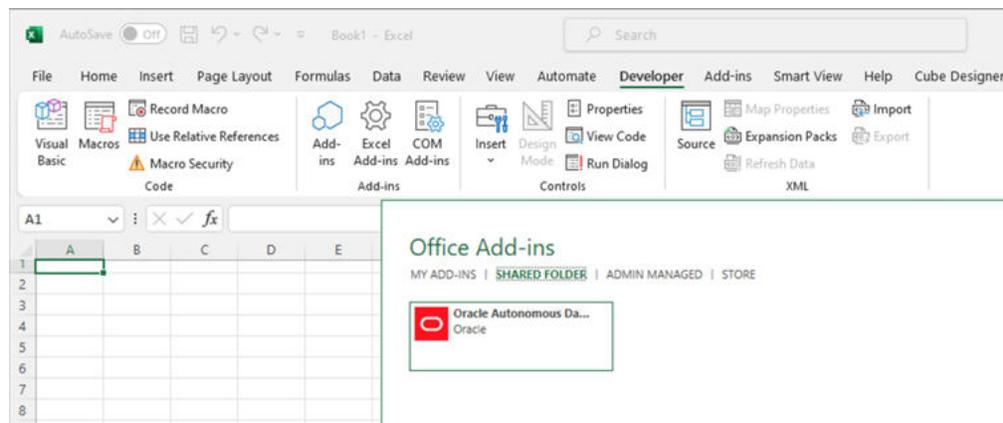
After running the installer on Windows, the add-in automatically creates a network share and adds the shared location as a trusted catalog location for Office add-ins. A catalog is used to store the manifest for the Excel Add-in. It enables publishing and management of the Excel add-in as well as other add-ins that are available in the Office Store and licensed for corporate use. You can acquire the Excel add-in by specifying the shared manifest folder as a trusted catalog.

3. Select **Run as administrator**.

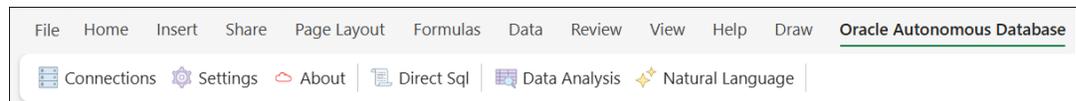
Note

You must have Administrator privileges to install the Excel add-in for Oracle Autonomous Database successfully.

4. Start Excel and open a new or existing workbook.
5. From the **Developer** menu in the Excel ribbon, click **Add-ins**, select the **SHARED FOLDER** tab on the pop-up window and select **Oracle Autonomous Database**.



- After you install the add-in, a new **Oracle Autonomous Database** ribbon tab appears in MS Excel.



Note

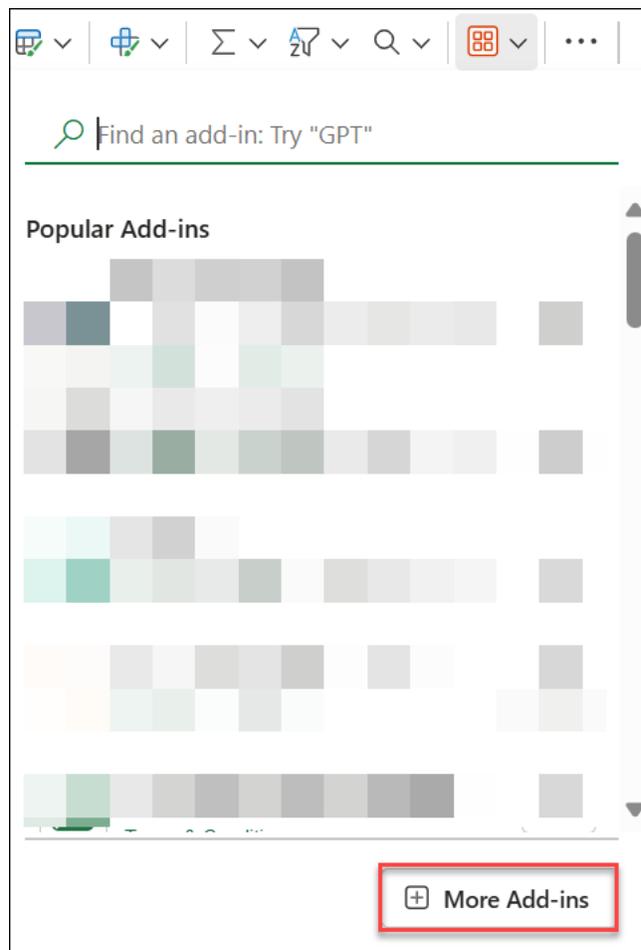
You can re-run the installer after the initial installation. Re-run the installer and choose the option of your preference. You can repair your existing installation by deleting it, selecting the installed trusted catalog or adding another manifest to the working installation.

Uninstall the add-in

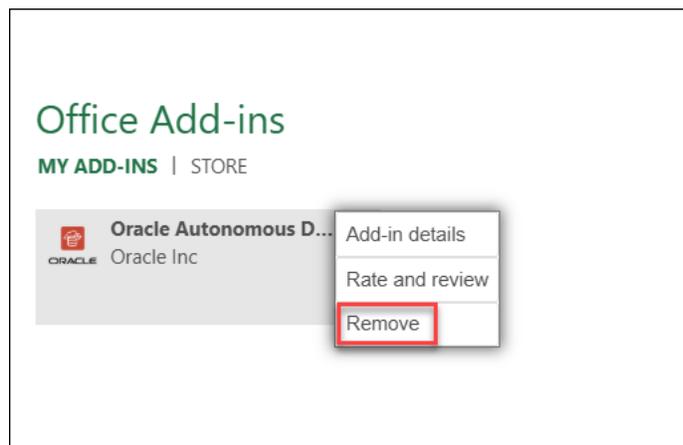
The following section describes the steps to uninstall the Oracle Autonomous Database add-in.

If you have installed **Oracle Autonomous Database for Excel** for Windows using the Microsoft AppSource, you can uninstall the add-in by following these steps:

- Click the **Add-ins** icon from the **Home** menu and select **+ More Add-ins** from the bottom of the Add-ins drop-down list.

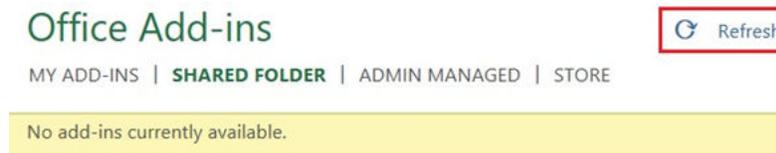


- Click **MY ADD-INS** from **Office Add-ins** and right-click the **Oracle Autonomous Database for Excel** add-in.
- Click **Remove** to uninstall the add-in from Excel.



If you have installed **Oracle Autonomous Database for Excel** for Windows using Database Actions, you can uninstall the add-in by following these steps:

- Delete the *manifest.xml* file from the folder located in `%LOCALAPPDATA%\Oracle\Autonomous Database\manifest`.
- Click **Refresh** in the Office Add-ins window to remove the Autonomous Database tab from MS Excel.



No add-ins will now be available from the Shared folder of the Office Add-ins window.

Note

After uninstalling the Add-in, if you re-install it from a different Autonomous Database (ADB), the add-in attempts to load the old ADB. You must check if the shared manifest folder's location (share path) points to the correct location. For more details, refer to *Configuring the Excel Trusted Add-in Catalog* in [FAQs for Troubleshooting errors with Excel Add-in](#).

Uninstall the add-in for Mac

To uninstall the Oracle Autonomous Database for Excel for Mac:

- Enter the following command in the terminal to remove the *manifest.xml* file.

```
rm ~/Library/Containers/com.microsoft.Excel/Data/Documents/wef/manifest.xml
```

The Oracle Autonomous Database for Excel is uninstalled from Mac.

Using Oracle Autonomous Database for Excel

After you install the add-in, a new ribbon tab, **Oracle Autonomous Database** appears in MS Excel.

You can connect to multiple Autonomous Databases, work with Analytic Views, tables, and Views, and view the data in the worksheet.

This ribbon provides buttons that let you connect to the Autonomous Database.

Click **Connections** to connect to an Autonomous Database. You must Refer to the *Connection management* for more details.

Click **Settings** to view the logging level settings of the Excel Add-in. You can also clear the logs or export the log files by copying the logging information to the clipboard.

Click **About** to view the Add-in and the supported Excel versions. The About window also displays whether the spreadsheet is connected to the database. It also shows version information for the database and Oracle Rest Data Services.

Click **Direct SQL** to run custom SQL queries.

Click **Table Hyperlinks** to read and access tabular data using table hyperlinks.

Click **Data Analysis** to query an existing Analytic View and run queries.

Click **Natural Language** to run Natural Language Queries on Tables.

Connection management

Each time you start the add-in for Excel, you must create a connection.

The connections feature lets you manage and connect to multiple Autonomous Databases with a single add-in. Multiple connections can be created. However, only one connection can remain active.

The connection panel lets you connect to the Autonomous Database through a connection where you provide the login credentials and access the Autonomous Database.

Within the Connections panel, you can:

- Create or delete multiple connections using a single add-in.
- Share connection information by exporting and importing connection information to a file.
- View existing connections.

Selecting **Connections** opens the Connections wizard.

Note

This is an implicit type of connection. Refer to [Authenticate using Implicit connection](#) to understand more about implicit connection.

The Connections wizard consists of **Manage Connections** icon that has the following options available:

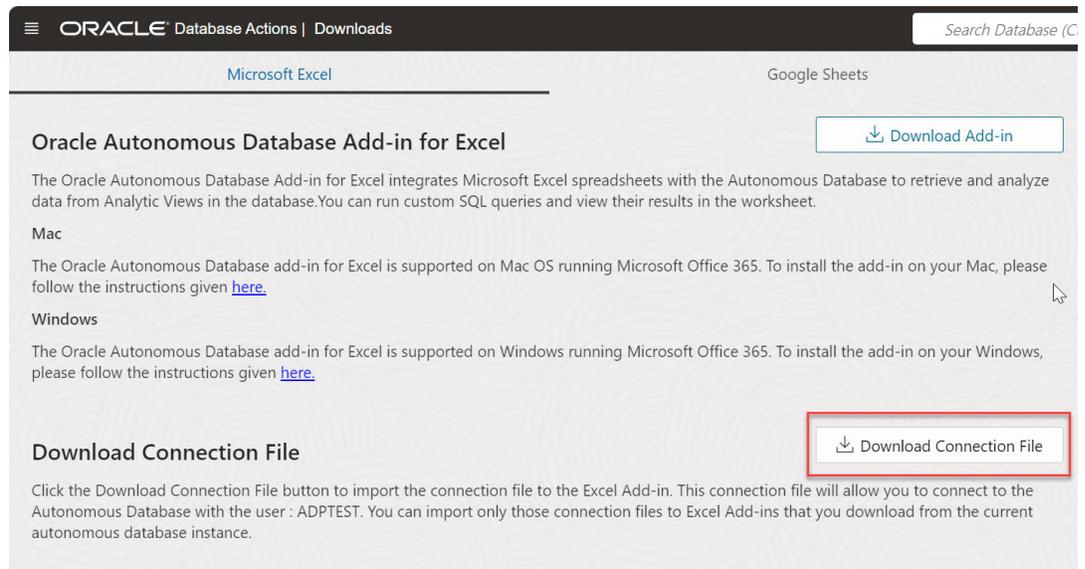
- **Add Connection:** Select **Add Connection** to Add a Connection. Refer to [Add a connection](#) section for more details.
- **Export Connection:** Select **Export Connection** to export connections. Refer to the [Share a connection](#) section for more details.
- **Import Connection:** Select **Import Connection** to launch the import wizard to choose a connection file. These files are in JSON format. Refer to the [Share a connection](#) section for more details.
- [Import a Connection](#)
You can import a connection file that you can download from the Database Actions launchpad. This file is in JSON format.
- [Add a connection](#)
You can manually create a connection to an Autonomous Database.
- [Share a connection](#)
You can import or export a connection using the **Import Connection** and **Export Connection** menus from the **Manage Connections** drop-down field on the Connections panel.

Import a Connection

You can import a connection file that you can download from the Database Actions launchpad. This file is in JSON format.

Follow the below steps from the Database Actions instance, to download the connection file.

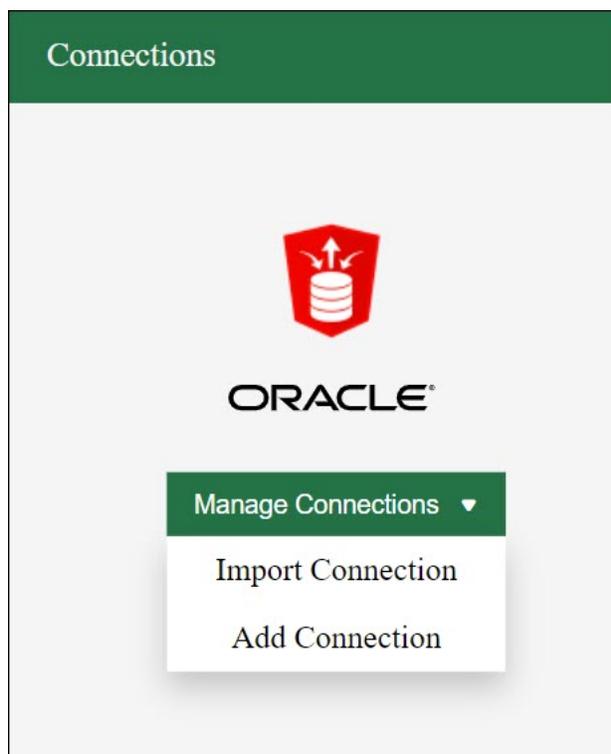
1. From the Downloads page of the Database Actions instance, click the **Download Connection File** icon to download the connection file in your system.



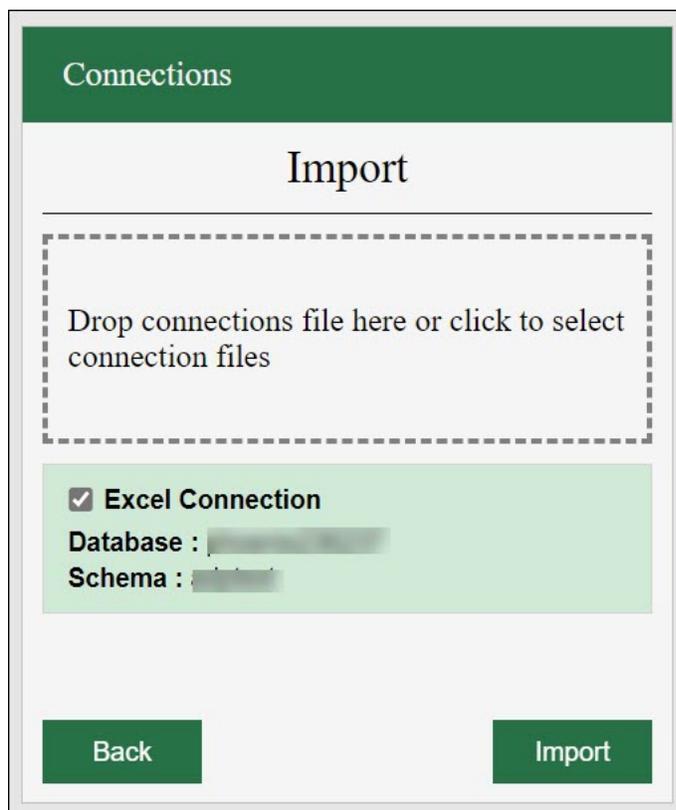
2. The file is downloaded onto your system.

To import a connection:

1. Select **Connections** from the Autonomous Database menu in the Excel sheet. This opens the Connections wizard that consists of a **Manage Connections** icon.
2. Click **Import Connection** to import the connection file you downloaded from the Database Actions instance.



3. Click and drop the connection file from your system to the drop area of the wizard. After the connection file loads, select the check box beside the connection file you want to import from the file.



4. Click **Import** to proceed.
5. Click the three vertical dots beside the connection file and select **Connect**.

Note

If you view a red icon beside the connection file, the connection is inactive or incorrect. Click the three vertical dots beside the connection file and select **Edit** to update the connection file. Ensure the Autonomous Database URL is correct and click **Save**. An example of a correct URL is "https://<hostname>-<databasesname>.adb.<region>.oraclecloudapps.com/"

6. Specify the schema name in the username field and the corresponding password in the credentials screen you view. Click **Sign in** to log in to the autonomous database.

Once signed in, you will view the connection highlighted in green. This means the connection is Active.

Add a connection

You can manually create a connection to an Autonomous Database.

Warning

This feature of adding connection manually is not recommended for this release.

Please use the [Import a Connection](#) feature to import a connection file from the Downloads menu of the Database Actions launchpad.

Share a connection

You can import or export a connection using the **Import Connection** and **Export Connection** menus from the **Manage Connections** drop-down field on the Connections panel.

- **Import Connection:** Click **Import Connection** from the **Manage Connections** drop-down icon and select a connection file from your local device. Once you import the connection file, you can view the connection in the Connection panel with a check box beside it. Select the check box and click **Import** to add the connection to the list of connections in the panel. The add-in copies the connection information you can use as a new connection. With the import feature, you do not have to enter the connection information to add a new connection.
- **Export Connection:** The export connection button exports an existing connection which you can import later. Clicking **Export Connection** opens a check box beside each connection in the connections list. You can select the connection you wish to export. Multiple selections are allowed. After you select the connection, click **Export**. Once the connection file is exported, you can view that the add-in for Excel downloads the connection file (*.json file) to your local device. The exported connection file is named spreadsheet_addin_connections.json.

Log Settings

The Settings icon enables you to set the log levels using the Excel add-in.

You can export or save the log. The log can be sent to the support team for diagnosing errors. The Log helps you troubleshoot errors. This field allows you to set the time limit for logging information that is stored in the log file. This setting automatically clears log information older than the entry made in this field. The default is 72 hrs. If the records are older than 72 hours, they will be deleted automatically.

Use Log Levels

Each log message is associated with a severity level. The level gives a rough guide to the importance and urgency of a log message. The log levels are specified with the following values ranging from one having lowest impact (left-most) to the one having highest impact (right-most):

STOP, ERROR, WARNING, INFO, VERBOSE

The following table lists the severity levels of log messages:

Log Level	Description
STOP	Records no information.
ERROR	Records events that may require corrective action, for example, a component is unavailable.
WARNING	Records issues that may lead to an error or require corrective action in the future. This is the default log level.
INFO	Records completed actions or the current state of a component, for example, the component is initializing.
VERBOSE	Displays detailed processing information along with errors, warnings and info types of severity levels.

Each logging level is cumulative, that is, each level contains all the data generated by the higher levels. For example, Warning logs contain all the data generated by the Error logs, plus the events that are specific to the Error category.

Log message format

The Log Settings writes a message to the specified log file consisting of a module, Timestamp, Severity, Server Name, Connection, Thread ID or User ID or Transaction ID, Message ID, and

the Message, along with a stack trace if any. Each attribute is contained between angle brackets.

The following is an example of a message in the server log file (split for readability; in practice, the message might be on one line):

```
{ "module": "ServerHelper", "logMessage": "connection not is
  avialble", "timestamp": 1688901058020, "level": 3, "payload":
{ "conAlias": "Excel
  Connection", "status": 400 }, "id": 161, "timeString": "9 Jul 2023
  16:40:58" }, { "module": "ServerHelper", "logMessage": "connection not is
  avialble", "timestamp": 1688901059724, "level": 3 }
```

Hours to keep a log

This field has the automatic cleaning operation. The log settings automatically clears old logs. If the logs are older than 72 hours which is the default value, they will be deleted automatically.

Clean Log

This button deletes all the logs.

Export Log

The Export Log button exports or saves logs to a file. Selecting the Export Log icon in Windows enables the add-in in downloading the exported log to "Downloads" folder.

Selecting the Export Log icon in Mac enables the add-in in copying the log information to clipboard. You can paste it to a file and save the log. The log you export will be in JSON format. The JSON file might contain sensitive information like your password, date of birth or any other data you want to secure.

Run Direct SQL queries in an Excel worksheet

The Oracle Autonomous Database for Excel lets you run Direct SQL queries to work with your data in an Excel worksheet.

With the add-in, you can create a table and insert, update and delete rows from the existing tables or views. You can view the results in the current worksheet or different worksheets.

The following image shows your data retrieved from the Autonomous AI Database and displayed in the worksheet. The Query Info section comprising the Timestamp, User name and SQL Query are shown in Excel. You can edit custom queries and run them. The worksheet displays the results of queries from the retrieved data in tabular format.

The add-in maintains a live connection with the database. However, the data retrieved is local to Excel. In case of inactivity, the connection times out, and you must log in again. You can change the active connection from the connections panel. The image shows the results from a single query, but you can insert many queries in a single workbook.

The screenshot displays an Excel spreadsheet with the following data:

Query Info	ADB URL	User	SQL-Query	Timestamp
	https://	adptest	SELECT	2024/08/20 - 14:33:53

Result Data	eeid	full_name	job_title	department	business_unit	gender	ethnicity
	E01019	Dominic Scott	Sr. Analyst	Sales	Corporate	Male	Caucasian
	E04335	Parker Allen	Sr. Analyst	Sales	Speciality Products	Male	Caucasian
	E00099	Brooklyn Salazar	IT Systems Architect	IT	Manufacturing	Female	Latino
	E03830	Madison Her	Technical Architect	IT	Speciality Products	Female	Asian
	E04242	Alice Lopez	Test Engineer	Engineering	Speciality Products	Female	Latino
	E00144	Theodore Ngo	Controls Engineer	Engineering	Research & Development	Male	Asian
	E03332	Ruby Sun	Cloud Infrastructure Architect	IT	Manufacturing	Female	Asian
	E00965	Jacob Khan	Computer Systems Manager	IT	Speciality Products	Male	Asian
	E00436	Lincoln Reyes	Computer Systems Manager	IT	Manufacturing	Male	Latino
	E03430	Leo Herrera	Sr. Business Partner	Human Resources	Research & Development	Male	Latino
	E04538	Adeline Yang	Cloud Infrastructure Architect	IT	Corporate	Female	Asian
	E03890	Everett Khan	Solutions Architect	IT	Manufacturing	Male	Asian
	E01242	Emery Doan	Controls Engineer	Engineering	Corporate	Female	Asian
	E04387	Everleigh White	Network Architect	IT	Speciality Products	Female	Caucasian
	E00276	Kennedy Romero	Engineering Manager	Engineering	Research & Development	Female	Latino

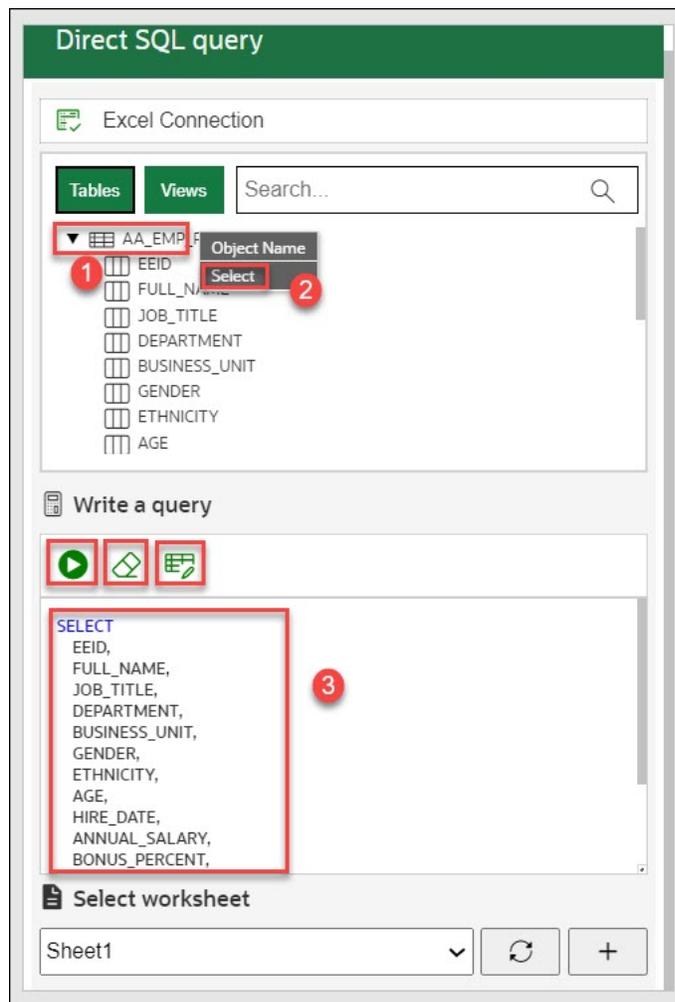
The Oracle Autonomous Database dialog box is open, showing the following details:

- Excel Connection:** Tables, Views, Search...
- Write a query:**

```
SELECT
  eeid,
  full_name,
  job_title,
  department,
  business_unit,
  gender,
  ethnicity,
  age,
  hire_date,
  annual_salary,
  bonus_percent
```
- Select worksheet:** Sheet1

To run a query using the add-in, run Excel and create a blank workbook using the standard Excel workbook file format.

1. In the Excel ribbon, select the **Autonomous Database**.
2. Click the **Direct SQL** icon from the ribbon. This opens an **Oracle Autonomous Database** dialog box in the Excel Task Pane with **Tables** and **Views** icons and a search field beside them.



3. Select **Table** to view all the existing tables in the schema. Click **Views** to see the current views in the schema.
4. You can right-click the table whose data you want to query and choose **Select** to view all the table's columns. The column names will be displayed in the Write a Query section. You can click on the table and view individual columns as well. Click the **Run** button to run the SQL query in the query editor. The query results will be displayed in the worksheet you select.

Note

You will view an error message if you click the **Run** icon while the query editor is empty.

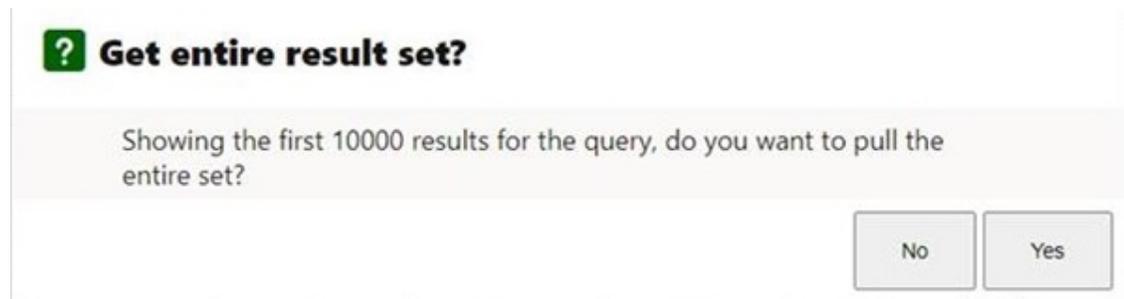
5. You can click + sign beside the **Select worksheet** drop-down to display the results in a new worksheet.
6. The worksheet also displays the timestamp, the user who creates and runs the query and the autonomous database URL.

To run another query follow these steps:

1. Click the eraser icon to clear the previous query from the SQL editor and write a new query.

2. Click **Retrieve query from Sheet** to import the SQL query from the existing worksheet and generates that query in the SQL editor.
3. In the **Select worksheet** drop-down, select a new sheet, *Sheet 2*, in this case. The Add-in adds a sheet for the user. If you work on the same sheet, the Add-in refreshes the data in the existing worksheet.
4. Click the **Run** icon to display the query results.

The worksheet displays the result of the query at a go. While this behavior works for most scenarios, sometimes, for large data sets, the query result might exceed 10K rows. Although you can view the 10K rows, a confirmation window asks if you want to view the rest of the results.



Select **Yes** to view the entire result set. Loading all the data may take a while. You must fetch all data before working with Pivot tables, or it will lead to incorrect results from aggregation in Pivot tables.

Close the Query Wizard panel to cancel the operation of fetching the result.

Note

Close the Query Wizard panel to cancel the operation of fetching the result.

Read and Access Data Using Table Hyperlinks in Excel Sheet

Table Hyperlinks in Oracle Autonomous Database provide secure, preauthenticated URLs that allow read-only access to data stored in tables, views, or the result of SQL queries.

They enable easy sharing and retrieval of database data without requiring additional authentication beyond the hyperlink itself. You can read and access data in a Microsoft Excel sheet using a Table Hyperlink through the Oracle Autonomous Database Add-in for Excel, which allows direct retrieval and viewing of tabular data from the Autonomous Database within Excel.

You can use **Table Hyperlinks** with the add-in.

For more information of Table Hyperlinks, refer to the [Use Table Hyperlinks for Read Only Data Access on Autonomous Database](#) Chapter.

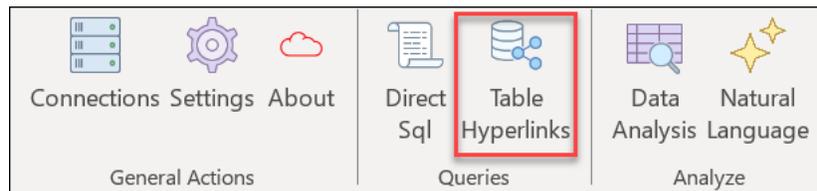
Note

This feature allows read-only access to data through secure, table hyperlink URLs without requiring an additional database connections or an Oracle Autonomous Database account.

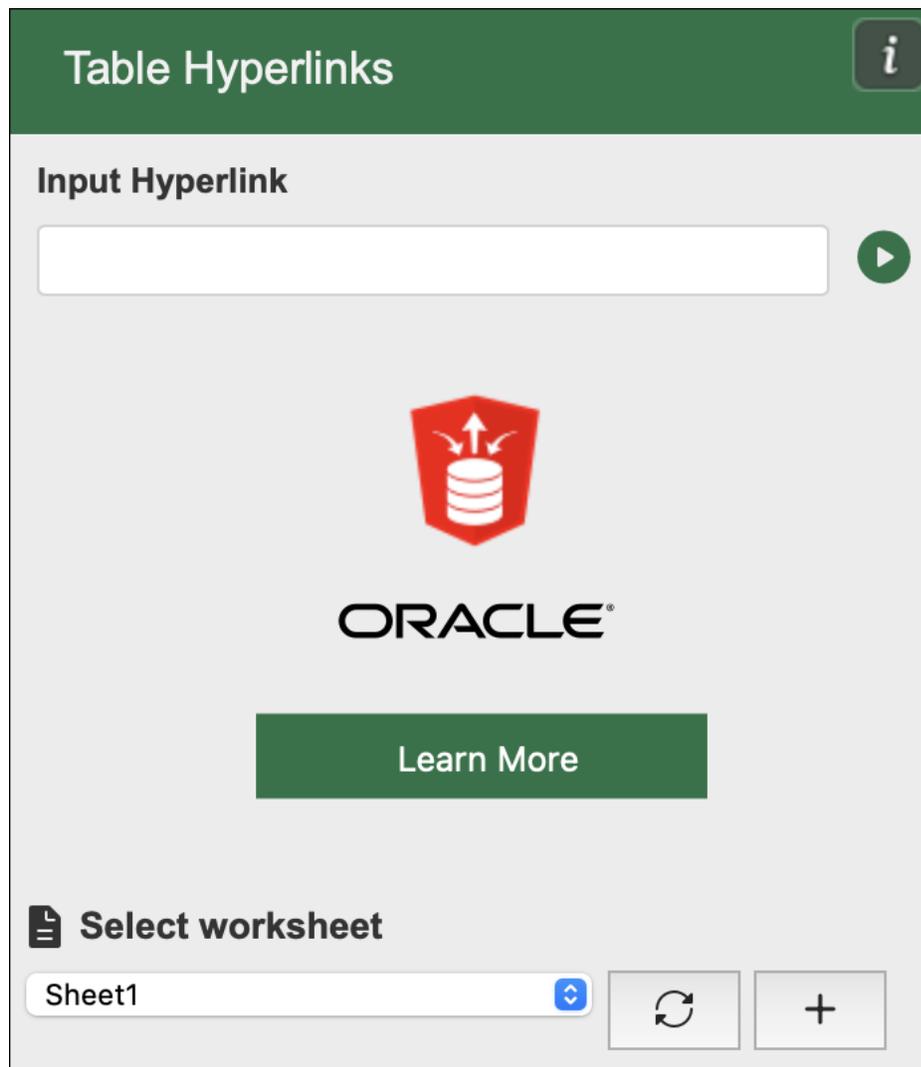
Use **Table Hyperlinks** to read and access data and view the data in a tabular format.

To return data from the add-in using Table Hyperlinks:

1. On the Excel sheet, select the menu **Autonomous Database**.
2. Select **Table Hyperlinks**.



3. Selecting Table Hyperlinks opens a **Table Hyperlinks** panel.



4. On the **Input Hyperlink** field, enter the input Table Hyperlink the producer shares with you. A Producer generates a PAR URL in their database and provides it to you. For example, <https://dataaccess.adb.eu-frankfurt-1.oraclecloudapps.com/adb/p/F5Sn....> You can create Table Hyperlinks in Oracle Autonomous Database Serverless using the `DBMS_DATA_ACCESS` PL/SQL package and share them with the consumers.

See [DBMS_DATA_ACCESS Package](#) for more information on this topic.

Select Worksheet: Enter the name of the sheet where you want to display the results. For example, *Sheet1*.

You can click the + sign beside the Select worksheet drop-down to display the results in a new worksheet.

5. Click the **Play** button to load the Autonomous Data table in the worksheet. The result data gets retrieved in the worksheet. You can view only first 10000 records in the first load.

A	B	C	D	E	F	G
EEID	FULL_NAME	JOB_TITLE	DEPARTMENT	BUSINESS_UNIT	GENDER	ETHNICITY
E03549	Mateo Vu	Account Representative	Sales	Speciality Products	Male	Asian
E02005	Isabella Bui	Enterprise Architect	IT	Manufacturing	Female	Asian
E02139	Rylee Yu	Director	Accounting	Research & Development	Female	Asian
E03928	Miles Dang	IT Coordinator	IT	Speciality Products	Male	Asian
E04109	Leah Bryant	IT Coordinator	IT	Manufacturing	Female	Caucasian
E03719	Jack Brown	Analyst	Marketing	Corporate	Male	Caucasian
E02732	Alice Tran	Analyst	Marketing	Corporate	Female	Asian
E04332	Luke Martin	Analyst	Finance	Manufacturing	Male	Black
E02183	Sarah Ayala	Analyst	Sales	Corporate	Female	Latino
E01361	Emma Hill	IT Coordinator	IT	Manufacturing	Female	Caucasian
E01895	Gabriel Zhou	IT Coordinator	IT	Manufacturing	Male	Asian
E02798	Charles Henderson	Systems Analyst	IT	Speciality Products	Male	Caucasian
E04601	Allison Ayala	Analyst	Finance	Corporate	Female	Latino
E00691	Avery Yee	Systems Analyst	IT	Manufacturing	Female	Asian
E01877	Abigail Garza	Analyst	Accounting	Manufacturing	Female	Latino
E03655	Samantha Rogers	Analyst	Marketing	Speciality Products	Female	Caucasian
E02639	Sebastian Le	Analyst	Finance	Corporate	Male	Asian
E00145	Kinsley Dixon	Analyst	Accounting	Manufacturing	Female	Caucasian
E02252	Lillian Park	Analyst	Marketing	Research & Development	Female	Asian
E01466	Connor Vang	Analyst	Sales	Speciality Products	Male	Asian

After selecting the play icon, along with the results retrieved in the worksheet, you can also see the following generated fields:

- **Load More Data:**
Select **Load More Data** to retrieve more data from the table.
- **Color Settings:**
Select this icon to enable color in the column from a predefined pool of colors. If the feature runs out of colors, the pool is refilled with the same colors for the new value keys.

Click the icons you want to add color to. In this example, select *JOB_TITLE* and *BUSINESS_UNIT*.

Select Columns for Coloring

EEID

FULL_NAME

JOB_TITLE

DEPARTMENT

BUSINESS_UNIT

GENDER

ETHNICITY

Enable Color
Reset Color

- Select **Enable Color** to color the selected columns in the data.

A	B	C	D	E	F	G
EEID	FULL_NAME	JOB_TITLE	DEPARTMENT	BUSINESS_UNIT	GENDER	ETHNICITY
E03549	Mateo Vu	Account Representative	Sales	Speciality Products	Male	Asian
E02005	Isabella Bui	Enterprise Architect	IT	Manufacturing	Female	Asian
E02139	Rylee Yu	Director	Accounting	Research & Develo	Female	Asian
E03928	Miles Dang	IT Coordinator	IT	Speciality Products	Male	Asian
E04109	Leah Bryant	IT Coordinator	IT	Manufacturing	Female	Caucasian
E03719	Jack Brown	Analyst	Marketing	Corporate	Male	Caucasian
E02732	Alice Tran	Analyst	Marketing	Corporate	Female	Asian
E04332	Luke Martin	Analyst	Finance	Manufacturing	Male	Black
E02183	Sarah Ayala	Analyst	Sales	Corporate	Female	Latino
E01361	Emma Hill	IT Coordinator	IT	Manufacturing	Female	Caucasian
E01895	Gabriel Zhou	IT Coordinator	IT	Manufacturing	Male	Asian
E02798	Charles Henderson	Systems Analyst	IT	Speciality Products	Male	Caucasian
E04601	Allison Ayala	Analyst	Finance	Corporate	Female	Latino
E00691	Avery Yee	Systems Analyst	IT	Manufacturing	Female	Asian
E01877	Abigail Garza	Analyst	Accounting	Manufacturing	Female	Latino
E03655	Samantha Rogers	Analyst	Marketing	Speciality Products	Female	Caucasian
E02639	Sebastian Le	Analyst	Finance	Corporate	Male	Asian
E00145	Kinsley Dixon	Analyst	Accounting	Manufacturing	Female	Caucasian
E02252	Lillian Park	Analyst	Marketing	Research & Develo	Female	Asian
E01466	Connor Vang	Analyst	Sales	Speciality Products	Male	Asian

You can select **Reset Color** to reset your current selection, remove colors from all the selected columns and choose different columns you wish to enable color to.

- **Auto Refresh Data:**

Enable this option to automatically update data at regular intervals with the latest data from its source database. This automatic refresh keeps the tabular data in worksheet in sync with the current data within a specific refresh frequency.

You can select the frequency from the frequency slider where the selection varies from a range of 5 minutes to 1 day.

- **Reload:** Select **Reload** to reload the preloaded table hyperlink data.

Note

This button loads a maximum of 50,000 records.

- **Reset:** Select **Reset** to clear all data from the sheet, reset your selection of all the current options and enter the **Input hyperlink** again.

Reporting and Analysis in Excel

You can view Reports and Analytic Views or visualizing data for analysis purpose using the **Analysis and Reports** menu.

Note

This menu is unavailable if you install **Oracle Autonomous Database for Excel** using Microsoft Store.

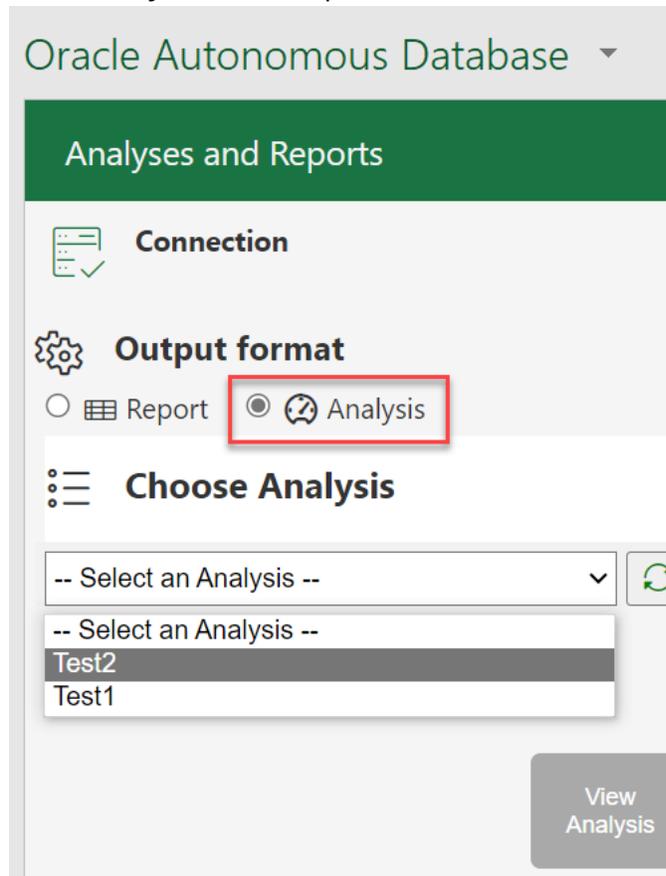
The reports and charts can be viewed in of different range of charts, namely, Bar Chart, Area Chart, Line Chart and Pie Chart. The Analyses and Reports icon enables you to retrieve Analyses and Reports from the Autonomous Database.

View Analysis

To view Analysis and explore the Analyses and Reports menu:

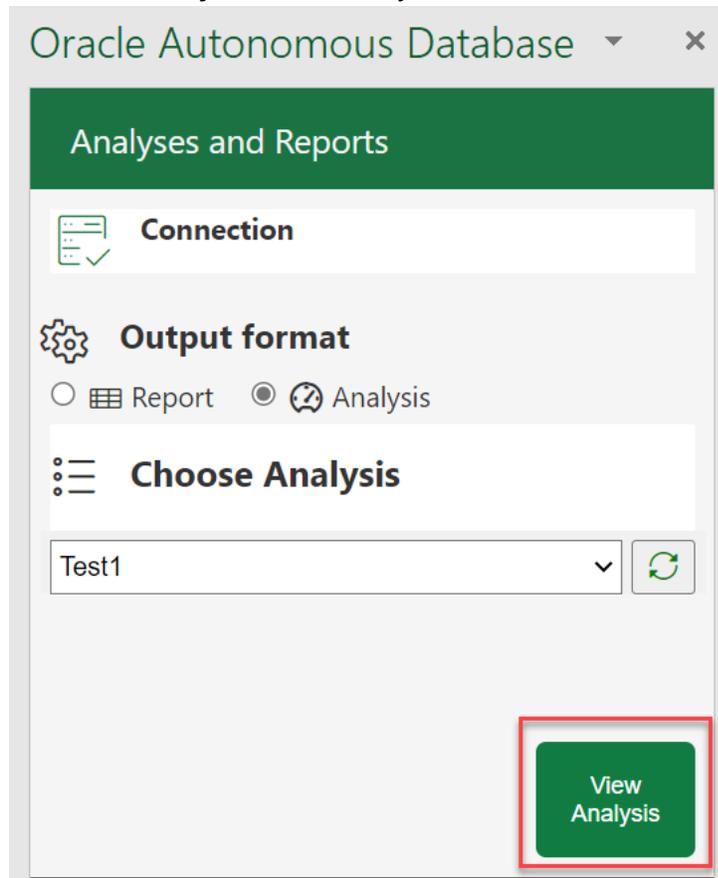
1. On the MS Excel ribbon, select the **Analyses and Reports** icon from the Autonomous Database menu.

2. Select **Analysis** under Output format.



3. Use the **Select an Analysis** drop down to choose the Analysis you want to view.

4. Click **View Analysis** to view analysis in a new excel sheet.

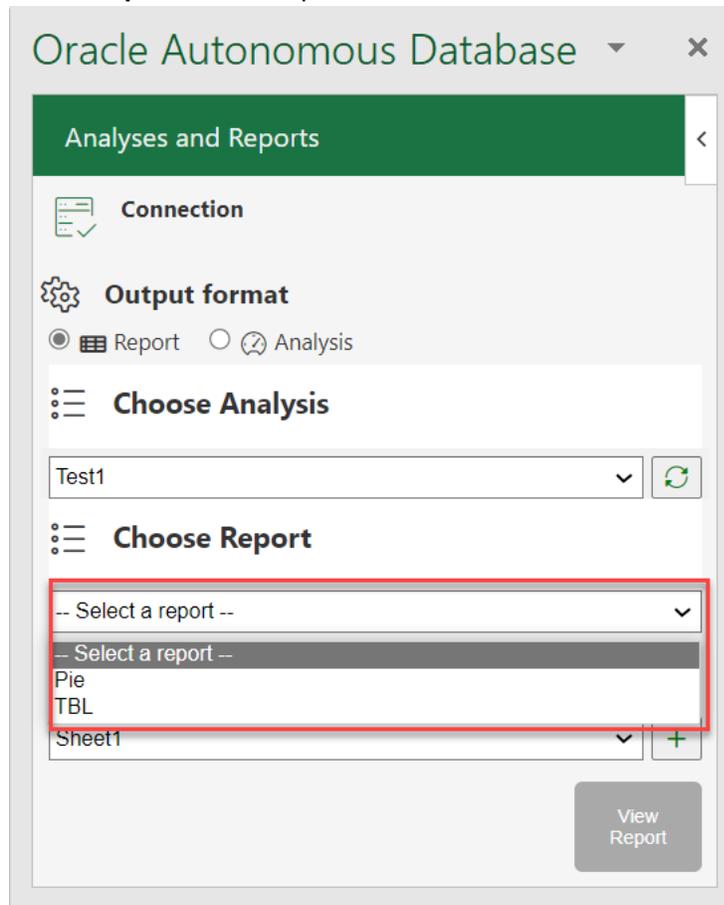


View Report

To view Reports :

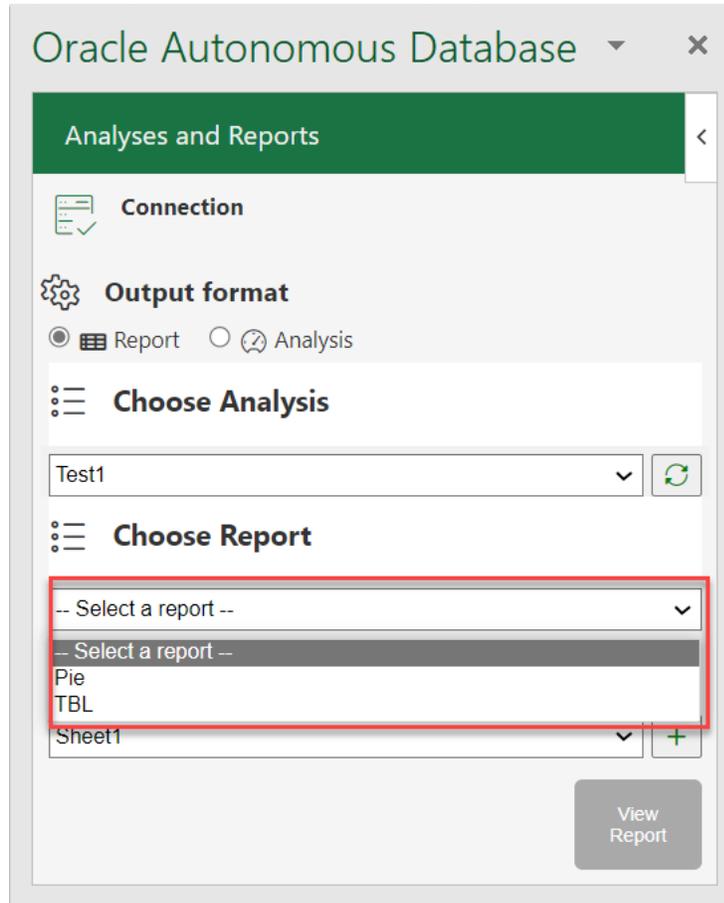
1. On the MS Excel ribbon, select the **Analyses and Reports** icon from the Autonomous Database menu.

2. Select **Report** under Output Format.



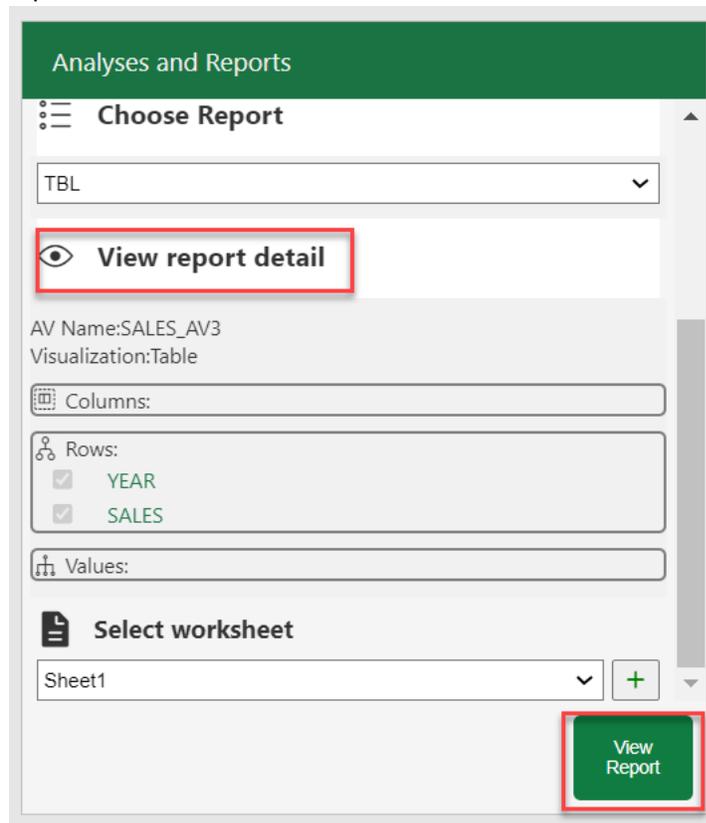
3. Use **Select an Analysis** drop down under **Choose Analysis** to choose the Analysis you want to view.

4. After you select the Analysis, to view the report present in the Analysis, click the **Select a report** drop-down and select the report you wish to view.



5. Click **View Report Detail** to view more information about the report namely, Analytic View Name, Type of visualization and rows, columns or values you select while creating the

report.



6. Select the worksheet from the drop-down where you would want to view the report.
 7. Click **View Report** to view the report in the sheet you select from the previous step.
- You can now view the report in the worksheet you select.

Query an Analytic View in an Excel worksheet

The Query Wizard menu enables you to query an Analytic View and retrieve the results in an Excel Worksheet. Once the wizard retrieves the data, it becomes local to Excel. You can further edit the data in Excel but not write back to the Autonomous AI Database.

Note

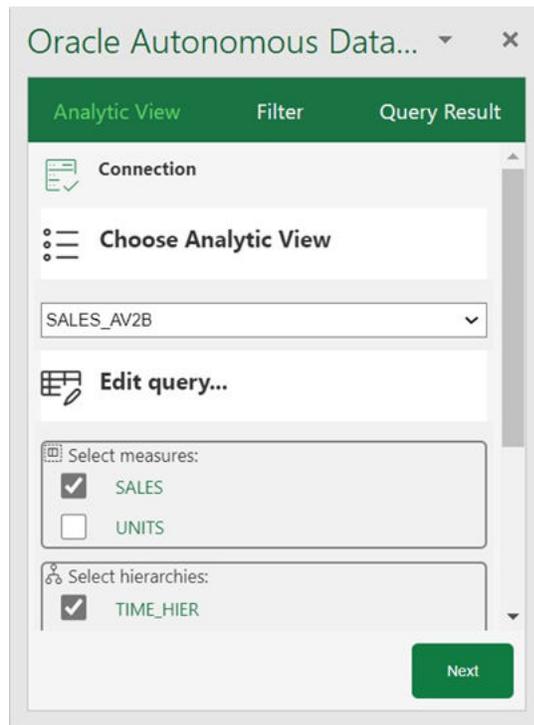
This menu is unavailable if you install **Oracle Autonomous Database for Excel** using the Microsoft Store.

You can query an Analytic View to visualize the result data in the worksheet. You can search for the Analytic View, and select measures, hierarchies, and levels from the query. You can also add filters and calculated measures to the query and view the query result in the spreadsheet.

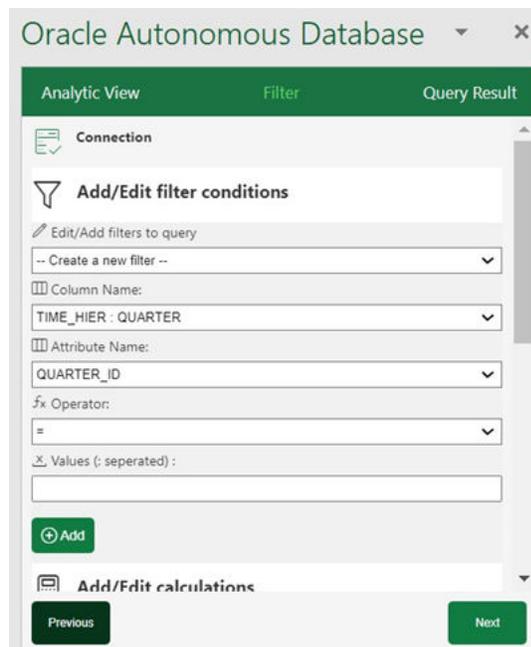
By default, the data is retrieved in tabular format. You also have the option to create an Excel pivot from this data.

The Query Wizard has three panels:

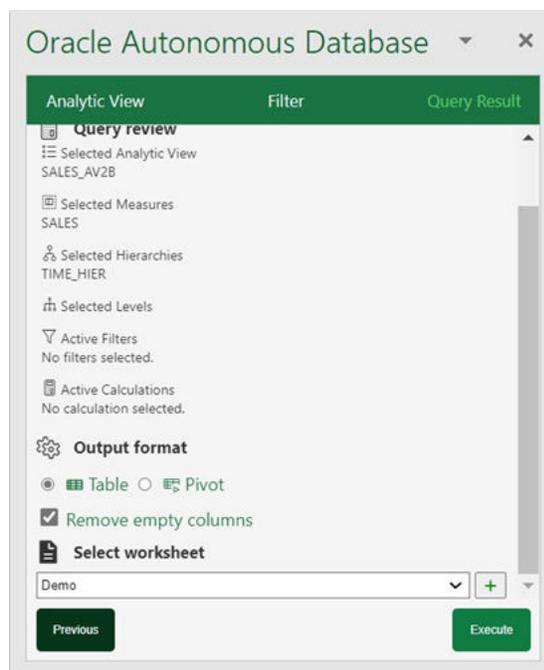
1. **Analytic View panel:** The Analytic View panel contains a list of Analytic Views from which you build queries. You edit the query by selecting
 - measures
 - hierarchies,
 - and levelsand progress to the next panel.



2. **Filter panel:** The Filter panel displays to the right of the Analytic View panel when you click **Next** on the wizard. You can create filter conditions to filter the data and also add manual calculations to the Analytic View query in this panel.



3. **Query Result panel:** When you click **Next** on the wizard, the Query Result panel displays to the right of the Filter panel. You run the query once you select the filter criteria and determine what calculated measures to add to your query. You can view and revise the SQL query. After the SQL query runs, you view the query results in the worksheet. You can select the output format of the result here. You can view the results in tabular format or a Pivot table.



To query an analytic view and explore the Query Wizard menu in the MS Excel ribbon:

1. On the ribbon, select the Query Wizard icon.

2. Selecting the Query Wizard opens an **Oracle Autonomous Database** dialog box in the Excel Task Pane.
3. Select an existing Analytic View from the drop-down in the Analytic View pane. As you select the Analytic View, it appears on the Analytic View field.
4. Select your choice of measures, hierarchies, and levels the available measures, hierarchies and levels associated with the Analytic View. Click **Next**.
5. The wizard window progresses to the **Filter** pane where you can add or edit filters to query.
6. Under Add or Edit filter conditions, do the following.
 - Select the column name and the attribute name from the drop-down- the values of the attribute change dynamically with the change in column names.
 - Select an operator in the Operator field to apply to the values that you specify in the Value field.
 - Specify a value or values from the list containing your selected column members. You need to enter the value into the Values field manually. For example, you can select > in the Operator field to use only values greater than the value that you select in the Value list. If you select 100,000 from the Value list, the filter uses values from the column greater than 100,000. You can use this information in an analysis to focus on products performing well. For multiple values use ":" as the separator.
 - Click **Add Filter** to add another filter condition.
7. Under Add or Edit Calculations, do the following.
 - Specify the column whose values you want to include in the group or calculated item.
 - On the Calc expression field, enter a custom calculated expression you want to perform on the column value. You can add functions or conditional expressions.
8. Click **Next** to progress to the **Query Result**.
9. You can view, edit, and review the query you have generated from the Query Review editor.
10. Select **Remove empty columns** to remove columns with no values returned in the result.
11. Select **Column per level** to retrieve all hierarchy levels in a single column.
12. Select the worksheet from the drop-down where you want to view the result.
13. Click **Execute** to run the query.
14. You can view the result of the query in the worksheet you select.
15. You can always modify the query in the Oracle Autonomous Database dialog box editor even after results are generated.
16. Select **Table** in the Query Result pane to view the results in the worksheet in a tabular format.
17. Select **Pivot** in the Query Result pane to view the results in a new worksheet in Pivot format.

View the results in Pivot tables

A Pivot table view is interactive and allows you to transpose rows and columns. A pivot table can summarize, sort, reorganize, count the total and perform an average of the result data. They are navigable and drillable.

Apart from tabular mode, to view the query results in pivot table mode, select the **Pivot Table** option in the Autonomous Database wizard. Click **Run** to view the query results in the Pivot table.

Clicking **Run** opens the query results in a new sheet with a PivotTable Fields wizard.

The screenshot shows an Excel spreadsheet with a PivotTable and the PivotTable Fields task pane. The PivotTable is titled 'Sum of sales' and has 'Row Labels' and 'Column Labels'. The data is organized by year and month, with a 'Grand Total' column. The PivotTable Fields task pane is open on the right, showing the 'Columns' area with 'year_name' selected and the 'Values' area with 'Sum of sales' selected.

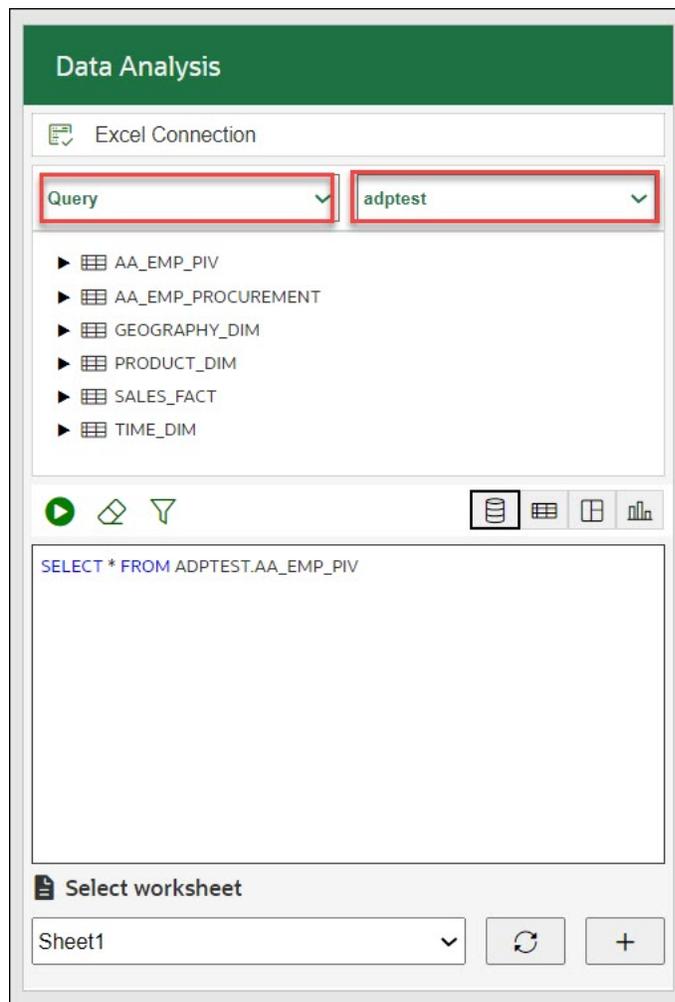
Row Labels	CY2011	CY2012	CY2013	CY2014	CY2015	(blank)	Grand Total
<1/11/2022	13510231961	13803364798	14481877435	15159492706	15882205770	36418586335	1.09256E+11
Jan	545626199	557903048.2	586697240.8	615564800.9	639168429		2944959718
11-Jan	545626199						545626199
12-Jan		557903048.2					557903048.2
13-Jan			586697240.8				586697240.8
14-Jan				615564800.9			615564800.9
15-Jan					639168429		639168429
Feb	516587219	511152916.7	534879895.4	560790625.6	605433694.4		2728844351
Mar	563086209.4	575801818.3	601994321.3	633500395.9	663497449.4		3037880194
Apr	556371561.4	564675899.1	597234064.5	626694389.9	653705205.6		2998681121
May	583962050.2	600112602.7	629297653.4	658664372.8	688767456.9		3160804136
Jun	574826596.4	587625680.2	614454114.5	644068466	675659370.3		3096634227
Jul	573020434.3	586637425.8	616822529.5	642472388.9	672973205.1		3091925984
Aug	560416194.2	574924871.2	604086459.4	632014115.9	659039357.9		3030480999
Sep	557581064.5	570811114.7	601009674.7	625527006.3	657213135.6		3012141996
Oct	586680546.8	602433312.1	631751487.9	658760860.7	686954599.2		3166580807
Nov	563791422	578250122.5	605937555.1	634859016.7	663712789.3		3046550906
Dec	573166483.6	591353587.5	616773721	646829913.3	674978192.5		3103101898
Grand Total	20265347942	20705047197	21722816153	22739239059	23823308655	36418586335	1.45674E+11

Data Analysis in Excel Sheet

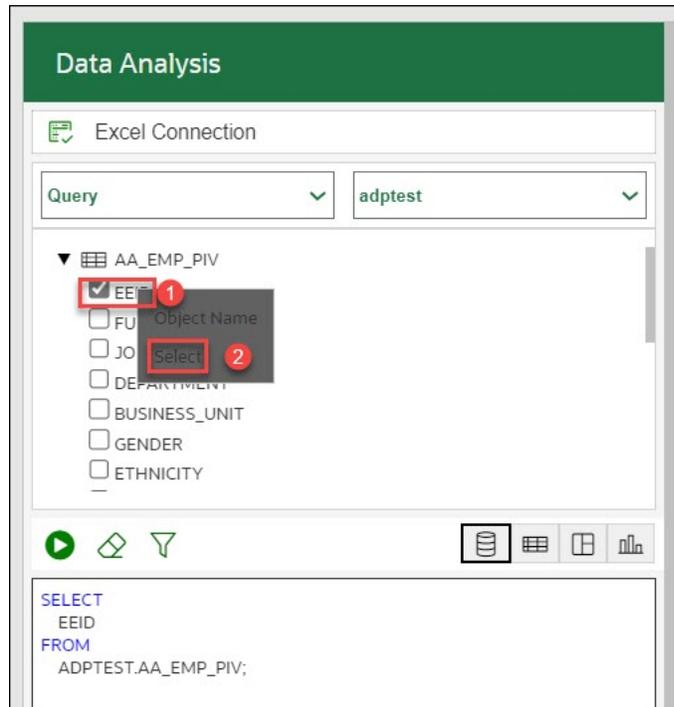
The Data Analysis tool enables you to analyze data in the Autonomous Database by running SQL queries or querying an Analytical View using an intuitive drag-and-drop interface. You can also write custom queries to be run. You now can dynamically apply filters to the result set retrieved using the new faceted search capability on specific columns.

Run query on database tables using the add-in:

1. On the Excel Sheet, select the menu item **Autonomous Database**.
2. Select **Data Analysis**. Selecting Data Analysis opens a Data Analysis panel. On the Data Analysis panel, select **Query** from the drop-down and the **schema** you want to use from the drop-down.



3. You can select a column of the table you want to query, right-click the column, and click **Select** to assist the add-in in forming a select query of the column from the table. Alternatively, you can drag and drop the selected column to the query area that enables the panel to produce a select query of the column in the query display area.



You will view the default query in the query editor area.

- You can select any of the four modes to visualize the results of the SQL query report you generate.



- Base Query:** This type of view is by default. The query written in the SQL editor is the Base Query. This query will be the base query of the other three modes.

Note

All the numeric columns are displayed in different colours.

The Table browser now has all the columns from the base query.

- Table:** You can view the SQL results in tabular form. When you select this view, a column drop zone appears for selecting Rows, enabling you to drag and drop columns of the base query. Moving the selected columns in the drop zone allows you to view only those columns in the Result data generated in the worksheet. Select the cross mark beside the Column name to remove it from the drop zone.
- Pivot:** You can view the SQL query results in pivot format. By selecting this format, an X and Y drop zone appears where you can drag and drop the selected columns from the Tables browser based on the Base Query to the drop area.

Note

Only numeric values are allowed to be dropped in the values section.

- **Chart:** You can view *Area Chart*, *Bar Chart*, *Line Chart*, or *Pie Chart* when you select this option. The mappings displayed when you select one of the options are as follows:
 - **Orientation:** Choose between horizontal and vertical orientation types from the drop-down list.
 - **X axis label and Y axis label:** Optionally enter labels for X axis and Y axis.

Note

Only numeric values are allowed to be dropped in the Y axis drop zone.

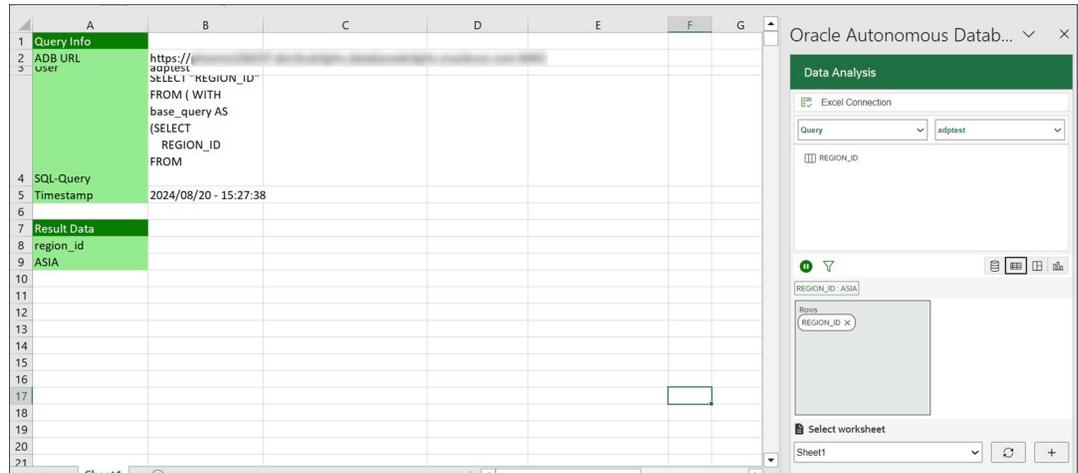
5. Click the **funnel icon** (Faceted Filter) to add filters to the result. The panel generates a filter for each value in the column retrieved from the query result. You can filter different columns on the faceted filter panel and view the results in the worksheet to view only the data you wish. For example, to view the customer reports by Region, click the **faceted filter** and select **Asia** under `Region_ID`. The number in the brackets displays the count of the items with this property. In the example shown below, there are 79 records with the region as ASIA.

The screenshot shows a 'Data Analysis' panel with the following elements:

- Data Analysis** (Header)
- Excel Connection (Dropdown)
- Faceted Visual (Toggle)
- REGION_ID (Section Header)
- ASIA(79) (Selected)
- SOUTH_AMERICA(37)
- NORTH_AMERICA(30)
- AFRICA(21)
- EUROPE(13)
- OCEANIA(1)
- Back (Button)
- Save (Button)

6. Click **Save** to view the results. Click **Back** to go back to the main panel.

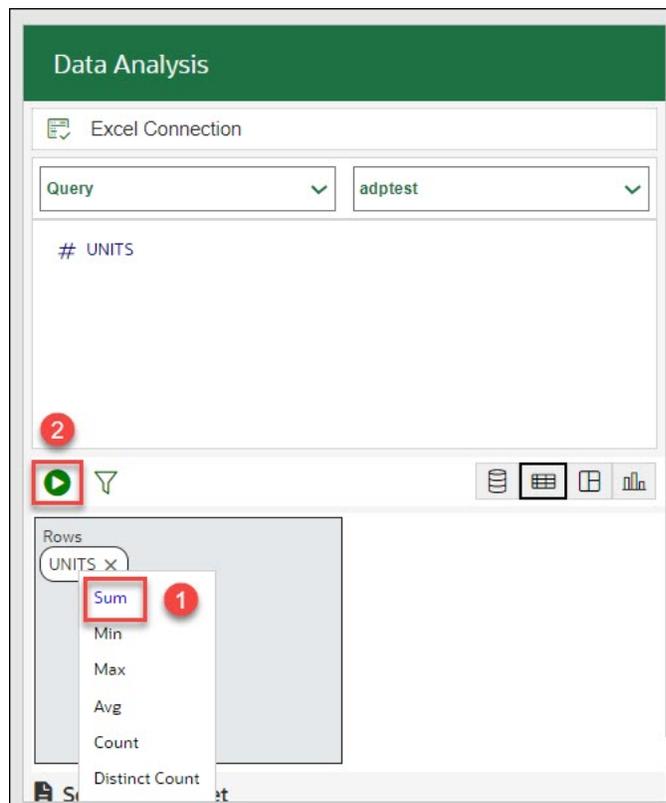
7. Select **Run** to generate the results of the custom query in the worksheet. Click **Pause** to make any changes to the query, such as updating the columns of the table without updating the worksheet.



Perform aggregate functions using the Excel add-in

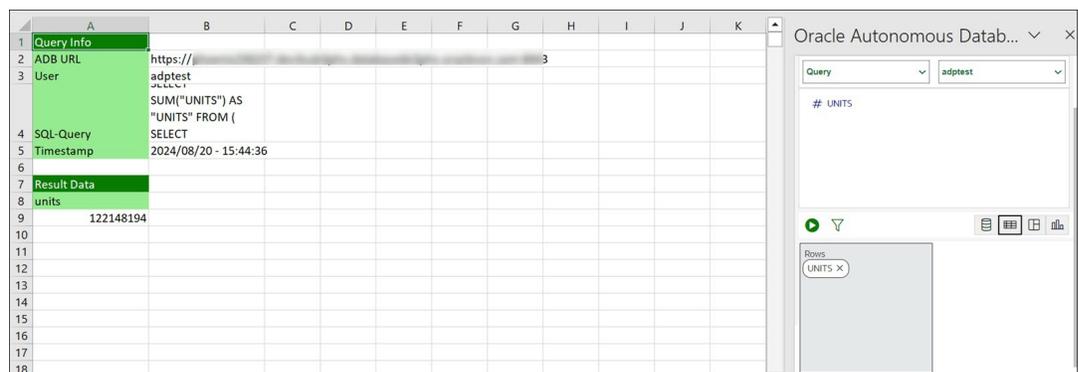
You can also perform aggregate functions such as SUM, MIN, MAX, AVG, COUNT, and DISTINCT COUNT. In this example, we're primarily going to focus on using the Data Analysis feature to gain insights from our sales data.

1. Select **Data Analysis**. Selecting Data Analysis opens a Data Analysis panel. On the Data Analysis panel, select **Query** from the drop-down and the **schema** you want to use from the drop-down. Drag and drop the sales value to the query editor and click **Table** to view the sales in tabular format.
2. To calculate the sum of the sales value, click the sales value and select **Sum** from the list of available aggregate functions.



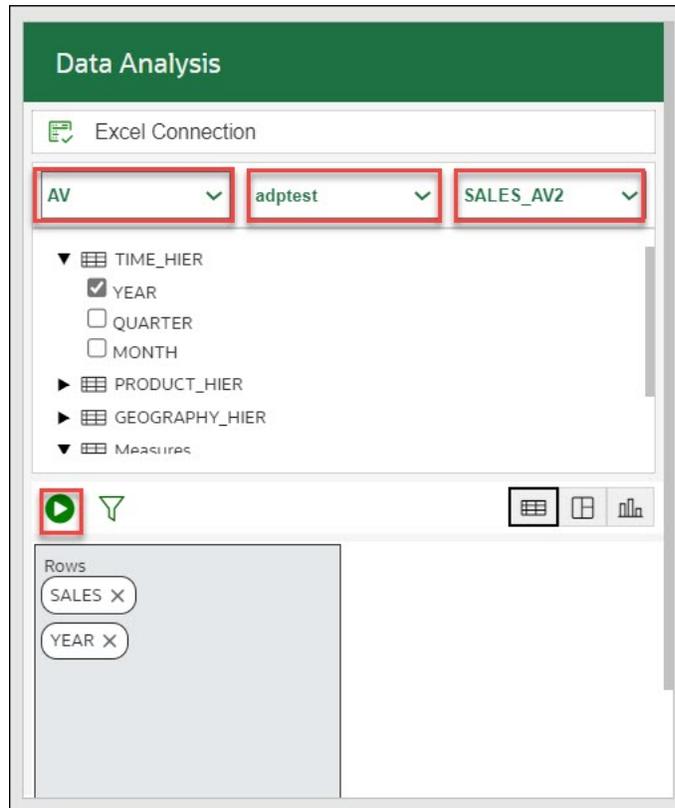
Click **Run** to generate the sum of sales amount in the tabular format.

You will view the result generated in the excel worksheet.



To query an Analytic View and explore the **Data Analysis** menu in the Excel Sheets:

1. Select the ribbon item **Autonomous Database > Data Analysis** on the Excel Sheet. This opens a **Data Analysis** panel.
2. Select **AV** from the drop-down, select a schema you can access from the schema drop-down, and the Analytic View from the list of available Analytic Views.

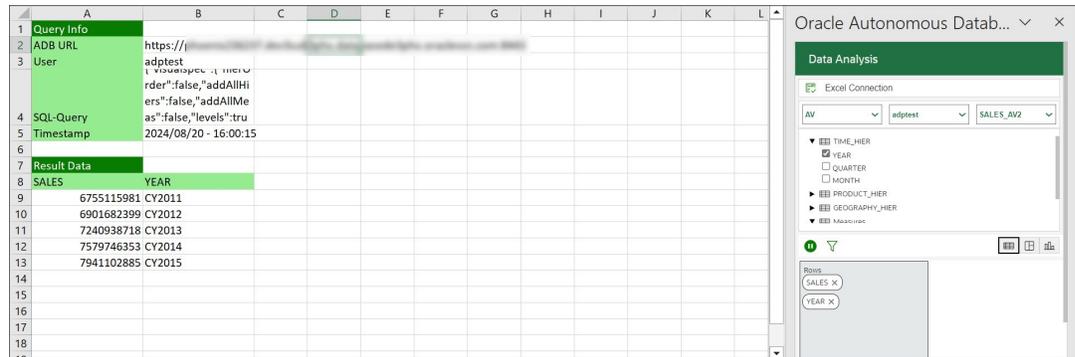


3. You can select any of the three modes to visualize the results of the AV query you generate:
 - **Table:** You can view the AV query results in tabular form. By selecting this view, a column drop zone appears, enabling you to drag and drop selected columns from the Table browser. Moving the selected columns in the drop zone allows you to view only those columns in the Result data generated in the worksheet. Select the cross mark beside the Column name to remove it from the drop zone.
 - **Pivot:** You can view the SQL query results in pivot format. By selecting this format, an X and Y drop zone appears where you can drag and drop the selected columns from the Tables browser to the drop area.
 - **Chart:** You can view the results of the AV query in chart format. By selecting this format, an X and Y drop zone appears where you can drag and drop the chosen hierarchies and measures from the AV browser to the drop area.

Note

You are allowed to drop measures in the Y-axis.

4. Click **Run** to view the results in the worksheet . You can view the total Sales generated along with it's year of generation.



Natural Language in Excel Sheet

You can use Natural Language Query to query the Oracle Autonomous AI Database using the **Natural Language** menu in the **Oracle Autonomous Database for Excel**.

Prerequisites

- Before you use the Natural Language menu in the add-on, you must [perform prerequisites](#) to use DBMS_CLOUD_AI to configure AI profiles.
- [Create and Set an AI Profile](#)

After your AI Profile is ready, the Data Studio tool uses these profiles in the [Data Studio Settings](#) menu to configure access to a Large Language Model (LLM) and to setup for generating, running, and explaining SQL based on natural language prompts. This also enables chatting with the LLM.

Note

You can run Natural Language Queries on Tables and not Analytic Views.

Generate SQL Queries from Natural Language on Tables

Using natural language to interact with your database data is now achievable with Oracle Autonomous Database add-on for Google Sheets.

This means you can use natural language, for example, plain English, to query the database. You can provide a natural language prompt instead of SQL code to interact with your data. When you select the **Generate SQL from natural query** icon, the add-on converts natural language to SQL.

To run a natural query using the add-on:

1. On the Excel Sheet, select the menu **Oracle Autonomous Database**.
2. Select **Natural Language**. Selecting **Natural Language** opens a **Natural Language** wizard.
3. On **Natural Language** wizard, select the **Table** on which you perform the query from the drop-down and the **AI profile** you want to use from the drop-down.

Note

This is the same AI profile you configure in the [Data Studio Settings](#).

4. Let's say you want details of the product with highest sales. Enter the following natural language query in the query display area:

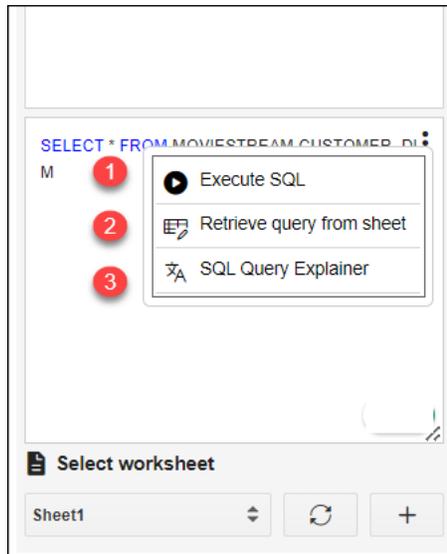
```
show movie with highest sales
```

5. Select **Generate SQL from natural query** to produce the equivalent SQL query in the bottom query display area.
You will view the following code in the bottom SQL code area.

```
SELECT
  *
FROM
  "MOVIESTREAM"."MOVIESALES_WEEKEND_USA" MS
WHERE
  MS."SALES_AMT" = (
    SELECT
      MAX("SALES_AMT")
    FROM
      "MOVIESTREAM"."MOVIESALES_WEEKEND_USA"
  )
```

6. Selecting the **Generate SQL from natural query** icon also displays the result of the natural language query in the Google Sheet.
You can click the + sign beside the **Select worksheet** drop-down to display the results in a new worksheet.
7. Click **Clear query content** to clear the content displayed on the natural language query area and the SQL code query area.

Actions you can perform from the SQL code area:



- **Execute SQL:** Runs SQL code and displays the query result in Google Sheet.
- **Retrieve query from Sheet:**
You can retrieve SQL query displayed in the Google Sheet to the SQL code area in the Natural Query.

This feature is to retrieve the latest data from the database for a future date. For example, if you want to receive the latest data from the table the next day, the connection may be inactive, but you can still connect to the instance, **retrieve query from Sheet** and **select Execute SQL** to display the latest query results in the Google Sheet.
- **SQL Query Explainer:** Explains SQL code in Natural language form.
Select this icon to translate SQL queries to natural language that is understood by you. The natural language query is displayed in the natural language query display area of the wizard.

Ask Questions with Chats

Chats option in the Oracle Autonomous Database for Google Sheets refer to an interactive conversation between you and the add-on where the add-on uses natural language to query or interact with the Autonomous AI Database.

The data we use in this example is of a company called Oracle MovieStream to analyze movie sales data.

The Chats displays recommendations for the default table you select.

Let's find out the top 5 streamed movies in the Moviestream company by weekend gross:

1. Enter the text in the **Start typing your question..** text field.

What are the top 5 movies by Weekend Gross?

2. Click **Enter**.
3. The Chat displays top 5 movies by Weekend Gross.

It also displays equivalent SQL code of the result.

Click **SQL** to expand the SQL.

Click **Copy to clipboard** to copy the SQL.

Click **Run SQL** to run the generated SQL query and display the query result in Google Sheet.

The **Chats** option remembers the context of previous chat history.

For example, if you enter `now show me top 10` in the text field.

It displays the top 10 movies by Weekend Gross. It remembers which metric top 10 to fetch without us having to type the whole thing.

Select **New Chat** for deleting the present conversation.

Select **Home** to go back to the main home page.

Note

LLMs are remarkable at inferring intent from the human language and they are getting better all the time; however, they are not perfect! It is very important to verify the results.

FAQs for Troubleshooting errors with Excel Add-in

If you experience issues with Oracle Autonomous Database for Excel, refer to frequently asked questions in this section to identify and resolve issues.

1. Why is the **My Add-ins** icon from the **Insert** ribbon in the MS Excel workbook greyed out?

Even before installing the Excel add-in, sometimes the **My Add-ins** icon from the **Insert** ribbon in the MS Excel workbook appears greyed out.

- a. From the **File** menu in the Excel ribbon, go to **Account** and select **Manage Settings** from the Account page.
- b. Ensure that you select the **Turn on optional connected experiences**.
- c. From the **File** menu in the Excel ribbon, go to **Options** and select the **Trust Center** option from Excel Options.
- d. Click **Trust Center Settings** and ensure that deselect **Disable all Application Add-ins** (if selected) from the Add-ins tab in the Trust Center dialog box.
- e. Select the **Trusted Add-in Catalogs** menu from the Trust Center dialog box and ensure that you deselect the **Don't Allow any web add-ins to start** checkbox.

2. Why doesn't the sign-in page of the Excel Add-in load or appear?

Sometimes you might encounter issues with the Excel Add-in even after loading it correctly. For example, an add-in fails to load or is inaccessible. Check the compatibility version of Excel and the operating system you use.

If the compatibility is correct and the sign-in page to the Excel Add-in still does not show up, or does not load properly, we recommend applying all pending Windows, Office, and browser updates.

- a. Select **Settings, Update & Security**, and then **Windows Update** from the Windows Start menu.
- b. If updates are available on the Windows Update page, review the updates and click **Install Now**.

Note

The details of applying Windows updates can vary from version to version and if required, check with your system administrator for assistance.

3. Why doesn't the add-in work correctly after re-installing?

Configure the Excel trusted Add-in catalog to set the add-in correctly after re-installation.

To configure the Excel add-in, check or remove the add-in if it is pointing at the wrong location in the Trusted catalog address. This address should be the same as the shared manifest folder's location (share path).

Click Excel's **File > More > Options > Trust Center > Trust Center Settings > Trusted Add-in Catalogs**

Trusted Web Add-in Catalogs

Use these settings to manage your web add-in catalogs.

Don't allow any web add-ins to start.

Don't allow web add-ins from the Office Store to start.

Next time Office starts, clear all previously-started web add-ins cache.

Trusted Catalogs Table

You should only add a catalog if you trust its owner. You may also select one of each catalog type to show in the insert add-in menu. We will automatically start web add-ins from your insert add-in menu catalogs when opening documents.

Catalog Url:

Trusted Catalog Address	Catalog Type	Show in Menu
\\Catalog Address	Network share	<input checked="" type="checkbox"/>

Checking is only required the first time you use the installer, or if the shared manifest folder is changed. The change occurs during uninstalling and re-pointing to a new ADB.

To remove the catalog from the trusted table and add a new catalog pointing to a different address:

- Select the Catalog you want to remove from the trusted catalog table and click **Remove**.
- Enter the correct share path of the shared manifest folder in the **Catalog URL field** and click **Add Catalog** to add the shared folder to the trusted catalog.

Restart Excel to make the new shared folder active to access the add-in.

4. Why doesn't the add-in work correctly even after configuring the Excel trusted Add-in catalog?

Let's say you configure the Excel trusted add-in catalog after re-installing the add-in but, it does not load correctly. Sometimes the database server changes are not reflected in Excel even after you set the share path of the shared manifest folder as a trusted add-in catalog. Clear the Office cache to resolve this issue.

Refer to this page: <https://docs.microsoft.com/en-us/office/dev/add-ins/testing/clear-cache#clear-the-office-cache-on-windows> to clear the Office cache on Windows and Mac.

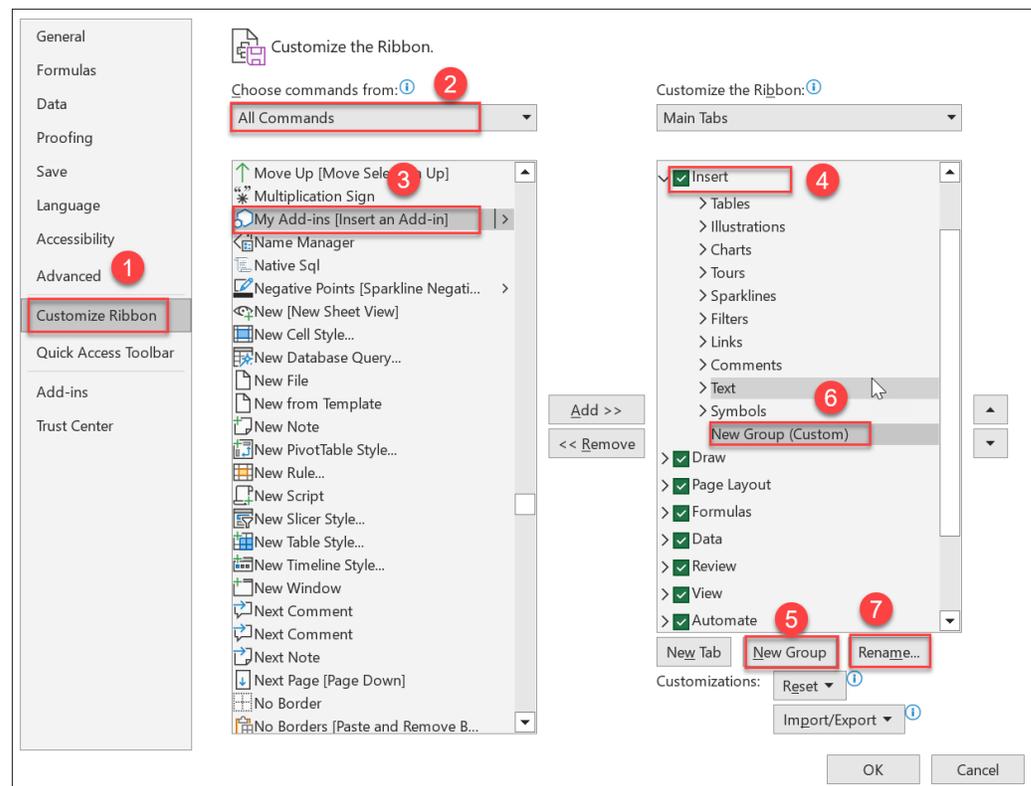
Clearing the Office cache unloads the Excel add-in. Install the add-in and check the configuration of the Excel trusted add-in catalog. This should solve the issue of the Excel add-in needing to be correctly loaded.

5. What should you do if you cannot view My Add-ins icon from the Excel ribbon?

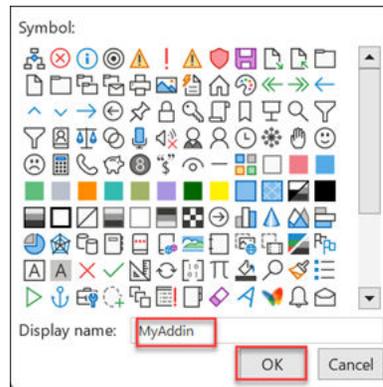
After installing the add-in once, if you cannot view the **My Add-ins** icon from the Insert ribbon and instead you view the Add-ins icon 

Follow these instructions:

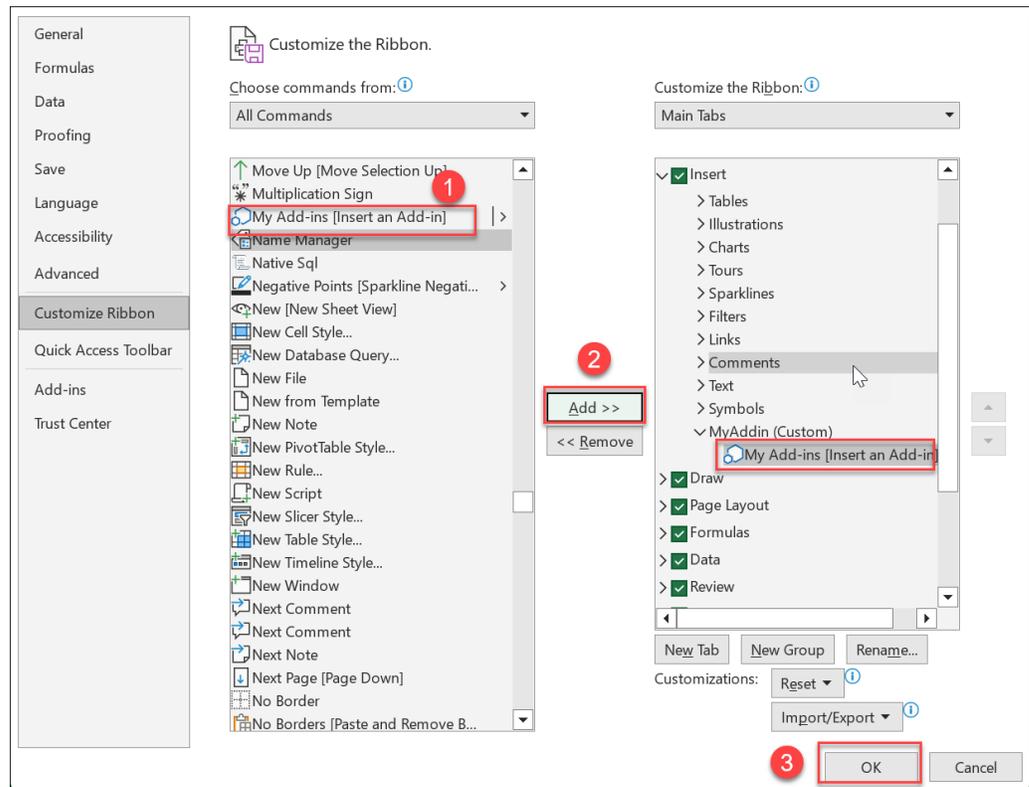
- From the **File** menu in the Excel ribbon, go to **Options** and select the **Customize Ribbon** option from Excel Options.
- Select **All Commands** drop-down from the **Choose commands from** drop-down.
- Click **My Add-ins** and click **Insert** option from the Main Tabs list.
- Click **New Group** to add a new menu item under the **Insert** menu. You will view a **New Group (Custom)** menu option added to the Insert menu.
- Click **Rename** to rename the newly created menu.



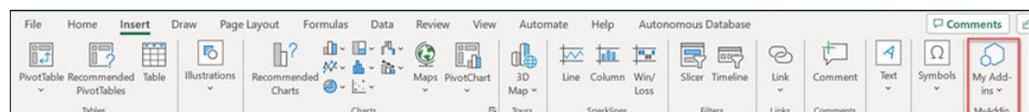
- f. Specify the name of the add-in. For example, **My Addin**. Click **OK**.



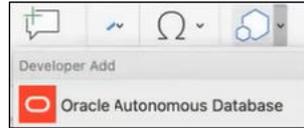
- g. Click **My Add-ins[Insert and Add-in]** from All Commands list and select **Add**. My Add-ins menu is added to the newly created menu “MyAddin”.
- h. Click **OK** to save the changes.



- i. Clicking **OK** takes you to the main Excel sheet page where you can view “**MyAdd-ins**” menu under the Insert menu.



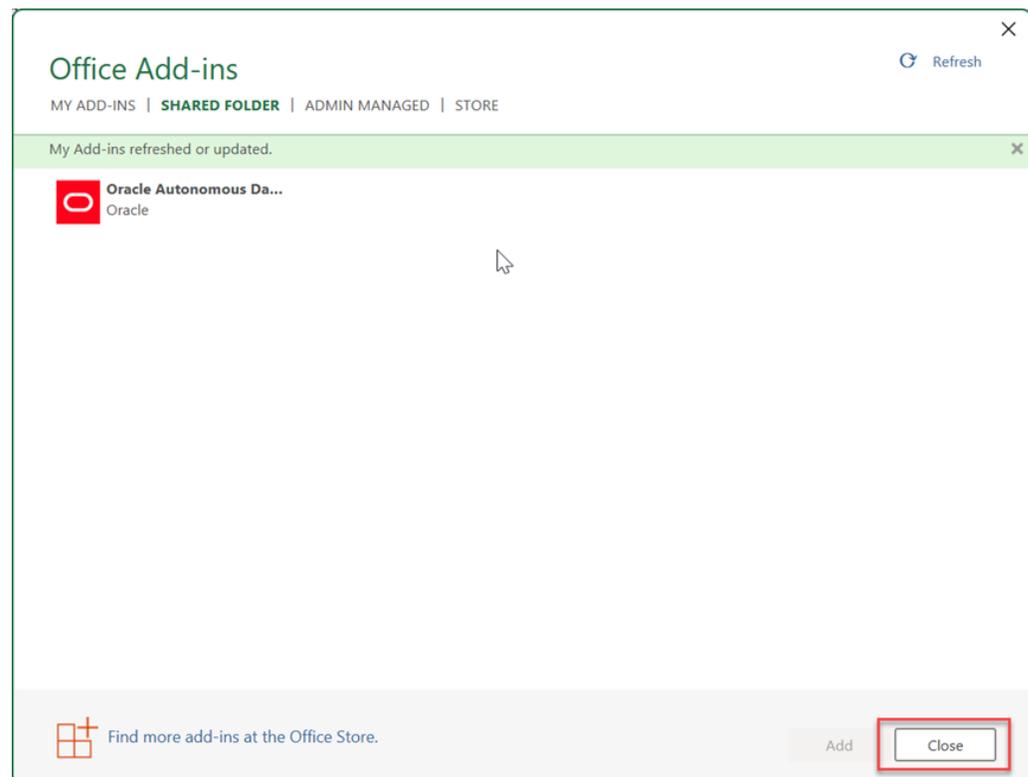
- j. Click **My Add-ins**. You can now view the Oracle Autonomous Database for Excel loaded.



6. What happens when you cannot view the latest added menu items in the **Oracle Autonomous Database for Excel** add-in?

Sometimes when you cannot view the changes updated in the latest version of the plug-in, you must:

- Select **My Add-ins** from the Insert menu on the Excel ribbon.
- Click the **Shared Folder** tab of the Office Add-ins dialog box. You will see the add-in under the Shared Folder list.
- Click **Refresh**.
After the add-ins are refreshed, you will receive a notification on the dialog that says "My Add-ins refreshed or updated". The refresh button loads the manifest file again for the latest changes to appear.
- Click **Close** to close the dialog box.

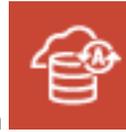


Oracle Autonomous Database for Excel Privacy Policy Details

This topic covers details for writing policies to control access to Autonomous Database resources.

Data use by **Oracle Autonomous Database for Excel** complies with Oracle's [Privacy Policy](#).

Oracle Autonomous Database for Google Sheets



The Oracle Autonomous Database add-on enables you to query tables using SQL or Analytic Views using a wizard directly from Google Sheets for analysis.

The data retrieved from the Autonomous Database is available locally in Google Sheets for further analysis. The results are stored in the local copy and cannot be written back to the Autonomous Database. You can run direct SQL queries or query Analytic Views and view their results in the worksheet. The add-on allows you to filter the query results, and perform table joins and calculations. You can also generate and manage Table Hyperlinks to access data on Autonomous Database.

Note

The Oracle Autonomous Database add-on for Google Sheets must comply with [Privacy Policy](#). For information on details of privacy policy, see [Oracle Autonomous Database for Google Sheets Privacy Policy Details](#).

How does the add-on for Google Sheets work?

To query an Analytic View or Tables from the Autonomous Database, you must select an Analytic View or Table to work with. While retrieving data from the Analytic View, you can configure the query according to your requirements. You can select specific hierarchies and create custom calculations on the wizard. The add-on configures your query and returns the result to the Google Sheets. You can save the results of your queries locally in the Google Sheet. The add-on can also query the schema directly to which you have access. Using the Web UI, you can also view reports and analyses you create in the Data Analysis menu in the Data Studio tool.

To use the add-on, you must enable Web Access on the Autonomous Database account. You must have the CONNECT, DWROLE, RESOURCE and ADPUSER roles grant in the SQL worksheet to access the Google Sheets add-on.

Note

- The **Download Microsoft Excel/Google Sheets add-in** is available to you under the **Downloads** menu of your Database Actions instance only if you have the ADPUSER role.
- The Oracle Autonomous Database add-on for Google Sheets is not supported in Safari web browser.

- [Install the add-on from Google Workspace Marketplace](#)
The **The Oracle Autonomous Database add-on for Google Sheets** is available in **Google Workspace Marketplace**.

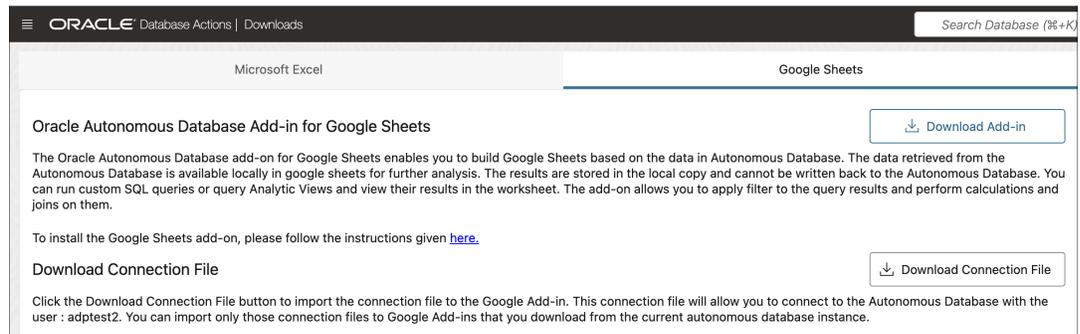
- [Download Connection File](#)
To connect to the Autonomous Database, you can download a connection file from the Database Actions instance and import it to the Google Sheet add-on you have setup.
- [Connecting to Autonomous Database](#)
The Oracle Autonomous Database add-on for Google Sheets enables you to connect to multiple Autonomous Databases with a single add-on using the Connections feature. The add-on connects to Google Sheets by providing authentication to Google. Multiple users or databases can connect simultaneously to the add-on. However, only one connection can remain active.
- [Generate Client ID and Client Secret using UI](#)
In this section you use the Web UI to obtain the `client_id` and `client_secret`.
- [Authorize Google Sheets to use Autonomous Database](#)
After your identity is determined using OAuth authentication, Google Sheets needs permission to access the Autonomous Database.
- [Natural Language in Google Sheets](#)
You can use Natural Language Query to query the Oracle Autonomous AI Database using the **Natural Language** menu in the **Oracle Autonomous Database for Google Sheets** "add-on".
- [Data Analysis in Google Sheets](#)
Selecting **Data Analysis** opens an Oracle Autonomous Database wizard in the Google sheet.
- [Run Direct SQL Queries](#)
The Oracle Autonomous Database add-on for Google Sheets lets you run SQL queries to work with your data in a Google Sheet. With the add-on, you can type your SQL code in the SQL editor area and click **Run** to run the command.
- [Read and Access Data Using Table Hyperlinks in Google Sheets](#)
Table Hyperlinks in Oracle Autonomous Database provide secure, preauthenticated URLs that allow read-only access to data stored in tables, views, or the result of SQL queries.
- [About Autonomous Database menu](#)
Use this option to view details about the add-in.
- [Share or Publish](#)
Once you generate the query results in the Google Sheet, you can share it with other users. With sharing, creates a copy of the worksheet and sends it with the design tools hidden and worksheet protection turned on.
- [Oracle Autonomous Database for Google Sheets Terms of Service](#)
You can view the following Terms of Service here:
- [Oracle Autonomous Database for Google Sheets Privacy Policy Details](#)
Effective Date: 10/4/2024
- [Oracle Autonomous Database for Google Sheets Support](#)
Welcome to the support page for the **Oracle Autonomous Database for Google Sheets**. This resource is designed to assist you with any issues or questions you may have while using the add-on.

Download Connection File

To connect to the Autonomous Database, you can download a connection file from the Database Actions instance and import it to the Google Sheet add-on you have setup.

Follow the steps shown below to download the connection file.

1. Navigate to the launchpad of your Database Actions instance, and select the **DOWNLOAD MICROSOFT EXCEL/ GOOGLE SHEETS ADD-IN** Card. Click the **Download Connection File** button in the Google Sheets tab of the **Downloads** page to import the connection file to the Google Add-in.
2. This connection file will allow you to connect to the Autonomous Database with the logged-in user. You can import only those connection files to Google Add-ins that you download from the current Autonomous Database instance.



3. Selecting the Download Connection File button downloads a connection file in JSON format. You can view the downloaded file in your `Downloads` folder.

Install the add-on from Google Workspace Marketplace

The **The Oracle Autonomous Database add-on for Google Sheets** is available in **Google Workspace Marketplace**.

Prerequisites

- You must have an active Google account to sign in as an *administrator*.

Follow these steps to install the add-in:

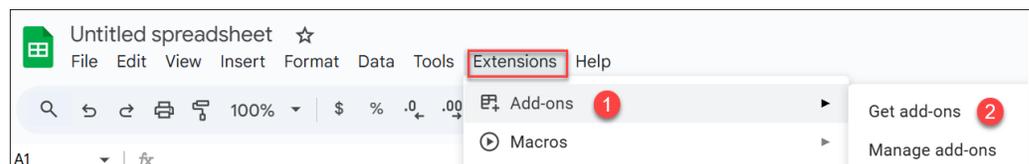
1. Start Google Sheets and open a new or existing sheet:

- You can access Google Sheets using a web browser. Go to *sheets.google.com*.
- Login with your account.

You will view Google Sheets.

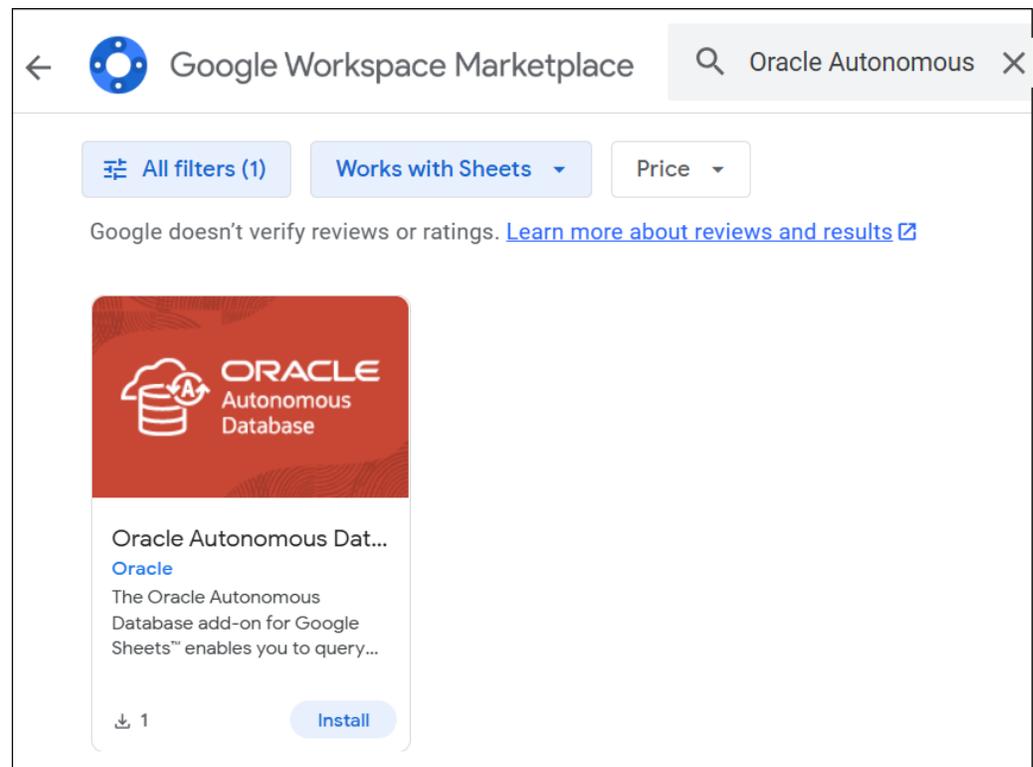
2. Install the Add-on from Google Workspace Marketplace:

- From the Home menu, go to **Extensions**, click **Add-ons** and select **Get add-ons**.

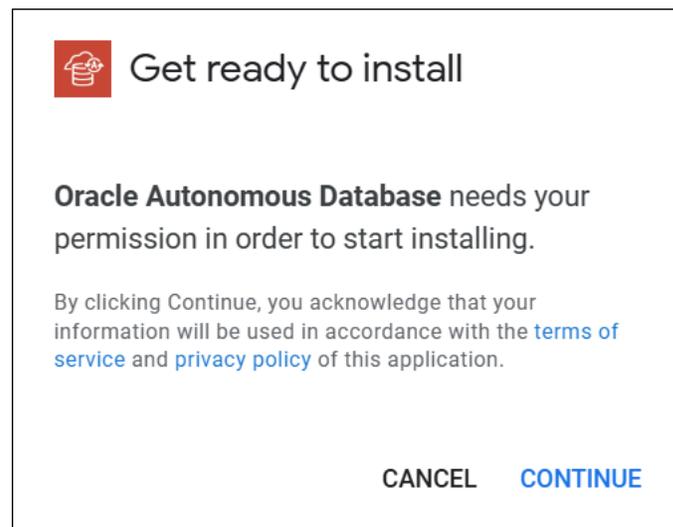


You can view the **Google Workspace Marketplace** application.

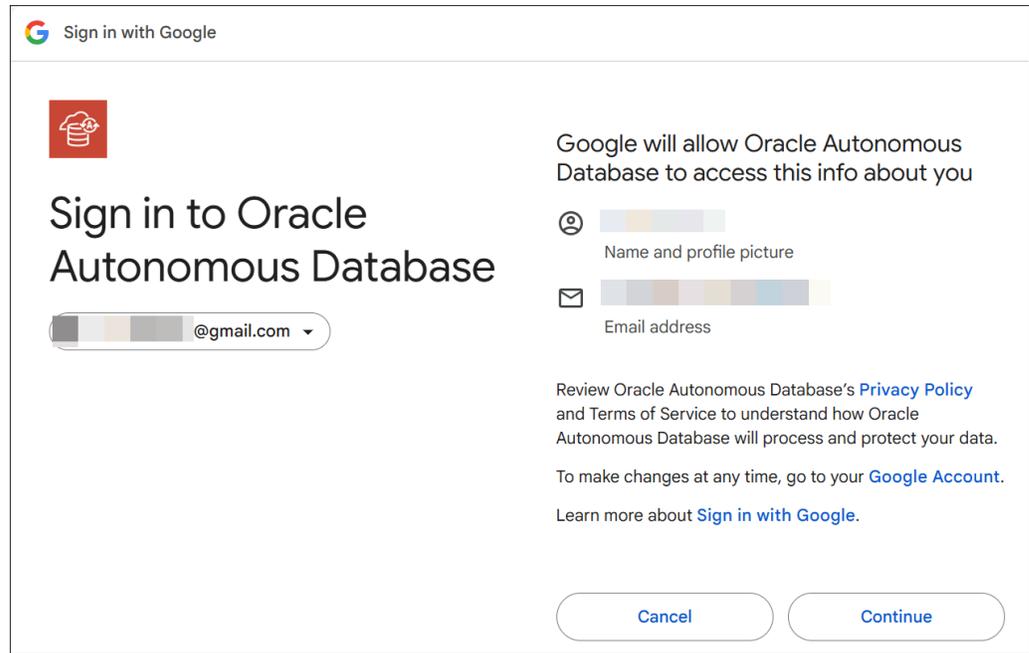
- Search for **Oracle Autonomous** in the search bar. You can view the **The Oracle Autonomous Database add-on for Google Sheets** in the search results.



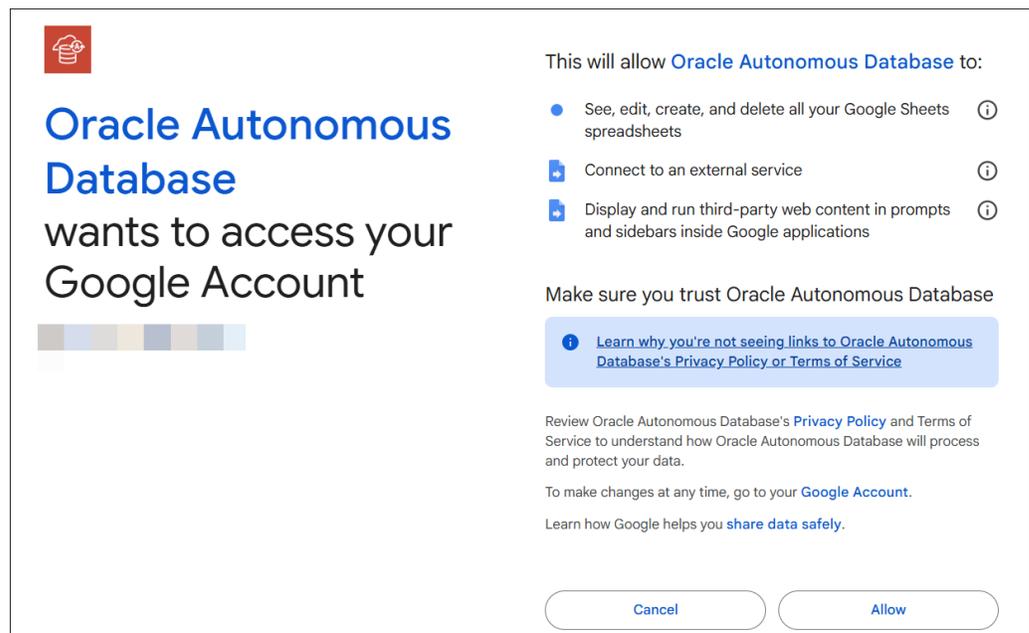
- Click **Install**.
- You will view a dialog to accept [terms of service](#) and [privacy policy](#). Click **CONTINUE** to agree to it.



- Upon authorization, you will be redirected to the Google Accounts page where you must select your Gmail account.
- Selecting your account opens a new window that ensures you review the [terms of service](#) and Oracle Autonomous Database's [privacy policy](#). Click **Continue** to proceed.

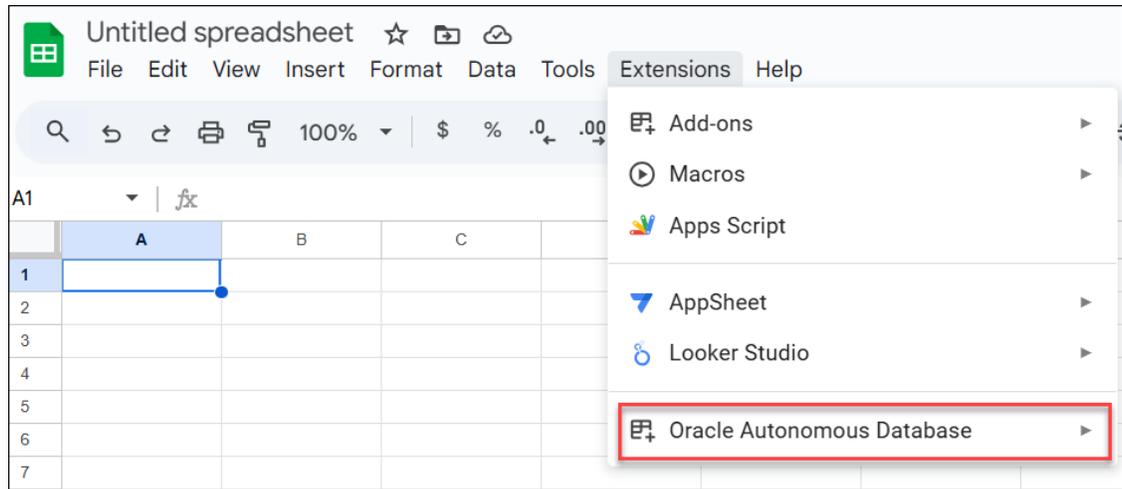


- Click **Allow** to grant permission for Oracle Autonomous Database to access your Google Account at this point.



You will view a message confirming the add-on loaded successfully.

A new **Oracle Autonomous Database** menu appears under **Extensions** menu in the Google Sheet.



You can now connect to the Autonomous Database to analyze and query the data.

Connecting to Autonomous Database

The Oracle Autonomous Database add-on for Google Sheets enables you to connect to multiple Autonomous Databases with a single add-on using the Connections feature. The add-on connects to Google Sheets by providing authentication to Google. Multiple users or databases can connect simultaneously to the add-on. However, only one connection can remain active.

The connection icon enables you to manage one or more database connections to databases in your Google Sheets. You can connect, edit, duplicate, and remove a connection.

You use the OAuth authentication and credentials to access the Oracle Autonomous Database for Google Sheets. The Add-on connects with the database using implicit and explicit types of connections.

Explicit Connection

You use the OAuth Client ID and OAuth Client Secret values to authenticate and authorize Google Sheets to use the Autonomous Database. Use this when you use **CODE** as the **Response Type** while downloading the connection file from Database Actions page.

Implicit Connection

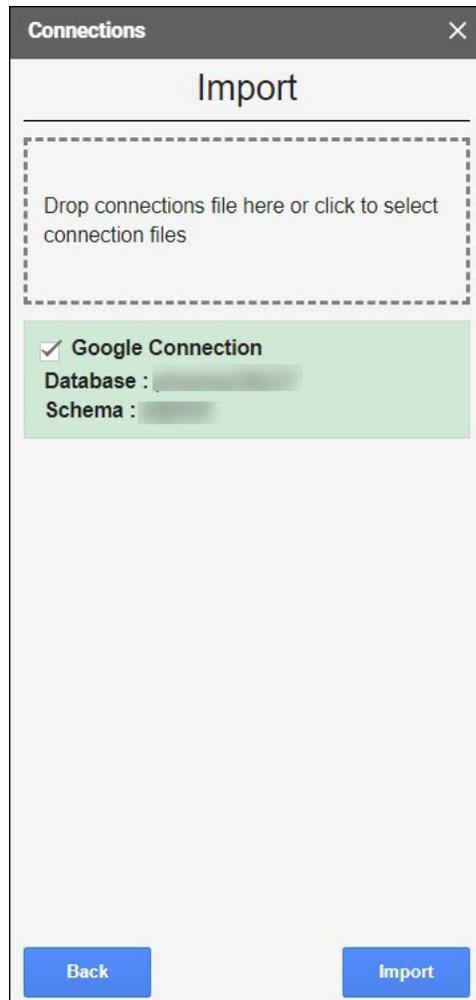
You will need an OAuth Client ID to implicitly access the Autonomous Database. Use this when you use **Token** as the **Response Type** while downloading the connection file from the Database Actions page.

Connection Management in Google Sheets

Import the connections file

Import an existing connection by clicking the **Import Connection** menu in the **Manage Connections** drop-down field.

1. Click the drop connection area and drag and drop the connection file saved in your local system to import the connection. You can import the **Connection File** you downloaded from the [Download Connection File](#) section.
2. Click and drop the connection file from your system to the drop area of the wizard. After the connection file loads, select the check box beside the connection file you want to import from the file.



3. Click **Import**. After you import the connection, you can view the connection in the list of connections.
4. Select the **three vertical dots** beside the connection name and click **Connect** to connect to the database.

Generate Client ID and Client Secret using UI

In this section you use the Web UI to obtain the `client_id` and `client_secret`.

You generate the client keys by accessing the Autonomous Database instance URL appending with `oauth/clients`.

For example, if your instance is " `https://<hostname>-<databasename>.adb.<region>.oraclecloudapps.com/ords/<schema Name>/_sdw`", you need to sign in to the link " `https://<hostname>-<databasename>.adb.<region>.oraclecloudapps.com/ords/<schema Name>/oauth/clients`". Be sure to include the trailing slash.

1. Sign in to Database Actions with the "`https://machinename.oraclecloudapps.com/ords/SchemaName/oauth/clients`" link. You can view an OAuth Clients page in the link "`https://localhost:port/ords/schemaName/_sdw/?nav=rest-workshop&rest-workshop=oauth-clients`".

- Click the **+Create OAuth Client** button to create a new client.

Create OAuth Client

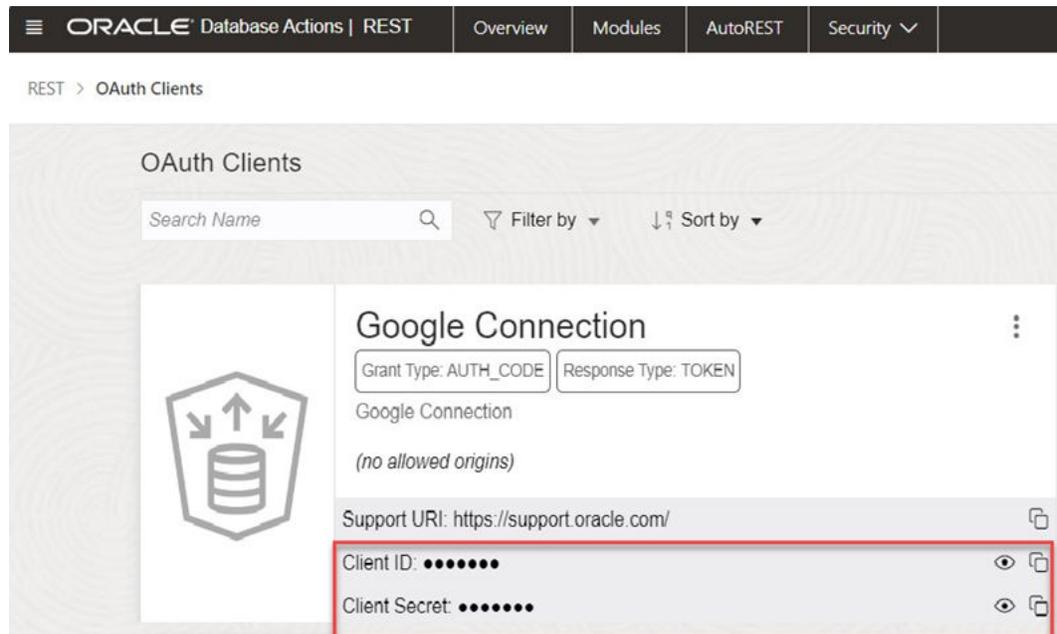
Client Definition *	Roles	Allowed Origins	Privileges
Owner ADPTEST		Grant type * AUTH_CODE	
Name * <input type="text"/>			
Description * <input type="text"/>			
Redirect URI * <input type="text" value="https://example.com/help/#/"/>			
Support URI * <input type="text" value="https://example.com/help/#/"/>			
Support Email * <input type="text" value="test@example.com"/>			
Logo <div style="border: 1px dashed gray; padding: 5px;"> <p>Choose a file</p> <p>Upload a JPG, BMP or SVG file lesser than 100KB in size to be used as your application logo, recommended size is 192x192 pixels,</p> </div>			
<input type="checkbox"/> Show code		<input type="button" value="Create"/> <input type="button" value="Cancel"/>	

- From the Grant type drop-down, select the type of client connection you want. You can select the following options:
 - AUTH_CODE:** Select this option for implicit connection. Use this response type when the autonomous database is in a private subnet or within a customer firewall.
 - IMPLICIT:** Select this option for explicit connection. This is the more secure method and is preferred to use if the Autonomous database has public access.
- Enter the following fields. The fields with an asterisk (*) are mandatory:
 - Name:** Name of the client.
 - Description:** Description of the purpose of the client.
 - Redirect URI:** web application deployment URL.
 - Support URI:** Enter the URI where end users can contact the client for support. Example: *https://script.google.com/*
 - Support Email:** Enter the email where end users can contact the client for support.
 - Logo:** Optionally, select an image from your local system to insert a logo for your new client.

Navigate to the **Roles** tab to select the roles of the client. This is not a mandatory field.

- Progress to the **Allowed Origins** tab. Specify and add the list of URL prefixes in the text field. This is not a mandatory field.

6. Progress to the **Privileges** tab to add any privilege. You are not required to have any privileges to create an OAuth Client.
7. Click **Create** to create the new OAuth Client. This registers the OAuth Client which you can view on the OAuth Clients page.

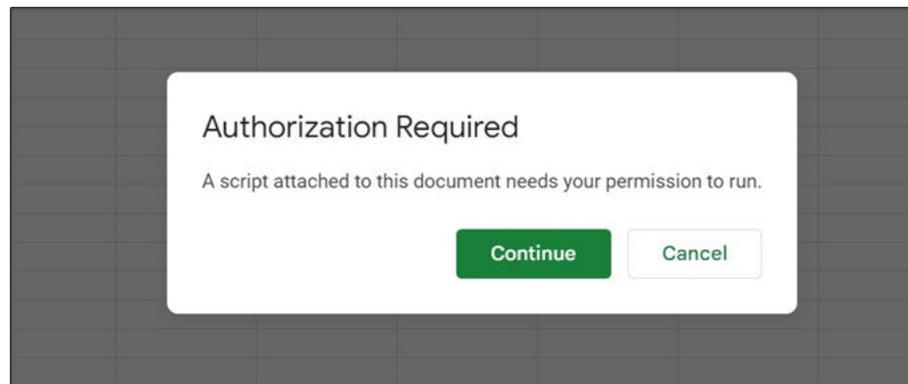


8. Click the show icon to view the Client ID and the Client Secret fields.

How do I connect manually?

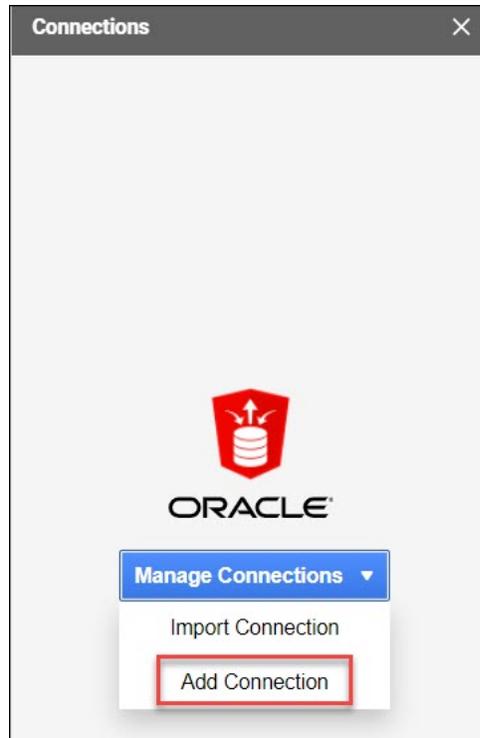
The following sections demonstrate how to connect using implicit and explicit connections. Google Sheets needs permission to access the Autonomous Database. You must first complete the authorization to connect to the autonomous database. The add-on requires one-time authentication for the setup.

1. On the Google Sheet, click **Oracle Autonomous Database** and select **Connections**. Selecting Connections requires one-time Google authentication.
 - Clicking **Connections** opens a pop-up window that asks your permission to run the authorization. Click **Continue**.



- You will now view a window that informs you that the application requests access to sensitive information in your Google account.

- Click **Advanced** and select the **Go to Untitled project (unsafe)** link. Selecting the link opens new window, ensuring you trust the application. Click **Allow** to continue. You have now completed the setup.
2. On the Connections wizard, click **Add Connection** from the **Manage Connections** drop-down menu to add a connection.



3. Selecting Add Connection opens an **Add Connection** wizard in the Connections wizard's connection list panel.

The screenshot shows a dialog box titled "Connections" with a close button (X) in the top right corner. The main heading is "Add Connection". Below this, there are several input fields and a radio button group:

- Connection Name:** A text input field containing "NewConnection".
- Autonomous Database URL:** An empty text input field.
- OAuth Client Grant Type:** Two radio buttons are present: "IMPLICIT" (unselected) and "AUTH_CODE" (selected). Below the radio buttons is a note: "(Select AUTH_CODE when the Autonomous Database has public access)".
- OAuth Client ID:** An empty text input field.
- OAuth Client Secret:** An empty text input field.
- Schema Name:** An empty text input field.

At the bottom of the dialog, there are two blue buttons: "Cancel" on the left and "Save" on the right.

4. Specify the following field values in the wizard:

Connection Name: Enter the connection's name—for example, *TestConnection*.

Autonomous Database URL: Enter the URL of the Autonomous Database you wish to connect to. For example, "*https://<hostname>-<databasename>.adb.<region>.oraclecloudapps.com/*"

In the OAuth Client Grant Type field, select one of the two options based on the type of connections you want. Refer to the [Generate Client ID and Client Secret using the UI](#) section.

This option varies with implicit and explicit connections.

Implicit: Select this option for implicit connection. Use this response type when the autonomous database is in a private subnet or within a customer firewall.

AUTH_CODE: Select this option for explicit connection. This is the more secure method and is preferred to use.

When you select the Implicit option, you can view the following fields:

The screenshot shows a dialog box titled "Connections" with a close button (X) in the top right corner. The main heading is "Add Connection". Below this, there are several sections:

- Connection Name:** A text input field containing "NewConnection".
- Autonomous Database URL:** An empty text input field.
- OAuth Client Grant Type:** A section containing two radio buttons: "IMPLICIT" (which is selected) and "AUTH_CODE". Below the radio buttons is a note: "(Select IMPLICIT when the Autonomous Database is in a private subnet or behind a firewall)".
- OAuth Client ID:** An empty text input field.
- Schema Name:** An empty text input field.

At the bottom of the dialog, there are two blue buttons: "Cancel" on the left and "Save" on the right. A red rectangular box highlights the "OAuth Client Grant Type" section and the "OAuth Client ID" and "Schema Name" fields.

OAuth Client ID: `client_id` you generate using the *Create New Client* wizard in the UI. Refer to the [Generate Client ID and Client Secret using the UI](#) section.

Schema Name: Specify the name of the schema.

When you select `AUTH_CODE`, you can view the following fields:

The screenshot shows a dialog box titled "Connections" with a close button (X) in the top right corner. The main heading is "Add Connection". Below this, there are several fields and options:

- Connection Name:** A text input field containing "NewConnection".
- Autonomous Database URL:** An empty text input field.
- OAuth Client Grant Type:** Two radio button options: "IMPLICIT" (unselected) and "AUTH_CODE" (selected). A red rectangular box highlights the "AUTH_CODE" option and its associated text: "(Select AUTH_CODE when the Autonomous Database has public access)".
- OAuth Client ID:** An empty text input field.
- OAuth Client Secret:** An empty text input field.
- Schema Name:** An empty text input field.

At the bottom of the dialog, there are two buttons: "Cancel" on the left and "Save" on the right.

OAuth Client ID: `client_id` you generate using the Create New Client wizard in the UI. Refer to the *Generate Client ID and Client Secret using the UI* section.

OAuth Client Secret: `client_secret` you generate using the Create New Client wizard in the UI. Refer to the *Generate Client ID and Client Secret using the UI* section.

Schema Name: Specify the name of the schema.

Click **Save**.

After you click **Save**, you can view the new connection in the connection list panel. The connection list displays the connection's name, the schema's name, and the OAuth type you grant. However, it is still in a disconnected state.

5. Click the three vertical dots beside the connection name and perform the following operations:

Connect: Select **Connect** to the Autonomous Database and change the connection status to active. Selecting **Connect** opens the sign-in page of the Autonomous database. After you log in, you will view a page that shows that database access has been granted to you. Close the window and return to Google Sheets. You will now see that the connection is active.

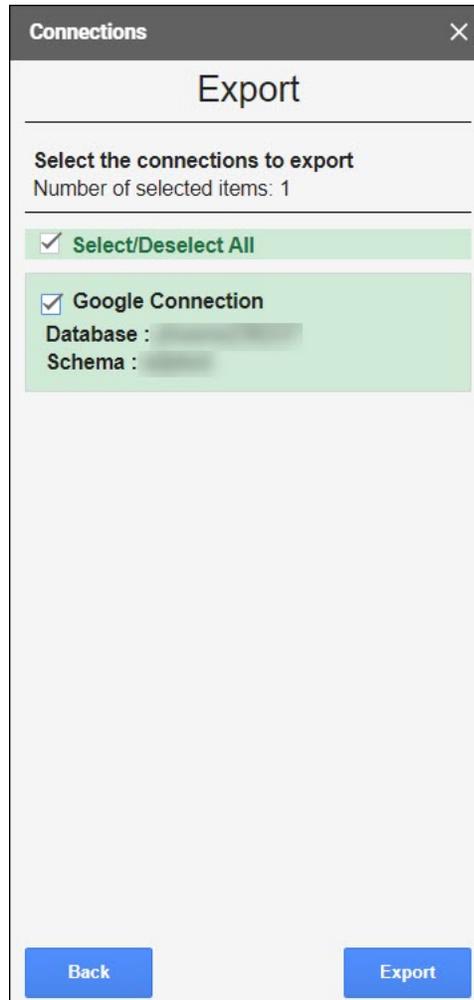
Edit: Select **Edit** to update any value of the connection. Click **Save** to update the edited values.

Duplicate: Select **Duplicate** to create a duplicate connection.

Remove: Select **Remove** to remove the connection from the connection list.

Exporting Connections

1. Click **Export Connection** from the **Manage Connections** drop-down menu to export the selected connection.
2. Select the connection you want to export, and click **Export**.



3. Click **Export**.
4. The exported connection downloads in your local system. The connection file is saved as *spreadsheet_addin_connections.json*.

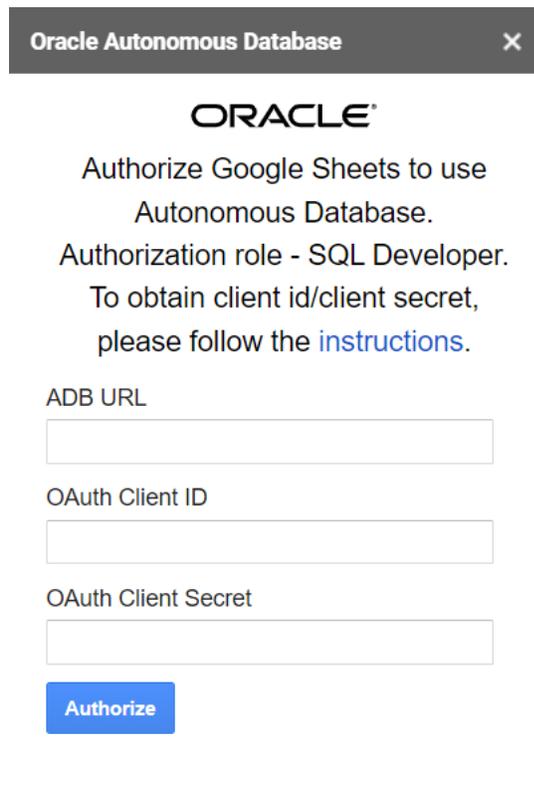
Authorize Google Sheets to use Autonomous Database

After your identity is determined using OAuth authentication, Google Sheets needs permission to access the Autonomous Database.

The `client_id` and `client_secret` values you generate during OAuth authentication are used for authorization.

1. Click on the **Oracle Autonomous Database** menu in the Google Sheet you are working on and select **Register**. This requires one-time Google authentication.

2. Clicking Register opens a pop-up window that asks your permission to run the authorization. Click **Continue**. Selecting Continue will redirect you to the Google Accounts page, where you must select your Gmail account.
3. You will now view a window that informs you that the application requests access to sensitive information in your Google account. Click Advanced and select the Go to Untitled project (unsafe) link.
4. Selecting the link opens a new window, ensuring you trust the application. Click **Allow** to continue.
5. You have now completed the setup. Select **Register** from the Autonomous Database menu in the Google sheet. This opens an Oracle Autonomous Database wizard in the Google sheet. Specify the following fields:
 - **ADB URL:** Enter the ADB URL. For example, "`https://<hostname>-<databasename>.adb.<region>.oraclecloudapps.com/ords/<Schema Name>`".
 - **OAuth Client ID:** client_id you generate during authentication.
 - **OAuth Client Secret:** client_secret you generate during authentication. Refer to the Create Connections with the Google spreadsheet section for more details.



The screenshot shows a window titled "Oracle Autonomous Database" with a close button (X) in the top right corner. The window contains the Oracle logo and the following text: "Authorize Google Sheets to use Autonomous Database. Authorization role - SQL Developer. To obtain client id/client secret, please follow the [instructions](#)." Below the text are three input fields: "ADB URL", "OAuth Client ID", and "OAuth Client Secret". At the bottom of the window is a blue button labeled "Authorize".

6. Select **Authorize**.

After successfully authorizing the credentials, you can view **Connections**, **Natural Language Query**, **Direct SQL**, **Data Analysis**, **Table Hyperlinks**, **Settings**, and **About Autonomous Database**, under **Oracle Autonomous Database**.

Natural Language in Google Sheets

You can use Natural Language Query to query the Oracle Autonomous AI Database using the **Natural Language** menu in the **Oracle Autonomous Database for Google Sheets** "add-on".

Prerequisites

- Before you use the Natural Language menu in the add-on, you must [perform prerequisites](#) to use DBMS_CLOUD_AI to configure AI profiles.
- [Create and Set an AI Profile](#)

After your AI Profile is ready, the Data Studio tool uses these profiles in the [Data Studio Settings](#) menu to configure access to a Large Language Model (LLM) and to setup for generating, running, and explaining SQL based on natural language prompts. This also enables chatting with the LLM.

Note

You can run Natural Language Queries on Tables and not Analytic Views.

Generate SQL Queries from Natural Language on Tables

Using natural language to interact with your database data is now achievable with Oracle Autonomous Database add-on for Google Sheets.

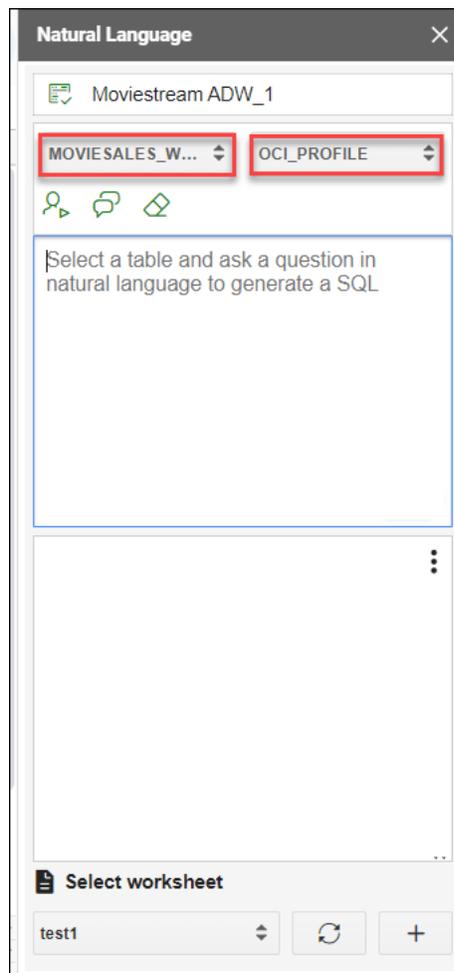
This means you can use natural language, for example, plain English, to query the database. You can provide a natural language prompt instead of SQL code to interact with your data. When you select the **Generate SQL from natural query** icon, the add-on converts natural language to SQL.

To run a natural query using the add-on:

1. On the Google Sheet, select the menu item **Oracle Autonomous Database**.
2. Select **Natural Language**. Selecting **Natural Language** opens a **Natural Language** wizard.
3. On **Natural Language** wizard, select the **Table** on which you perform the query from the drop-down and the **AI profile** you want to use from the drop-down.

Note

This is the same AI profile you configure in the [Data Studio Settings](#).



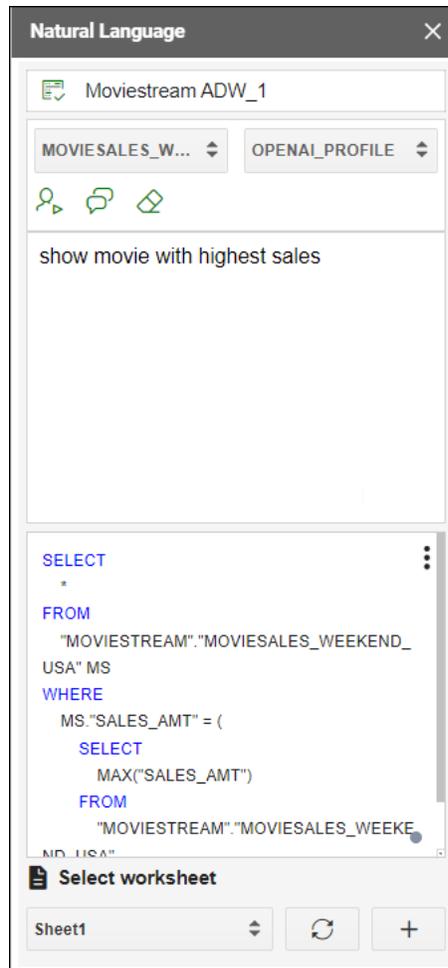
In this example, MOVIESALES_WEEKEND_USA is the name of the **Table** and OCI_PROFILE is the **AI Profile**.

- Let's say you want details of the product with highest sales. Enter the following natural language query in the query display area:

```
show movie with highest sales
```

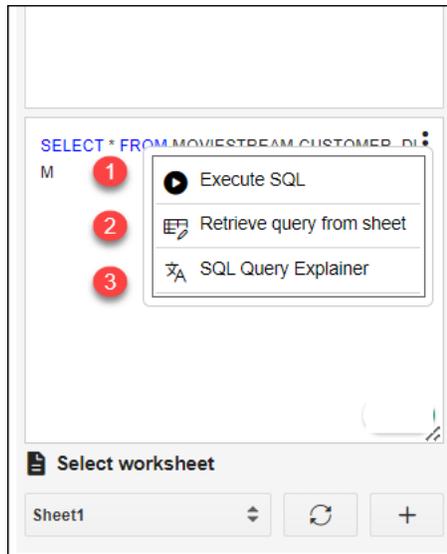
- Select **Generate SQL from natural query** to produce the equivalent SQL query in the bottom query display area. You will view the following code in the bottom SQL code area.

```
SELECT
  *
FROM
  "MOVIESTREAM"."MOVIESALES_WEEKEND_USA" MS
WHERE
  MS."SALES_AMT" = (
    SELECT
      MAX("SALES_AMT")
    FROM
      "MOVIESTREAM"."MOVIESALES_WEEKEND_USA"
  )
```



6. Selecting the **Generate SQL from natural query** icon also displays the result of the natural language query in the Google Sheet. You can click the **+** sign beside the **Select worksheet** drop-down to display the results in a new worksheet.
7. Click **Clear query content** to clear the content displayed on the natural language query area and the SQL code query area.

Actions you can perform from the SQL code area:



- **Execute SQL:** Runs SQL code and displays the query result in Google Sheet.
- **Retrieve query from Sheet:**
You can retrieve SQL query displayed in the Google Sheet to the SQL code area in the Natural Query.

This feature is to retrieve the latest data from the database for a future date. For example, if you want to receive the latest data from the table the next day, the connection may be inactive, but you can still connect to the instance, **retrieve query from Sheet** and **select Execute SQL** to display the latest query results in the Google Sheet.

The screenshot displays the Oracle Autonomous Database for Google Sheets interface. The main window shows a spreadsheet with a 'Query Info' section containing 'SQL-Query' and 'Timestamp'. The 'SQL-Query' field contains the query: `SELECT * FROM MOVIESTREAM.CUSTOMER_DIM`. A red box highlights this query, and a red arrow labeled 'Retrieve SQL Query' points to the 'Natural Language' panel on the right. The 'Natural Language' panel shows a dropdown menu with 'CUSTOMER_DIM' selected and a text input field containing the same SQL query. Below the input field is a 'Select worksheet' section with 'Sheet1' selected.

CUSTOMER_ID	CITY	STATE_PROVINCE
1386913	Bur Hacaba, Bay	Bay, SO
1256611	La Tinguiña, Ica	Ica, PE
1138009	Táplánszentkereszt	Vas, HU
1145231	Bükkfűrdő, Vas	Vas, HU
1144286	Bô, Vas	Vas, HU
1144245	Kám, Vas	Vas, HU
1142128	Szombathely, Vas	Vas, HU
1226956	Aba, Abia	Abia, NG
1373356	Aceh Utara, Aceh	Aceh, ID
1156093	Wick St Lawrence, Avon	Avon, GB
1161784	Uphill, Avon	Avon, GB
1162269	Bath, Avon	Avon, GB
1165339	Bristol, Avon	Avon, GB
1147574	Peasedown St John	Avon, GB

- **SQL Query Explainer:** Explains SQL code in Natural language form. Select this icon to translate SQL queries to natural language that is understood by you. The natural language query is displayed in the natural language query display area of the wizard.

Ask Questions with Chats

Chats option in the Oracle Autonomous Database for Google Sheets refer to an interactive conversation between you and the add-on where the add-on uses natural language to query or interact with the Autonomous AI Database.

The data we use in this example is of a company called Oracle MovieStream to analyze movie sales data.

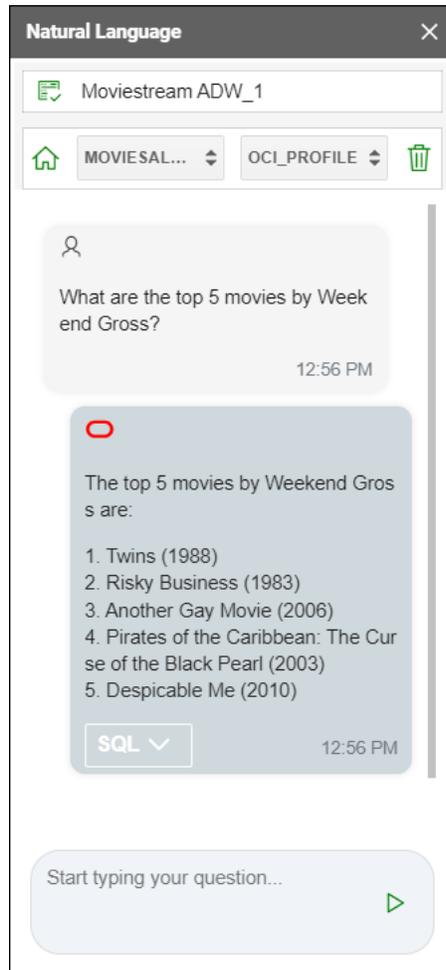
The Chats displays recommendations for the default table you select.

Let's find out the top 5 streamed movies in the MovieStream company by weekend gross:

1. Enter the text in the **Start typing your question..** text field.

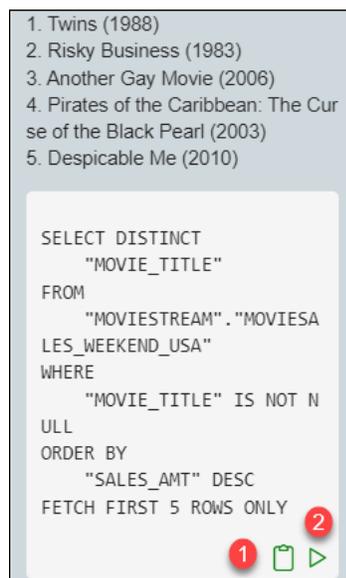
What are the top 5 movies by Weekend Gross?

2. Click **Enter**.
3. The Chat displays top 5 movies by Weekend Gross.



It also displays equivalent SQL code of the result.

Click **SQL** to expand the SQL.



Click **Copy to clipboard** to copy the SQL.

Click **Run SQL** to run the generated SQL query and display the query result in Google Sheet.

The **Chats** option remembers the context of previous chat history.

For example, if you enter `now show me top 10` in the text field.

It displays the top 10 movies by Weekend Gross. It remembers which metric top 10 to fetch without us having to type the whole thing.

Select **New Chat** for deleting the present conversation.

Select **Home** to go back to the main home page.

Note

LLMs are remarkable at inferring intent from the human language and they are getting better all the time; however, they are not perfect! It is very important to verify the results.

Data Analysis in Google Sheets

Selecting **Data Analysis** opens an Oracle Autonomous Database wizard in the Google sheet.

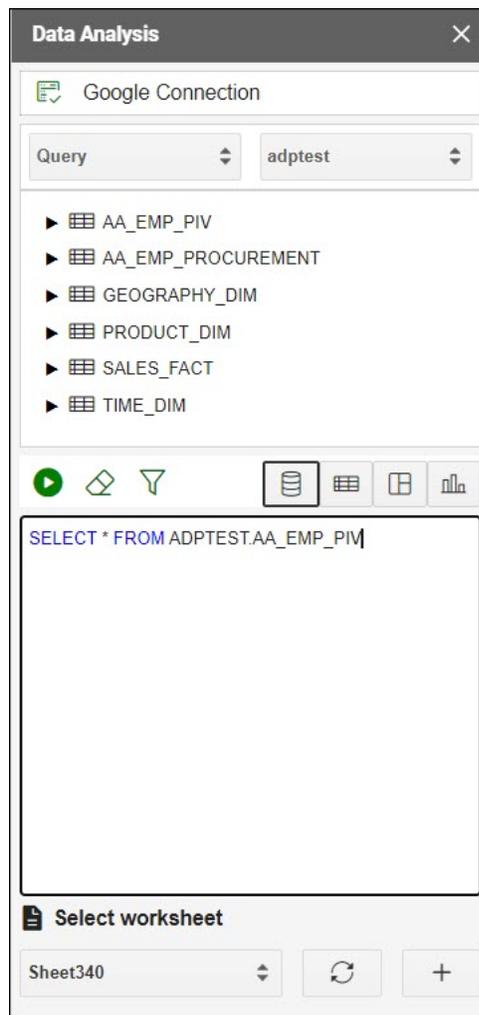
The add-on enables you to receive a copy of data from the Autonomous Database to the Google sheet. You can query an existing Analytic View and run SQL Query using the Oracle Autonomous Database Wizard.

You can retrieve the Analytic View and manipulate the query according to your requirements to visualize the result data in the worksheet. You can search for the Analytic View and select measures, hierarchies, and levels from the query. You can also add filters and calculated measures to the query and view the result in the sheet. By default, the data is retrieved in tabular format.

You can run custom queries. The add-on enables you to apply a filter to the query results. The add-on lets you to view query results that can be customized with selected columns using a faceted filter.

To run a custom query using the add-on:

1. On the Google Sheet, select the menu item **Oracle Autonomous Database**.
2. Select **Data Analysis**. Selecting Data Analysis opens a Data Analysis wizard. On the Data Analysis wizard, select **Query** from the drop-down and the **schema** you want to use from the drop-down.

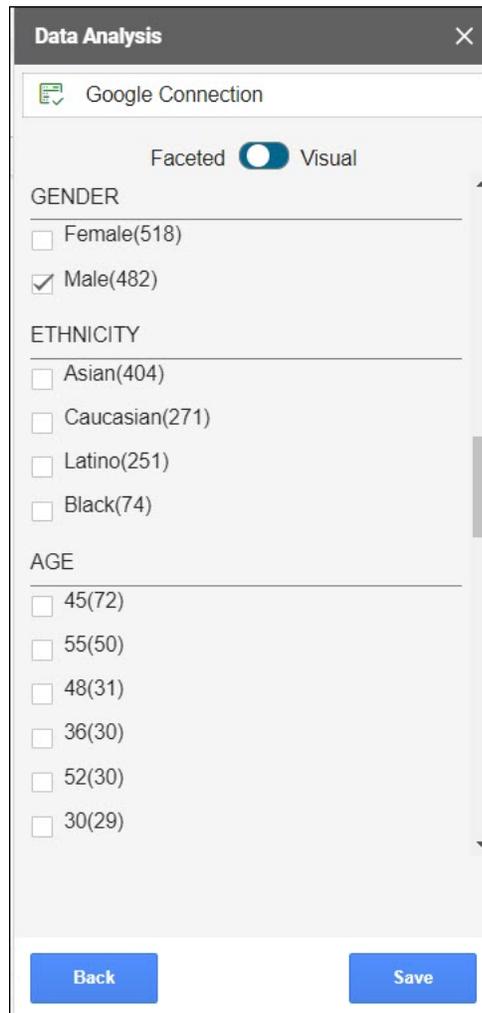


3. You will view the default query in the query editor area. You can select any of the four modes to visualize the results of the SQL query report you generate:

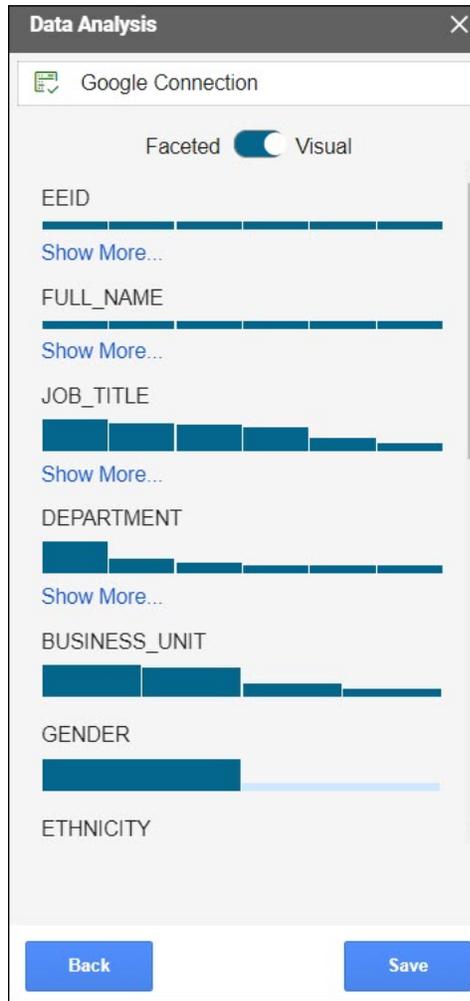


- **Base Query:** This type of view is by default. The query written in the SQL editor is the Base Query.
- **Table:** You can view the SQL results in tabular form. By selecting this view, a column drop zone appears, enabling you to drag and drop selected columns from the Table browser. Moving the selected columns in the drop zone allows you to view only those columns in the Result data generated in the worksheet. Select the cross mark beside the Column name to remove it from the drop zone.
- **Pivot:** You can view the SQL query results in pivot format. By selecting this format, an X and Y drop zone appears where you can drag and drop the selected columns from the Tables browser to the drop area.
- **Chart:** You can view *Area Chart*, *Bar Chart*, *Line Chart*, or *Pie Chart* when you select this option.. The mappings displayed when you select one of the options are as follows:

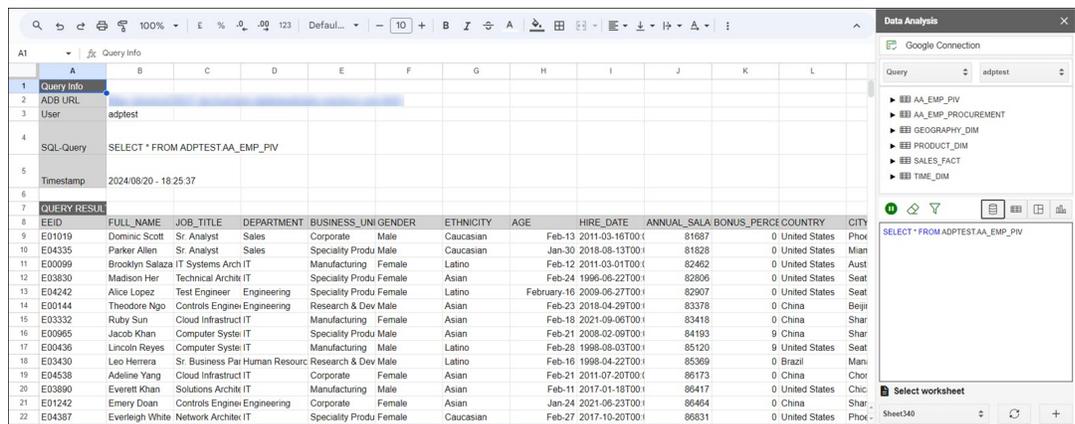
- **Orientation:** Choose between horizontal and vertical orientation types from the drop-down list.
 - **X axis label and Y axis label:** Optionally enter labels for X axis and Y axis.
4. Click the **funnel icon** (Faceted Filter) to add filters to the result. The wizard generates a filter for each value in the column retrieved from the query result. You can filter different columns on the faceted filter panel and view the results in the worksheet to view only the data you wish to view. For example, to view the customer reports by Gender, click the **faceted filter** and select **Male** under Gender.



Use the Visual facet: Use the visual indicator to graphically represent the faceted filter.



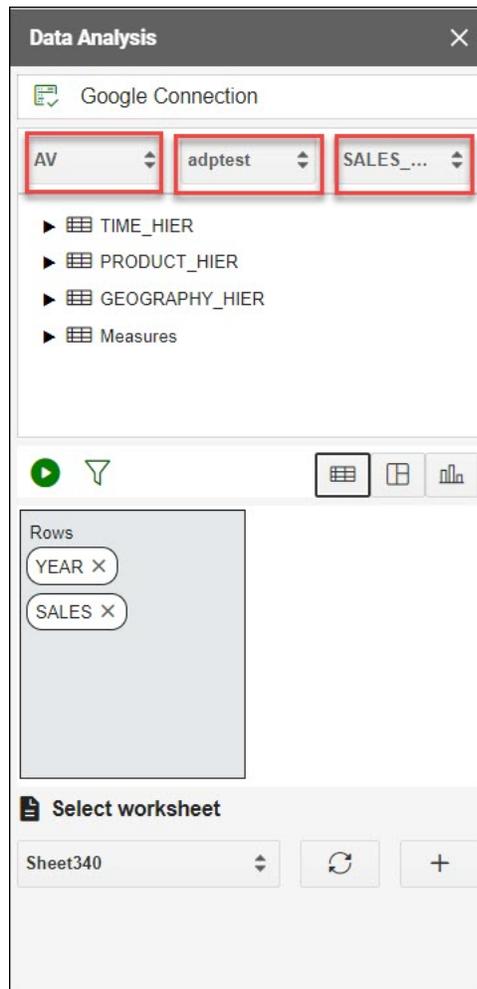
5. Select **Run** to generate the results of the custom query in the worksheet.



To query an Analytic View and explore the **Data Analysis** menu in the Google Sheets:

1. Select the menu item **Oracle Autonomous Database > Data Analysis** on the Google Sheet. This opens a **Data Analysis** wizard in the Google task pane.

- Select **AV** from the **AV** or **Query** drop-down, select a schema you can access from the schema drop-down, and the AV from the available Analytic Views.

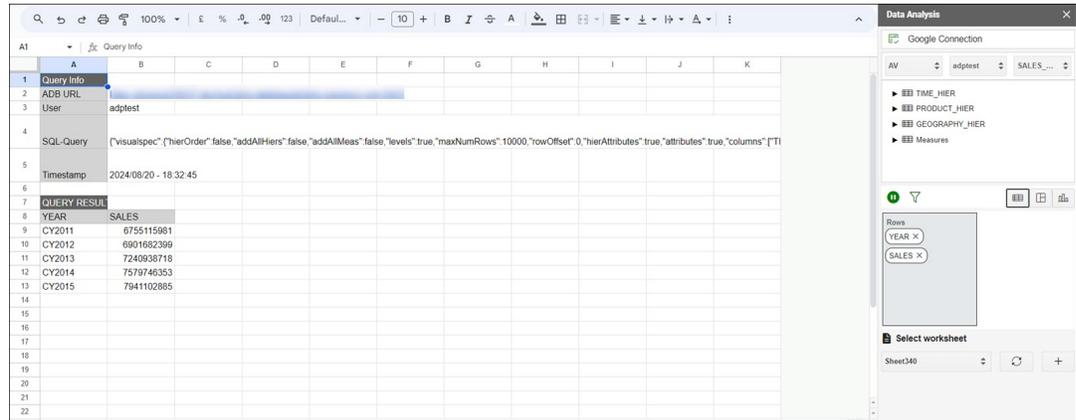


- You can select any of the three modes to visualize the results of the AV query you generate:
 - Table:** You can view the AV query results in tabular form. By selecting this view, a column drop zone appears, enabling you to drag and drop selected columns from the Table browser. Moving the selected columns in the drop zone allows you to view only those columns in the Result data generated in the worksheet. Select the cross mark beside the Column name to remove it from the drop zone.
 - Pivot:** You can view the SQL query results in pivot format. By selecting this format, an X and Y drop zone appears where you can drag and drop the selected columns from the Tables browser to the drop area.
 - Chart:** You can view the results of the AV query in chart format. By selecting this format, an X and Y drop zone appears where you can drag and drop the chosen hierarchies and measures from the AV browser to the drop area.

Note

You are allowed to drop measures in the Y-axis.

4. Click the **funnel** icon to view the Faceted and Visual Filter list. The wizard generates a filter for each value in the column retrieved from the query result. You can filter different columns on the faceted filter panel and view the results in the worksheet to view only the data you wish to view.
5. Click **Save**.
6. Click **Run** to view the results in the worksheet you select.



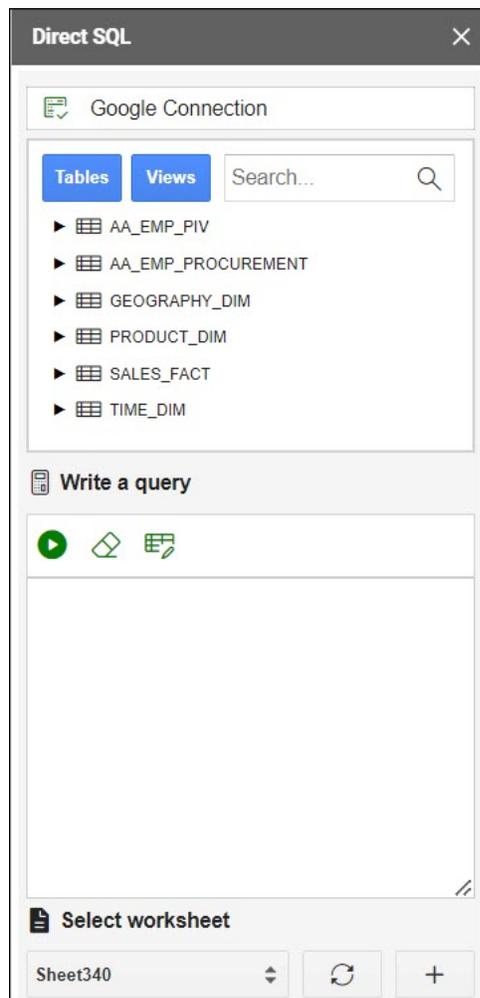
Run Direct SQL Queries

The Oracle Autonomous Database add-on for Google Sheets lets you run SQL queries to work with your data in a Google Sheet. With the add-on, you can type your SQL code in the SQL editor area and click **Run** to run the command.

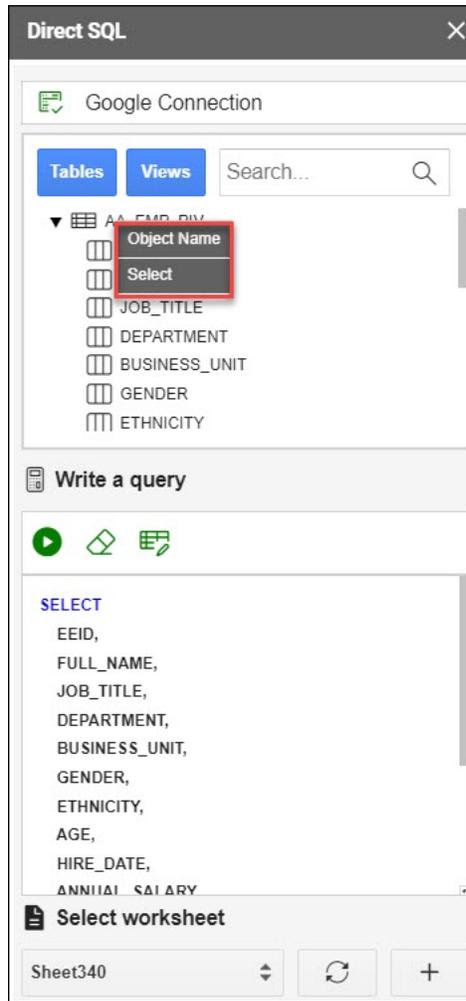
The add-on loads the result in the Google Sheet. The time taken to load the results depends on the number of records and the complexity of the query.

To run a query using the add-on, open Google Sheets and a blank workbook.

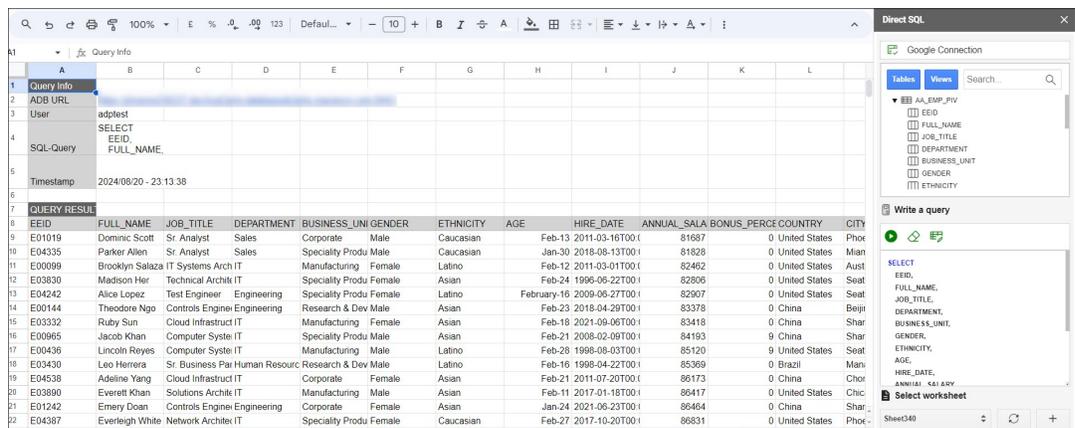
1. In the Google Sheet, select the menu item **Oracle Autonomous Database**.
2. Select **Direct SQL** to type and run the SQL command.
3. The Oracle Autonomous Database wizard opens Tables and Views icons and a search field beside it.



4. Select **Table** to view all the tables in the database. Perform the same operations for Views.
5. You can right-click on the table whose data you want to query and choose **Select** to view all the table's columns. The column names will be displayed in the **Write a Query** section. You can click on the table and view individual columns as well.



6. Click **Run** to run the query and display the results in the worksheet. You can click the **+** sign beside the **Select worksheet** drop-down to display the results in a new worksheet.
7. The worksheet also displays the timestamp, the user who creates and runs the query, the ADB URL, and the SQL Query.



Read and Access Data Using Table Hyperlinks in Google Sheets

Table Hyperlinks in Oracle Autonomous Database provide secure, preauthenticated URLs that allow read-only access to data stored in tables, views, or the result of SQL queries.

They enable easy sharing and retrieval of database data without requiring additional authentication beyond the hyperlink itself.

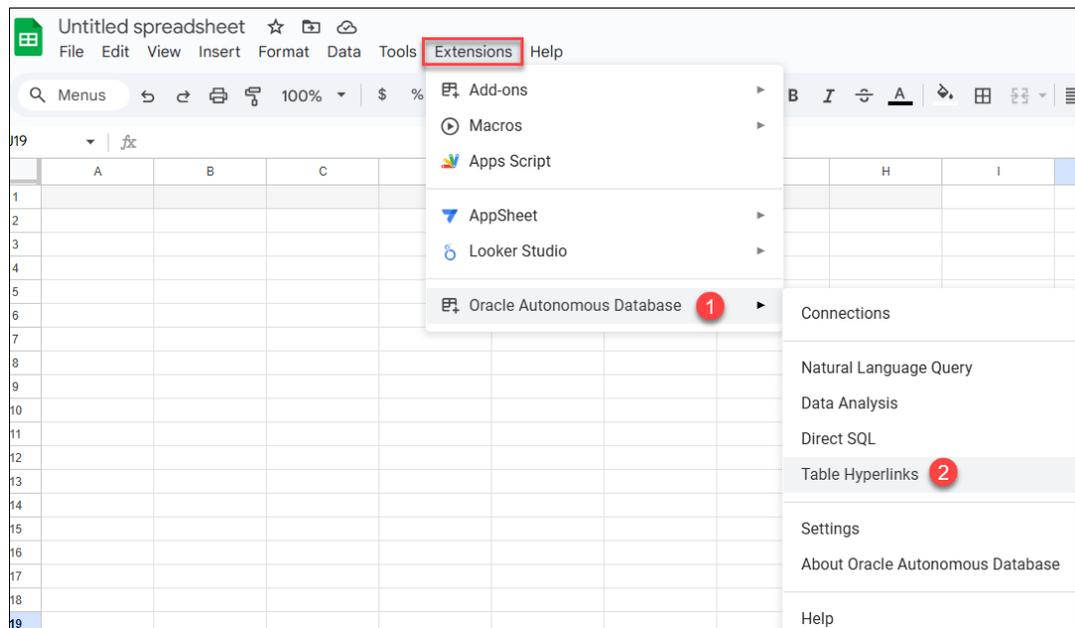
You can use **Table Hyperlinks** with the add-on.

For more information of Table Hyperlinks, refer to the [Use Table Hyperlinks for Read Only Data Access on Autonomous Database](#) Chapter.

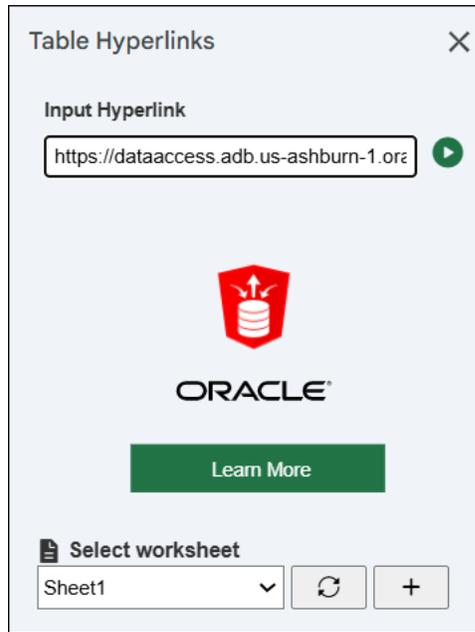
Use **Table Hyperlinks** to read and access data and view the data in a tabular format.

To return data from the add-in using Table Hyperlinks:

1. On the Google sheet, click **Oracle Autonomous Database** from the **Extensions** menu and select **Table Hyperlinks**.



2. Selecting Table Hyperlinks opens a **Table Hyperlinks** panel.



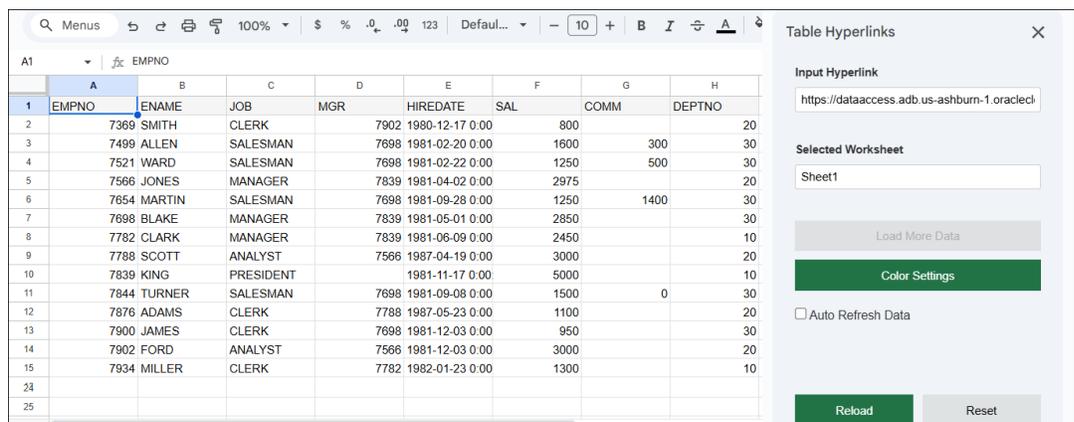
- On the **Input Hyperlink** field, enter the input Table Hyperlink the producer shares with you. A Producer generates a PAR URL in their database and provides it to you. For example, *https://dataaccess.adb.us-ashburn-1.oraclecloudapps.com/adb/p/F5Sn...* You can create Table Hyperlinks in Oracle Autonomous Database Serverless using the DBMS_DATA_ACCESS PL/SQL package and share them with the consumers.

See [DBMS_DATA_ACCESS Package](#) for more information on this topic.

Select Worksheet: Enter the name of the sheet where you want to display the results. For example, *Sheet1*.

You can click the + sign beside the Select worksheet drop-down to display the results in a new worksheet.

- Click the **Play** button to load the Autonomous Data table in the worksheet. The result data gets retrieved in the worksheet.



After selecting the play icon, along with the results retrieved in the worksheet, you can also see the following generated fields:

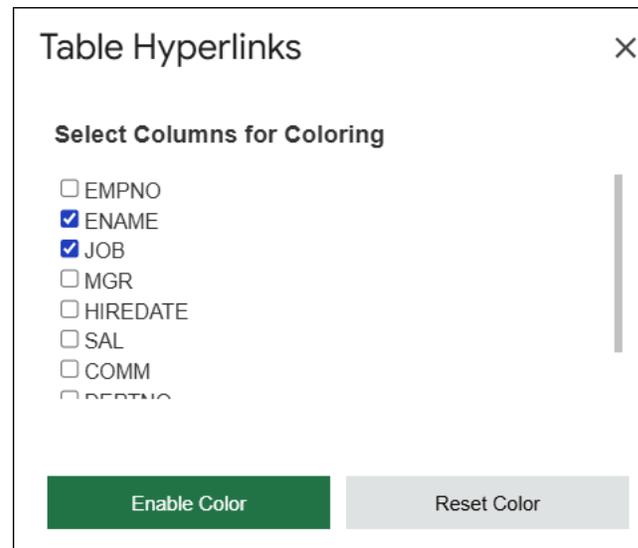
- Load More Data:**

Select **Load More Data** to retrieve the next 10000 records from the database. The button is enabled when there are additional records remaining to be retrieved from the database.

- **Color Settings:**

Select this icon to enable color in the column from a predefined pool of colors. If the feature runs out of colors, the pool is refilled with the same colors for the new value keys.

Click the icons you want to add color to. In this example, select *ENAME* and *JOB*.



- Select **Enable Color** to color the selected columns in the data.

The image shows a Google Sheet with a table of employee data. The columns are EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The ENAME and JOB columns are color-coded. The 'Table Hyperlinks' dialog box is open on the right, showing the 'Color Settings' section with the 'Auto Refresh Data' checkbox checked. The 'Auto Refresh Timer' is set to 5 minutes. The 'Reload' button is highlighted in green.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK		1980-12-17 0:00	800		20
7499	ALLEN	SALESMAN		1981-02-20 0:00	1600	300	30
7521	WARD	SALESMAN		1981-02-22 0:00	1250	500	30
7566	JONES	MANAGER		1981-04-02 0:00	2975		20
7654	MARTIN	SALESMAN		1981-09-28 0:00	1250	1400	30
7698	BLAKE	MANAGER		1981-05-01 0:00	2850		30
7782	CLARK	MANAGER		1981-06-09 0:00	2450		10
7788	SCOTT	ANALYST		1987-04-19 0:00	3000		20
7839	KING	PRESIDENT		1981-11-17 0:00	5000		10
7844	TURNER	SALESMAN		1981-09-08 0:00	1500	0	30
7876	ADAMS	CLERK		1987-05-23 0:00	1100		20
7900	JAMES	CLERK		1981-12-03 0:00	950		20
7902	FORD	ANALYST		1981-12-03 0:00	3000		20
7934	MILLER	CLERK		1982-01-23 0:00	1300		10

You can select **Reset Color** to reset your current selection, remove colors from all the selected columns and choose different columns you wish to enable color to.

- **Auto Refresh Data:**

Enable this option to automatically update data at regular intervals with the latest data from its source database. This automatic refresh keeps the tabular data in worksheet in sync with the current data within a specific refresh frequency.

You can select the frequency from the frequency slider where the selection varies from a range of 5 minutes to 1 day.

- **Reload:** Select **Reload** to reload the preloaded table hyperlink data.
- **Reset:** Select **Reset** to clear all data from the sheet, reset your selection of all the current options and enter the **Input hyperlink** again.

About Autonomous Database menu

Use this option to view details about the add-in.

The **About Oracle Autonomous Database** menu from **Oracle Autonomous Database** displays if the add-on is connected to server, the ORDS version, the Add-in version, the ORDS Schema version, the database major and minor version, the ADB URL and the Schema.

Share or Publish

Once you generate the query results in the Google Sheet, you can share it with other users. With sharing, creates a copy of the worksheet and sends it with the design tools hidden and worksheet protection turned on.

The recommended steps to take before you publish are:

1. Review and inspect to remove personal or sensitive information.
2. Save the source version of the worksheet. Consider adding a file name suffix of –src for the source worksheet. Then, remove the suffix in the distributed copy. Once you are ready to distribute to the users, click **Share**.
 - In the Share window that appears, add the user email IDs with whom you want to share the Sheets and to whom you want to provide permissions for accessing the Sheets.
 - You can select the permission of the users from the drop-down. Select **Editors** if you want the user to share the worksheet. **Viewers** and **commenters** can see the option to download, print and copy but not share the sheets.
 - Select **Notify people** check-box to notify the users of the share.
 - Under General access, select **Restricted** from the drop-down to share it with people who have access to the link. You could also share it with people who do not have access by selecting **Anyone with the link** from the drop-down.

Oracle Autonomous Database for Google Sheets Terms of Service

You can view the following Terms of Service here:

It is in addition to the agreement described in [Oracle Technology Network Developer License Terms](#):

By launching or using Oracle Autonomous Database for Google Sheets or by selecting the "I accept" button or box (or the equivalent), You agree to be bound by the [Oracle Technology Network Developer License](#), and that your use of Oracle Autonomous Database for Google Sheets is subject to the [Google Workspace Marketplace Terms of Service](#), (collectively, the "Terms of Use"). If You are agreeing to the Terms of Use on behalf of a company or other legal entity, You represent that You have the authority to bind it to the Terms of Use and, in such event, "You" and "Your" as used in the Terms of Use shall refer to such entity. If You are not willing to be bound by the Terms of Use, do not select the "I accept" button or box (or the equivalent) and do not launch or access Oracle Autonomous Database for Google Sheets.

Oracle Autonomous Database for Google Sheets Privacy Policy Details

Effective Date: 10/4/2024

Oracle Autonomous Database add-on for Google Sheets facilitates seamless data integration between Google Sheets and Oracle Autonomous Database. This Privacy Policy topic explains how we access, use, share, protect, retain, and delete your Google user data when you use our add-on.

Data use by **Oracle Autonomous Database add-on for Google Sheets** complies with Oracle's [Privacy Policy](#).

1. Google User Data We Access

- **Google Sheets Data:** The add-on accesses the content of your Google Sheets to import data from your Oracle Autonomous Database.
- **User Profile Data:** User Profile Data, which includes personal information linked to a Google account, is not accessed or used by our add-on, as authentication is done with Oracle credentials. The data our application handles is limited to Google Sheets content.

2. How We Use Google User Data

- **Data Integration:** Your Google Sheets data is used to enable functionalities like importing data from Oracle Autonomous Database into Google Sheets.
- **Authentication:** Authentication is performed using your Oracle credentials. We do not access or use any Google authentication data or Basic Profile Information.

3. Data Sharing and Disclosure

- **No Third-Party Sharing:** We do not share, sell, or disclose your Google user data to any third parties.
- **Legal Obligations:** We may disclose your information if required by law or in response to valid legal processes.

4. Compliance with Google API Services User Data Policy

- **Adherence to Limited Use Requirements:** To the extent the **Oracle Autonomous Database add-on for Google Sheets** uses and/or transfers information received solely from Google APIs to any other app, such use and transfer will adhere to [Google API Services User Data Policy](#), including the Limited Use requirements.

5. Data Protection Measures

- **Secure Transmission:** All data transferred between Google Sheets and Oracle Autonomous Database is encrypted using industry-standard protocols as detailed in our [Security and Authentication for Autonomous Database](#).
- **Access Controls:** Access to your data within the Add-on is restricted to authorized processes, ensuring that only necessary operations are performed.
- **Oracle's Security Standards:** We adhere to Oracle's comprehensive security practices, including regular security assessments and compliance with global security standards.

6. Data Retention and Deletion

- **No Data Storage:** We do not store your Google Sheets data on our servers. Data is accessed in real-time solely for the purpose of facilitating the integration between Google Sheets and Oracle Autonomous Database.

- **User Control:** You can revoke the Add-on's access to your Google Sheets data at any time through your Google account settings.
- **Deletion Requests:** Since we do not retain your data, deletion requests are generally unnecessary. However, if you have concerns, please contact us at adb_googlesheets_dev_ww_grp@oracle.com.

7. User Choices and Controls

- **Manage Permissions:** You can manage or revoke the add-on's permissions through your Google account settings at any time.
- **Uninstallation:** To discontinue using the add-on, you can uninstall it from Google Sheets and revoke its access permissions.

8. Changes to This Privacy Policy

We may update this Privacy Policy periodically. Changes will be reflected by the `Effective Date` at the top of this document. We encourage you to review this policy regularly to stay informed about how we protect your information.

9. Contact Information

If you have any questions or concerns regarding this Privacy Policy, please contact us at:

- **Email:** adb_googlesheets_dev_ww_grp@oracle.com
- **Address:**
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA

Oracle Autonomous Database for Google Sheets Support

Welcome to the support page for the **Oracle Autonomous Database for Google Sheets**. This resource is designed to assist you with any issues or questions you may have while using the add-on.

Support for the Add-on

If you encounter any problems specifically related to the Oracle Autonomous Database Add-on, please reach out to our development team at adb_googlesheets_dev_ww_grp@oracle.com

Our team is dedicated to providing prompt assistance to ensure you have a seamless experience.

Support for Oracle Database Issues

For issues related to Oracle Database that are not specific to the Google Sheets add-on:

- **Contact Your Oracle Administrator:** Your organization's Oracle administrator can help with database access, permissions, and other database-related concerns.
- **Oracle Support Portal:** File a support ticket for further assistance at [My Oracle Support](#).

The Data Marketplace Tool

Data Marketplace provides an innovative platform to access and share datasets effortlessly.

The Data Marketplace is part of the Data Studio suite of tools that enables data scientists and business users to load and query datasets for their organizational needs.

You can discover datasets curated by data providers. You can integrate shared data with existing data, augmenting it using AI capabilities.

As a producer, you can share data in the Marketplace, which users in the same or a different tenancy (depending on your listing configuration) can then load into their local database and consume.

With the Data Marketplace, you can:

- Publish listings as a producer to share data with the intended audience, based on your publishing configuration. You can also view a statistical profile of your dataset.
- As a consumer, load, analyze, and query the data, leveraging Data Studio AI capabilities such as language detection, sentiment analysis, and key phrase extraction. You can also browse datasets available to you based on your access profile.

Data Marketplace Terminology

- **Dataset:** A unit of data shared in the Autonomous AI Database Marketplace. Producers can create datasets that include tables and views.
- **Listings:** Published sets of tables available for consumption in the Data Marketplace. Producers can define descriptions, namespaces, tags, documentation links, and target audience. Listings appear as cards on the Data Marketplace page.

Topics:

- [Access Data Marketplace Tool](#)
To access shared datasets, navigate to Data Studio and launch the Data Marketplace tool.
- [Publish Listings to Marketplace](#)
As a producer, create and configure listings, then submit them for publishing to make your tables available in the Marketplace.
- [Consume Dataset](#)
Depending on your sharing configuration, consumers can access your dataset on the Data Marketplace page.
- [DBMS_DATAMART Package Reference](#)
This section provides information about packages used with the Data Marketplace tool in Data Studio. The `DBMS_DATAMART` topic also covers the procedures included in the `DBMS_DATAMART` package. The `DBMS_DATAMART` package provides APIs to support common operations with Data Marketplace.

Access Data Marketplace Tool

To access shared datasets, navigate to Data Studio and launch the Data Marketplace tool.

Prerequisites

You must run the following code as an ADMIN user to enable the Data Marketplace tool:

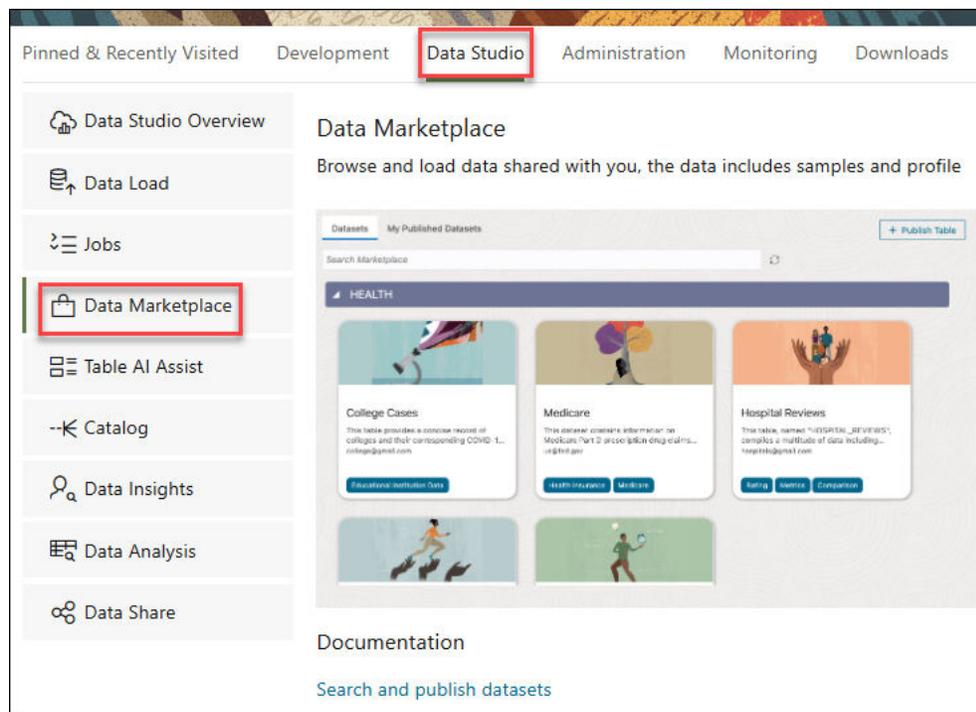
```
BEGIN
  DBMS_CLOUD_LINK_ADMIN.GRANT_READ( username => 'USER_NAME' );
END;
```

Note

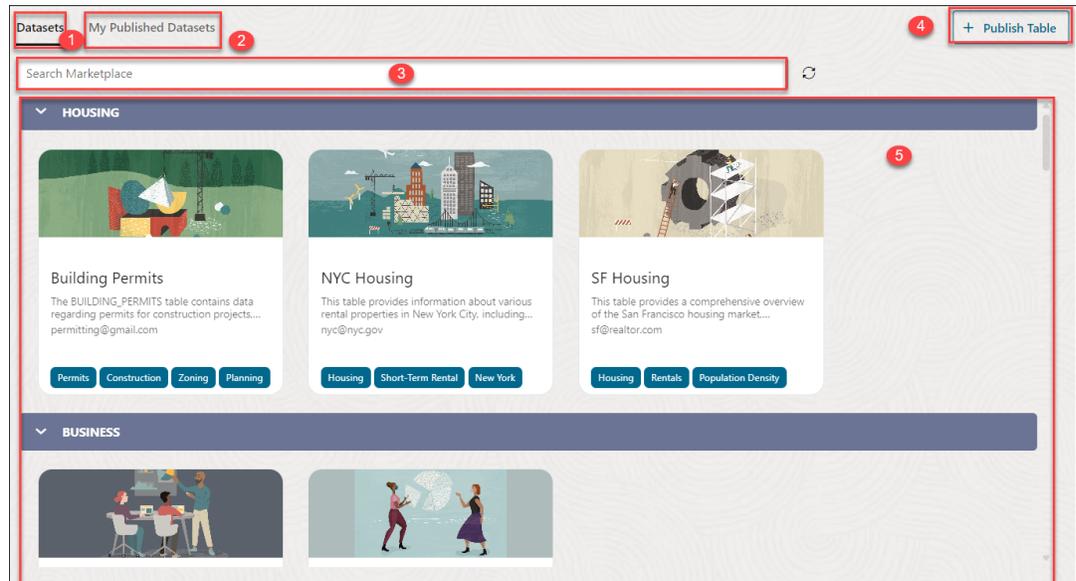
- You must have the Data Marketplace tool enabled to view the listings. Otherwise, you will view an empty Marketplace page.
- You can enable the Data Marketplace tool by running the above command or go to the Marketplace tab of the [Data Studio Settings](#).

Follow these steps:

1. Navigate to the **Launchpad** on your Database Actions instance.
2. Click the **Data Studio** tab and select the **Data Marketplace** pane.



3. Clicking the Data Marketplace tab opens the Data Marketplace page.



You can view the following components on the Data Marketplace page:

- **Datasets tab:** This is the default tab selected when you open the Data Marketplace tool. You can view all the datasets shared with you available on this page.
- **My Published Datasets:** The datasets you publish are visible on this tab under the namespace or category you specify while publishing. You can **unregister** the listing by clicking any of the published datasets and selecting **Unregister**.
- **Search Marketplace Field:** You can search for the datasets from this field.

Click the search field and enter the name of the Dataset you wish to view. You can also browse the datasets by the tags of the dataset. After you type the name and click enter, you can view the result in the display area. Click **Refresh** at the right end of the search field to refresh the page. Remove the search suggestion and click refresh to switch the view to its default view.

- **+Publish Table icon:** You can publish datasets using this icon. See [Publish Listings to Marketplace](#) for more details.
- **Display area:** The area below the search marketplace field displays the datasets returned by the search. You can expand the header which consists of the name of the category. It consists of the Dataset card with its name, description, and tags. Click the dataset card to view its details. See **Consume Dataset** for more information.

Publish Listings to Marketplace

As a producer, create and configure listings, then submit them for publishing to make your tables available in the Marketplace.

Prerequisites

You must run the following code as an ADMIN user to publish datasets for a given region:

```
BEGIN
DBMS_CLOUD_LINK_ADMIN.GRANT_REGISTER( username =>'USER_NAME', scope =>
'MY$REGION' );
END;
```

Note

If you do not run the above command, the tool will raise an exception.

This topic describes how to publish listings to the Marketplace.

1. On the Data Marketplace page, select **+Publish Table**. You can view a **Publish Table to Marketplace** wizard.
2. On the **General** tab, specify the following details:

The screenshot shows the 'Publish Table to Marketplace' wizard in the 'General' tab. At the top, there are two numbered steps: '1 General' and '2 Preview'. The 'General' tab is active. The form contains the following fields and controls:

- Schema:** A dropdown menu with 'RETAIL' selected.
- Source:** A dropdown menu with 'SALES' selected.
- Preview Data:** A button to the right of the Source dropdown.
- Namespace (Category):** A dropdown menu with 'Documentation' selected.
- Name:** A text input field containing 'Book'.
- Description:** A text area containing 'The table SHEET1 contains information pertaining to context and names of unilinks along with their corresponding URLs.'
- Share my data to:** A link icon.
- Show code:** A checkbox that is currently unchecked.
- Navigation buttons:** 'Back', 'Next', 'Publish', and 'Cancel' buttons are located at the bottom right.

- **Schema:** Select the schema from the drop-down. For example, *RETAIL*.
- **Source:** You can view the source tables you load in the Autonomous AI Database using the Data Load tool. For example, *SALES*.

Note

Note: If you have not loaded a table, the drop-down will be empty and the option will be greyed out. Make sure you have source data.

Click **Preview Data** to view the preview of the loaded table. You can also view the statistics of the table.

See [View Statistics of Data Entities](#) for more information.

- **Namespace:** Specify the category of the data set you wish to group your data sets under. For example, *Documentation*.
- **Name:** Specify the name of the dataset. This should fall under the Namespace category. For example, *Book*.
- **Description:** This field is Auto filled using the AI capabilities of the Marketplace feature. It displays the description of the dataset shared in the listing. The description helps consumers understand the content of your dataset.

Note

- You can view the auto filled description only after you have configured and set AI profile using the [Use DBMS_CLOUD_AI to Configure AI Profiles](#) procedure.
- After you have configured and set your AI profile, you must set the [Data Studio Settings](#) wizard on the Connections page.

The screenshot shows the 'Publish Table to Marketplace' wizard with the 'General' tab selected. A progress bar at the top indicates two steps: 'General' (1) and 'Preview' (2). The 'Share my data to' section has radio buttons for 'My Compartment', 'My Tenancy' (selected), and 'OCID'. There are checkboxes for 'Include Table Sample Data' and 'Include Table Statistics', both of which are checked. A 'Restricted' toggle switch is currently turned off. Below this, there is a text input field for 'Owner (email)' containing 'abc@xyz.com' and an empty text input field for 'Link (URL)'.

- **Share my data to:** Specifies who and from where you are allowed to access the registered table.

You can choose from any of the following available options:

- **MY TENANCY:** You can grant remote data access to any resource, tenancy, compartment, or database in the tenancy of the Autonomous AI Database instance that is registering the data set.
- **MY COMPARTMENT:** You can grant remote data access to any resource, compartment, or database in the compartment of the Autonomous AI Database instance that is registering the data set.
- **OCID:** Access to the data set is allowed for the specific Autonomous AI Database instances identified by OCID.

Note

Follow the steps provided in [GET SHARING ID](#) to obtain the OCID. It is alternatively known as `SHARING_ID`.

You can also include to share the Sample Data and Statistics of the table you select.

Restricted: When you publish a listing as restricted, the producer will receive an email notification with the Database ID for your listing.

Request Access

Email

Subject

Body

, a user identified by Cloud Link Sharing ID 14551723751541538224 would like to access your Data Set EMP_RESTRICTED.

To allow access, navigate to Marketplace in Database Actions for your Autonomous Database, find your data set and click "Grant Access." Enter the sharing ID above.

You can also paste the following into SQL Developer Web or SQL*Plus:

```
begin
  dbms_datamart.grant_access(namespace => 'BUSINESS',
    data_set_name => 'EMP_RESTRICTED',
    auth_info => '{ "db_id": "14551723751541538224" }');
end;
/
```

Best regards,

The producer shall grant access to the dataset for the consumer to consume the dataset by specifying the Database ID shared in the email.

- **Owner (email):** Specify the email address of the producer, for example, *abc@xyz.com*.
- **Link (URL):** Specify a URL that you can browse to discover more details on the data the provider shares. This field is optional.
- **Tags:** In the **Tags** text box, enter the tags of your choice. You can use tags to filter the Datasets on the search field.

Publish Table to Marketplace

1 General 2 Preview

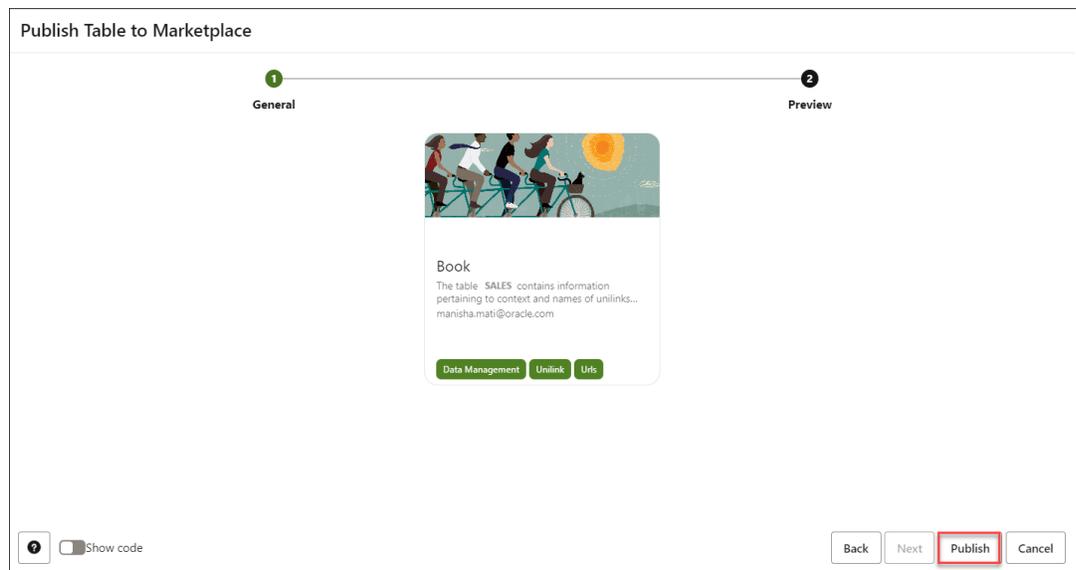
Link (URL)




Tags

Show code

3. Click **Next** to progress to the **Preview** tab. The preview tab displays the preview of the dataset card that will be displayed as a result of this publish.



4. Click **Publish** to publish the table to the Marketplace.

You can view the published dataset in the **My Published Dataset** tab of the Data Marketplace page.

Consume Dataset

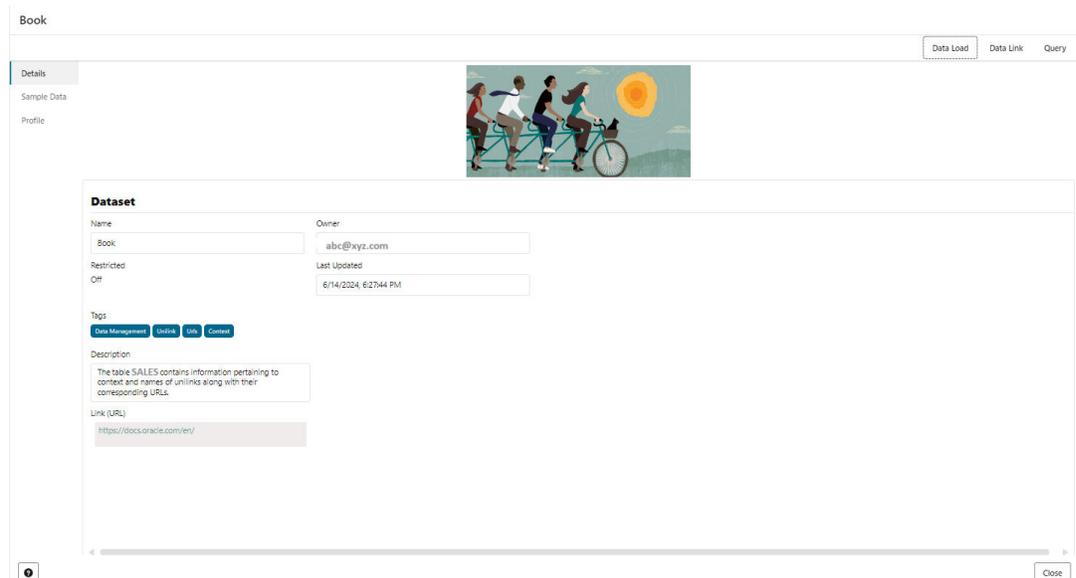
Depending on your sharing configuration, consumers can access your dataset on the Data Marketplace page.

As consumers you can view and consume the data in the dataset by following these steps:

1. On the Data Marketplace page select the listing whose data you wish to view.

This opens the listing card.

2. You can view the following details of the Dataset:



You can view the following fields on the Details tab:

- **Name:** Displays the name of the dataset.
 - **Owner:** Displays the email address of the provider.
 - **Restricted:** Specifies if the dataset shared is restricted or not.
 - **Tags:** Displays the tags you selected while configuring the listing.
 - **Description:** Displays description of the dataset shared in the listing.
 - **Link(URL):** Displays the URL the producer specifies during configuring the listing for publishing which displays more details on the data they share.
3. The Sample Data tab displays the preview of all the columns of the table selected in the dataset.

	CONTEXT	UNILINK_NAME	UNILINK_URL
1	en/database/oracle/sql-developer-web		(null)
2	en/database/oracle/sql-developer-web		(null)
3	en/database/oracle/sql-developer-web		(null)
4	en/database/oracle/sql-developer-web		(null)
5	en/database/oracle/sql-developer-web		(null)
6	en/database/oracle/sql-developer-web		(null)
7	en/database/oracle/sql-developer-web		(null)
8	en/database/oracle/sql-developer-web		(null)
9	en/database/oracle/sql-developer-web		(null)
10	en/database/oracle/sql-developer-web		(null)
11	en/database/oracle/sql-developer-web		(null)
12	en/database/oracle/sql-developer-web		(null)
13	(null)	(null)	(null)
14	(null)	(null)	(null)

4. The Profile tab displays the statistical data of the dataset. You can view:

- The total number of rows and columns in the data entity is displayed at the top.
- The statistics panel displays the thumbnail graphs for each column with information about the Min, Max, Distinct, and Null values.
- Two types of thumbnail representations are displayed based on the histogram:
 - A bar chart represents data for frequency and top-frequency histograms. The bar chart shows the first top 10 values for the number of rows in the table.
 - A table lists data for Hybrid and Height-Balanced histograms. The table displays the entire data and is scrollable. The table displays the range for the values and the percentage of rows in each range.
- You can click expand thumbnail to view the statistics of the column in a new tab.
- The detailed view of each chart also shows the type of histogram.

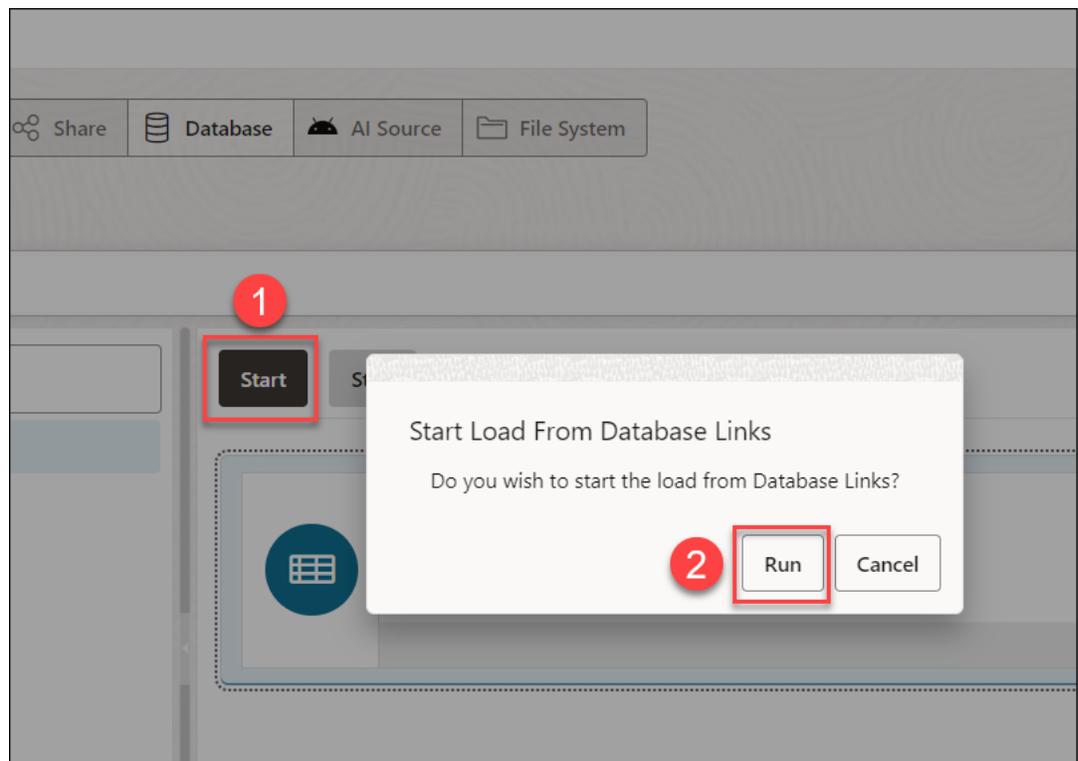
5. On the top header of the listing card details you can view the following icons:

Data Load: Selecting this option takes you to the Data Load page. You can create external tables derived from tables selected from the dataset.

Drag and drop tables from the selected dataset.

In this example, the only table selected is *BOOK*.

Create the external table by clicking **Start**, on the Load Data page, and then clicking **Run** on the Run Data Load Job dialog.



You can view the external table on the Dashboard under **Table and View Loads**.



- **Data Link:** Selecting this icon opens the Data Link page which is similar to the above operation of selecting the Data Load icon.
- **Query:** Selecting this icon opens the [The Data Analysis Tool](#) with the associated table as source input. This icon helps you analyze the data in the dataset and perform queries with the associated table. You can also create analytic views from the associated table in the dataset. See [Data Analysis tool](#) for more information.

DBMS_DATAMART Package Reference

This section provides information about packages used with the Data Marketplace tool in Data Studio. The DBMS_DATAMART topic also covers the procedures included in the DBMS_DATAMART

package. The `DBMS_DATAMART` package provides APIs to support common operations with Data Marketplace.

Topics:

- [DBMS_DATAMART Procedures](#)
- [Summary of DBMS_DATAMART Subprograms](#)
You can use Oracle Data Marketplace APIs to register and access datasets.

Summary of DBMS_DATAMART Subprograms

You can use Oracle Data Marketplace APIs to register and access datasets.

The `DBMS_DATAMART` package simplifies common operations with Oracle Data Marketplace, such as registering objects, requesting access, granting access, revoking access, unregistering and listing datasets.

The following table lists the `DBMS_DATAMART` subprograms and briefly describes them.

DBMS_DATAMART Package Subprograms

Subprogram	Description
register_object	This procedure registers datasets.
request_access	This procedure requests access to a specific dataset.
grant_access	This procedure grants access to a specific dataset.
revoke_access	This procedure revokes access to a specific dataset.
unregister	This procedure unregisters a dataset.
list_data_sets	This procedure returns the list of datasets.
get_stock_image_urls	This procedure returns the URL base from which you can select images.
get_reserved_namespaces	This procedure displays a JSON array of namespaces which are reserved by the system and cannot be used by you.
update_registration	This procedure updates an existing registration.

Register Object Procedure

This procedure registers datasets.

Syntax

```
procedure register_object(schema_name          in varchar2,
                        object_name          in varchar2,
                        namespace           in varchar2,
                        data_set_name       in varchar2,
                        scope                in clob default null,
                        thumbnail            in blob default null,
                        thumbnail_url        in varchar2 default null,
                        thumbnail_mime_type  in varchar2 default null,
                        parameters           in clob default null);
```

Register Object Parameters

If you are accessing Register Object procedures, use the following parameters.

Parameter	Description
schema_name	The name of the schema.
object_name	The name of the object.
namespace	The namespace for the object.
data_set_name	The name of the dataset.
scope	This parameter indicates who can remotely access the data. The scope can be set to various levels, including to the region where the database resides, to individual tenancies, or to compartments. MY\$REGION is not supported.
thumbnail	The image the tool displays in the cover of a dataset. You cannot use this parameter.
thumbnail_url	You cannot specify your own thumbnail_url. Oracle provides this parameter.
thumbnail_mime_type	It specifies Multipurpose Internet Mail Extensions (MIME) type of thumbnail.
Parameters	It specifies JSON parameters such as tags, description, data_set_owner, authorization_required, offload_targets, offload_targets and store_sample.

Request Access Procedure

This procedure requests access to the specified dataset.

Syntax

```
procedure request_access(namespace    in varchar2,
                        data_set_name in varchar2,
                        auth_info    out clob);
```

Request Access Parameters

If you are accessing Request Access procedure, use the following parameters.

Parameter	Description
Namespace	The namespace for the object.
Dataset name	The name of the dataset.
Auth Info	This parameter is generated populated as JSON on return. If the dataset provider has provided an email address and the requesting user has set up their Data Studio SMTP settings, the request will be submitted via email and this parameter will be generated with the delivery field set to smtp. Otherwise, the delivery field will be set to browser and you must use the rest of the metadata to submit their own request over email.

Grant Access Procedure

This procedure grants access to the specified dataset. The auth info parameter must contain the Sharing ID.

Syntax

```
procedure grant_access(namespace      in varchar2,  
                      data_set_name in varchar2,  
                      auth_info     in clob);
```

Revoke Access Procedure

This procedure revokes access to the specified dataset. The auth info parameter must contain the Sharing ID.

Syntax

```
procedure revoke_access(namespace      in varchar2,  
                        data_set_name in varchar2,  
                        auth_info     in clob);
```

Unregister Procedure

This procedure revokes access to the specified dataset. The auth info parameter must contain the Sharing ID.

Syntax

```
procedure unregister(namespace      in varchar2,  
                    data_set_name in varchar2);
```

List Dataset Procedure

This procedure displays a list of datasets.

Syntax

```
function list_data_sets(registrations in boolean default false)  
    return clob;
```

If `registrations` is `true`, it returns the list of datasets you register. If `false`, it returns the available data sets.

Get Stock Image URL Procedure

This procedure displays the URL from which you can select images.

Syntax

```
function get_stock_image_url return varchar2;
```

Get Reserved Namespaces Procedure

This procedure displays JSON array of namespaces that are reserved by the system. You cannot use these namespaces.

Syntax

```
function get_reserved_namespaces return clob;
```

Update Registration Procedure

This procedure updates an existing registration.

Syntax

```
procedure update_registration(namespace          in varchar2,  
                             data_set_name    in varchar,  
                             scope             in clob default null,  
                             thumbnail         in blob default null,  
                             thumbnail_url     in varchar2 default null,  
                             thumbnail_mime_type in varchar2 default null,  
                             parameters       in clob default null);
```

The parameters of this procedure are already described above.

The Table AI Assist Tool

The **Table AI Assist Tool** enables you to add, remove, and rename columns in a table using AI assistance. It enables you to augment or correct data found in tables in your local schema using natural language AI prompts. For example, prepare data for analytics by adding columns for days of the week or reporting quarters or deriving the distance between two sets of geographical co-ordinates, and more.

You can add columns for day, month, quarter and year using data in a date column or add a distance column from two columns with geographical co-ordinates.

The **Table AI Assist** tool enables you to prepare and enhance data for analytics, for example by deriving separate columns from a single date column for time-based reporting, or by calculating the distance between two geographical co-ordinates and adding that as a new column. The Table AI Assist tool works by using your configured AI profile to generate SQL to prepare your data. The output of the tool can then be a new view on top of your source table, or a new table.

Using the Table AI Assist tool, you can interact with the data that maintains context and have a smooth and natural conversation. See [Conversational AI for the Table AI Assist Tool](#) for more information.

Prerequisite:

See [Data Studio Settings](#) to set and configure your AI profile.

The following topics describe the **Table AI Assist** page and how to use it:

- [Prepare Data Using Table AI Assist Tool](#)
Data Augmentation helps you to add data to your source tables from your target tables. The Table AI Assist allows you to add, rename, and remove columns in a table using natural language prompts.
- [View Table Entity Details](#)
You can view the entity details of the selected entity by just clicking the **Actions** icon on the entity. This displays a list of icons that lets you view the entity's details. The icons vary from entity to entity.
- [Conversational AI for the Table AI Assist Tool](#)
The Table AI Assist tool uses conversational AI technology to make you feel close to the data.

Prepare Data Using Table AI Assist Tool

Data Augmentation helps you to add data to your source tables from your target tables. The Table AI Assist allows you to add, rename, and remove columns in a table using natural language prompts.

The **Table AI Assist** tool uses an AI service to generate SQL. You can choose to use the generated SQL to alter the table, create a view, or create a SELECT statement. It enables you to try suggested prompts generated by Artificial Intelligence (AI) and receive a query that is run where you can preview the data.

What is a Recipe?

A recipe is a set of steps. Each step lets you add, replace, remove and rename columns without making any changes to your base table. You add as many steps as you want to a recipe. You can save a recipe to your schema or export it as JSON to a file. You can return back to your saved recipe and keep working on it.

Different modes of Recipe

The following are different modes of a recipe:

- Create or Replace View
- Create a new table
- Alter table

Once you are satisfied with a recipe you can use it to Create or replace a View, Create a new Table or Alter the Table.

You can choose how to use the recipe at any time. The choices are different modes of recipe.

Let's say you have a table with four columns, and then add two new columns that are derived from the original data. The **Table AI Assist** tool can then:

- Alter the table to add the two new columns.
- Create either a new table, or a view on the table, with any of the four original columns and the two new columns.

Consider a table with *EMP* details. The table contains columns such as *HIREDATE*, *ENAME*, *JOB*, *MGR*, *SAL*, *COMM* and *DEPTNO*.

Using Table AI Assist, you can use natural prompts such as:

- add month in format year-month from *order_date*
- add year from *order_date*

The **Table AI Assist** tool generates SQL expressions for each prompt and gathers all changes into a recipe.

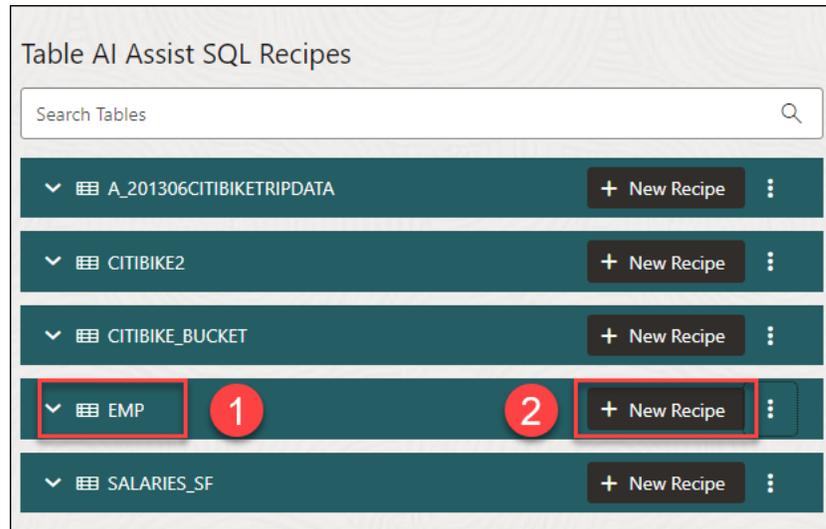
The following article demonstrates augmenting data using the **Table AI Assist** menu.

1. Select the table to be modified.

- a. From Launchpad, select Table AI Assist menu from the Data Studio suite of tools.
- b. On the **Table AI Assist** page, you can either search for the existing table from the search field at the top of the page or view it on the list of tables. In this example, let us select the **EMP** table.

2. Create a new Recipe.

- a. Click **+ New Recipe** to create a new recipe.



You can build a Recipe using a set of steps to Add, Update, Remove and Rename Columns without modifying your source Table.

b. Select the Target Type and Target Name for viewing the changes you make to the source data.

On the right panel of the Table AI Assist page, specify the following field values:

- **Source Table Name:** This is the name of the source table. In this example, it is *EMP*.
- **Target Type:** You can select one of the following options:
 - **Create View:** This option creates a view using steps in the recipe and updates the column values using column expressions in the view. Views contain only selected columns and expressions. Views do not store data.
 - **Create Table:** This option creates a new table using steps in the recipe. Data is stored in all columns, including columns added by using the recipe. It stores the updated column values.
 - **Alter Table:** This option adds, removes, and renames columns in the source table and stores the updated column values using steps in the recipe. The new columns are added as virtual columns using expressions (they do not store data).
 - **Query:** This option creates a query using steps in the recipe.

In this example, you will **Create Table**.

- **Target Name:** Enter the name of the newly created table or view. It is a table or a view depends on the selection in **Target Type** drop-down field.

In this example, you will view *EMP_1*.

Source Table Name

Target Type

Create Table ▼

Target Name

Add Step
Clear All

Build a Recipe using a set of steps to Add, Update, Remove and Rename Columns without modifying your source Table

Show code

▶ Create Table

3. Add Steps to build the recipe that is comprised of various column actions.

- a. Click **Add Step** to update the source tabular data.

Table AI Assist > Edit Recipe

EMP_RECIPE1

Preview Data

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH	CLERK	7902	12/17/1980	800	(null)	
2	7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	
3	7521	WARD	SALESMAN	7698	2/22/1981	1250	500	
4	7566	JONES	MANAGER	7839	4/2/1981	2975	(null)	
5	7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	
6	7698	BLAKE	MANAGER	7839	5/1/1981	2850	(null)	
7	7782	CLARK	MANAGER	7839	6/9/1981	2450	(null)	
8	7788	SCOTT	ANALYST	7566	4/19/1987	3000	(null)	
9	7839	KING	PRESIDENT	(null)	11/17/1981	5000	(null)	
10	7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	
11	7876	ADAMS	CLERK	7788	5/23/1987	1100	(null)	
12	7900	JAMES	CLERK	7698	12/3/1981	950	(null)	
13	7902	FORD	ANALYST	7566	12/3/1981	3000	(null)	
14	7934	MILLER	CLERK	7782	1/23/1982	1300	(null)	

Source Table Name

Target Type

Create Table ▼

Target Name

Add Step
Clear All

Build a Recipe using a set of steps to Add, Update, Remove and Rename Columns without modifying your source Table

Show code

▶ Create Table

b. Specify the following fields in the Add Step dialog box:
On the Column Action, you can select one of the following available actions:

- **Add or Replace:** You can add or replace the data in the source column based on the SQL code the tool generates in the prompt field. See [Add a Column](#) or [Replace a Column](#). For example, if you replace `SALES_DATE` with `YEAR`, (assume `SALES_DATE` as a date data type) with prompt such, you will get a `SALES_DATE` column with years.
- **Remove:** You can remove a column from the source data. See [Remove a Column](#).
- **Rename:** You can rename a source column. See [Rename a Column](#).

The fields generated when you add steps varies on the column action you select.

Using Guide for Oracle Cloud
G31944-10
Copyright © 2020, 2025, Oracle and/or its affiliates.

December 22, 2025
Page 4 of 19

Add Step

Column Action

Add Or Replace Remove Rename

Enter a prompt in natural language or specify SQL for column

Ask AI to generate SQL

1

Year from HIREDATE Month from HIREDATE

⚡ Ask AI 2

?

Add Step Clear Cancel

- Select **Year from HIREDATE**.
- Click **Ask AI**.
- Click **New Column** to create a new column and store the **Year from HIREDATE** value.

Add Step

Column Action

Add Or Replace Remove Rename

Enter a prompt in natural language or specify SQL for column

Year from HIREDATE

[Year from HIREDATE](#) [Month from HIREDATE](#)

⚡ Ask AI 1

New Column Replace Column

"YEAR FROM HIREDATE"

SQL

EXTRACT(YEAR FROM "HIREDATE")

Data Preview

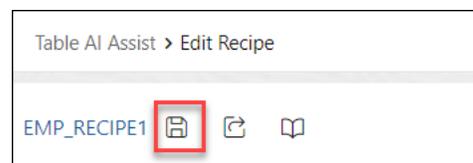
	HIREDATE	YEAR FROM HIREDATE	#
1	12/17/1980		1980
2	2/20/1981		1981
3	2/22/1981		1981
4	4/2/1981		1981
5	9/28/1981		1981
6	5/1/1981		1981
7	6/9/1981		1981
8	4/19/1987		1987

? 2 Add Step Clear Cancel

- Click **Add Step** to create a recipe.

4. Save Recipe.

- Click **Save Recipe** on the left panel of the Table AI assist page to save this recipe.



- Click the default Recipe name beside the **Save Recipe** to rename the Recipe. On clicking the Recipe name, you will view a **Rename Recipe** dialog.

Rename Recipe EMP_RECIPLE1

New recipe name *

NEW RECIPE

Description

This is new name of the Recipe.

? OK Cancel

Specify the following field values on the **Rename Recipe** dialog:

- **New recipe name:** Enter the recipe's name that you created. For example, *NEW RECIPE*.
- **Description:** Enter any description. This field is optional.

5. View the results

- You can view the newly added recipe in the main recipe panel on the Table AI Assist page.
- You can progressively keep adding steps, modifying the source data, storing it on the target table or target view and view the updated data in the **Preview Data** tab.

Table AI Assist > Edit Recipe

NEW RECIPE

Preview Data

	#	HIREDATE	NO	#	<i>HIRED_YEAR</i>	#
1	7902	1980-12-17T00:00:00Z	20		1980	
2	7698	1981-02-20T00:00:00Z	30		1981	
3	7698	1981-02-22T00:00:00Z	30		1981	
4	7839	1981-04-02T00:00:00Z	20		1981	
5	7698	1981-09-28T00:00:00Z	30		1981	
6	7839	1981-05-01T00:00:00Z	30		1981	
7	7839	1981-06-09T00:00:00Z	10		1981	
8	7566	1987-04-19T00:00:00Z	20		1987	
9	(null)	1981-11-17T00:00:00Z	10		1981	
10	7698	1981-09-08T00:00:00Z	30		1981	
11	7788	1987-05-23T00:00:00Z	20		1987	
12	7698	1981-12-03T00:00:00Z	30		1981	

Source Table Name: EMP

Target Type: Create Table

Target Name: EMP_1

Add Step Clear All

Extract YEAR from HIREDATE
Added HIRED_YEAR

Show code Create Table

You can view the newly added column header in *italics*.

6. Create a new target table, a new target view or Alter the target table with modified column source data.

- After you have modified the source data, you can create a new table, create a new view or Alter table based on the option you select in the **Target Type** drop-down.

- b. Select **Create Table**, **Create View** or **Alter Table** on the side panel of the Table AI Assist page based on the **Target Type** you select. The tool creates the target data upon confirmation.
- c. You will receive a successful notification message that says the creation of the target type is successful.

After you have created the recipe, you can view it under the **Table AI Assist SQL Recipes** on the Table AI Assist page.

Edit Recipe

You can edit an existing recipe from the Table AI Assist tool.

1. On the Table AI Assist page, click on the **Edit Recipe** (pencil) icon besides the recipe name. This opens the Edit Recipe page.
2. You can add steps or delete the existing steps to edit the existing recipe. Let's perform another operation of adding a new column that consists of "month" extracted from the hire date.

- Click **Add Step**. In the **Enter a prompt in natural language or specify SQL for column** text field, provide some natural language prompt of what you want to do such as

month from hiredate

- Click **Ask AI**.

Add Step

Column Action

Add Or Replace Remove Rename

Enter a prompt in natural language or specify SQL for column

month from hiredate

⚡ Ask AI

- The SQL expression that was used to create this column is displayed in the **SQL** field.

Add Step

⚡ Ask AI

New Column Replace Column

MONTH_FROM_HIREDATE

SQL

```
EXTRACT(MONTH FROM "HIREDATE")
```

Data Preview

	HIREDATE ⓘ	MONTH_FROM_HIREDATE #
1	1980-12-17T00:00:00 ⓘ	12
2	1981-02-20T00:00:00Z	2
3	1981-02-22T00:00:00Z	2
4	1981-04-02T00:00:00Z	4
5	1981-09-28T00:00:00Z	9

ⓘ Add Step Clear Cancel

Scroll to the right of the table to view the newly added **MONTH_FROM_HIREDATE** column is displayed in the **Preview Data** section.

Table AI Assist > Edit Recipe

NEW RECIPE 📄 ✎ 🗑️

Preview Data Columns

	#	HIREDATE ⓘ	HIRED_YEAR #	MONTH_FROM_HIREDATE #
1	7902	1980-12-17T00:00:00Z	1980	12
2	7698	1981-02-20T00:00:00Z	1981	2
3	7698	1981-02-22T00:00:00Z	1981	2
4	7839	1981-04-02T00:00:00Z	1981	4
5	7698	1981-09-28T00:00:00Z	1981	9
6	7839	1981-05-01T00:00:00Z	1981	5
7	7839	1981-06-09T00:00:00Z	1981	6
8	7566	1987-04-19T00:00:00Z	1987	4
9	(null)	1981-11-17T00:00:00Z	1981	11
10	7698	1981-09-08T00:00:00Z	1981	9
11	7788	1987-05-23T00:00:00Z	1987	5
12	7698	1981-12-03T00:00:00Z	1981	12
13	7566	1981-12-03T00:00:00Z	1981	12
14	7782	1982-01-23T00:00:00Z	1982	1

Source Table Name: EMP

Target Type: Create Table

Target Name: EMP_1

Add Step Clear All

Extract YEAR from HIREDATE
Added HIRED_YEAR ⓘ 🗑️

month from hiredate
Added MONTH_FROM_HIREDATE ⓘ 🗑️

Show code Create Table

3. Click **Save Recipe** to save the updated recipe.

Export Recipe

You can export the specific recipe.

1. On the **Edit Recipe** page.
2. Click the Export Recipe icon besides the Recipe name.
3. Select the **Export** icon on the to generate the JSON file for download.

Load or Import Recipe

You can load or import recipe.

1. Click **Load Recipe** on the Edit Recipe page. Selecting Load Recipe opens a **Load Recipe** Dialog which displays the following options:
 - Load
 - Select **Load** to load an existing recipe.

Load Recipe

Load Import

Table

CALIFORNIA_HOUSING

Recipe

CALIFORNIA_HOUSING_RECIP1

Steps

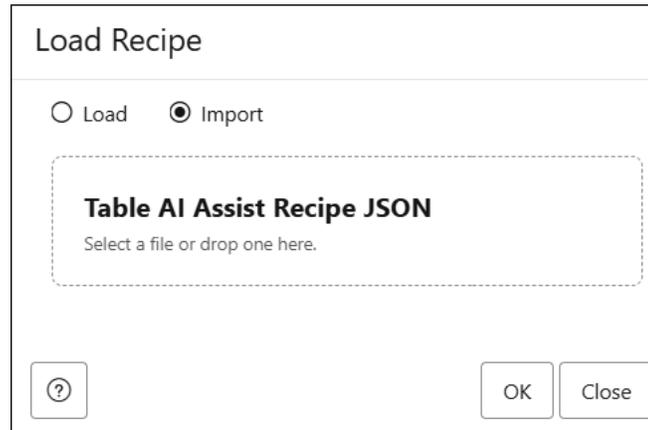
Calculate POPULATION_DENSITY
Added POPULATION_DENSITY

% of TOTAL_ROOMS are TOTAL_BEDROOMS
Added PERCENT_OF_TOTAL_ROOMS_ARE_TOTAL_BEDROOMS

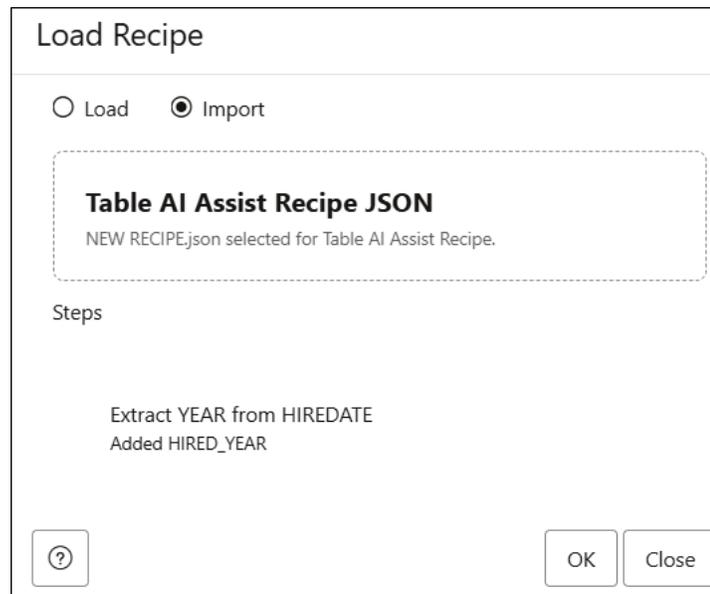
Remove column
COLNAME_COL11_MISSING

- Specify the following field values:
 - * **Table:** Select the table from the drop-down field you want to load the recipe from.
 - * **Recipe:** Select the recipe you want to load from the drop-down.
 - * Under Steps, you can view the Steps involved in creating the recipe.
 - * Click **OK** to load the recipe.

- Import
 - Select **Import** to import a recipe file you download. This file is in JSON format.



- Drop an existing Table AI Assist Recipe file (in JSON format) you have exported.



- Click **OK** to import the recipe. Click **Cancel** to cancel the ongoing process of loading or importing.

You can share a recipe with another user by importing it. For example, if you and another user both load the same table, and you use the Table AI Assist tool to set up a bunch of new and interesting columns, you can import and share the recipe with the other user so they can view the modified data.

- [Add a Column](#)
You can view the following fields when you select **Add or Replace** from Column Action.
- [Replace a Column](#)
You can view the following fields when you select **Add or Replace** from Column Action.
- [Remove a Column](#)
You can view the following fields when you select **Remove** from the Column Action field.

- [Rename a Column](#)
You can view the following fields when you select **Rename** from the Column Action field.
- [View Statistical Details about the Table you Load after you Create a Recipe](#)
After you have created a recipe using the Table AI Assist tool, you can view additional information such as statistics about the table you create. You can generate statistics that measure the data distribution and storage characteristics of the table.

Add a Column

You can view the following fields when you select **Add or Replace** from Column Action.

1. **Enter a prompt in natural language or specify SQL for column:** Select any one of the prompts suggested by the tool.

Note

The Data Studio suite of tools uses a Large Language Model (LLM) when using prompts to generate SQL.

2. Click **Ask AI**.
The **Ask AI** button sends the prompt, along with the table metadata (not data) to the Large Language Model (LLM) that is selected in your Data Studio settings. The LLM then attempts to generate SQL that will perform the request of the prompt.
3. Select **New Column** to create a new column that consists of the updated new column.
After you select **New Column**, you can view a text box where you specify the name of the new column.

The SQL textbox displays the SQL code the tool generates to display the action performed by the column action you selected in the previous step.
4. Under **Data Preview**, you can view the existing old column and the generated new column.

After you have decided the column action, click **Add Step** to add this step in the main recipe panel.

Clicking **Add Step** closes the Add Step dialog and opens the Table AI assist page.

Replace a Column

You can view the following fields when you select **Add or Replace** from Column Action.

1. **Enter a prompt in natural language or specify SQL for column:** Select any one of the prompts suggested by the tool.
2. Click **Ask AI**.
The **Ask AI** button to send the prompt with table metadata and asks the Large Language Model (LLM) to generate a SQL.
3. Click **Replace Column**.
Select a column you want to replace from the **Column Name to Replace** drop-down field.

The SQL textbox displays the equivalent SQL code the tool generates to display the action performed by the column action you selected in the previous step.
4. Under **Data Preview**, you can view the existing old column and the updated new column.
5. Save the newly created recipe.

Note

By default, the **Table AI Assist** tool does not replace the original source table. It only affects the new view or table that is generated by the **Table AI Assist** tool.

After you have decided the column action, click **Add Step** to add this step in the main recipe panel.

Clicking **Add Step** closes the Add Step dialog and opens the Table AI assist page.

Remove a Column

You can view the following fields when you select **Remove** from the Column Action field.

Select the column which you want to remove from the **Column Name** to remove drop-down.

After you have decided the column action, click **Add Step** to add this step in the main recipe panel.

Clicking **Add Step** closes the Add Step dialog and opens the Table AI assist page.

Note

By default, the **Table AI Assist** tool does not replace the original source table. It only affects the new view or table that is generated by the **Table AI Assist** tool.

Rename a Column

You can view the following fields when you select **Rename** from the Column Action field.

1. Select the column name you want to change the name of from the **Source Column Name** drop-down.
2. Enter the new name for the column you want to use.
3. After you have decided the column action, click **Add Step** to add this step in the main recipe panel.
4. Clicking **Add Step** closes the Add Step dialog and opens the Table AI assist page.

View Statistical Details about the Table you Load after you Create a Recipe

After you have created a recipe using the Table AI Assist tool, you can view additional information such as statistics about the table you create. You can generate statistics that measure the data distribution and storage characteristics of the table.

1. Select the table whose details you want to view.
 - a. On the Table AI Assist page, you can either search for the existing table from the search field at the top of the page or view it on the list of tables.
 - b. Click **Edit Recipe** on any of the recipes that were generated from the table.
2. View information about the tabular data
 - a. You can preview the tabular data.
 - b. Click **Columns** tab to view the following information about the table:

- **Table Size (Bytes):** Specifies the size of the table in bytes.
 - **Number of Rows:** Displays the number of rows in the table.
 - **Number of Columns:** Displays the number of columns in the table.
 - **Compressed Size:** Displays the size of compressed table in bytes.
 - **Statistics Gathered on:** Displays date and time of last statistics gathered.
- c. Click **Gather Statistics** to collect statistics periodically for tables where the statistics become stale over time because of changing data volumes or changes in column values.
3. **Column Statistics:**
- Number of distinct values (NDV) in column
 - Number of nulls in column
 - Data distribution (histogram)

The above information is displayed in the statistics details of the table.

View Table Entity Details

You can view the entity details of the selected entity by just clicking the **Actions** icon on the entity. This displays a list of icons that lets you view the entity's details. The icons vary from entity to entity.

On the **Table AI Assist** page, click the Action icon on the table header to view the details of a table. You can view the following options:

- **View Details:** Select **View Details** to view details about the table. Refer to [Viewing Entity Details](#) for more detailed information.
- **Gather Statistics:** Select this option to display new statistics after you modify the table's structure. Refer to [Gathering Statistics for Tables](#) for more detailed information.
- **Add to Share:** Select this option to share the table. Refer to [Provide Share](#) for more detailed information.
- **Register to Cloud Link:** Select this option to register the table for remote access for a selected audience you define. See [Registering Cloud Links to Access Data](#) section for more information.
- **Publish to Marketplace:** Select this option to publish the selected table to Data Marketplace. Refer to [Publish Listings to Marketplace](#) for more detailed information.
- **Create Analytic View:** Select this option to create an Analytic View from the selected associated table. See [Creating Analytic Views](#).
- **Query:** Select this option to analyze the selected table. See [The Data Analysis Tool](#) for more detailed information.
- **Export Data to Cloud:** Select [Export Data to Cloud](#) to export data to a cloud object store.
- **Edit:** Select **Edit** to edit the properties of the table. Refer to [Editing Tables](#) for more information.
- **Drop:** Select **Drop** to delete the table.

Conversational AI for the Table AI Assist Tool

The Table AI Assist tool uses conversational AI technology to make you feel close to the data.

You can have an interactive dialogue with the tool, where a sequence of natural language prompts is used to query or interact with the database. The tool adapts to what you are asking, picks up on context, and provides relevant responses. For example, if you are a risk analyst working in a banking firm, you can ask, “*Is this transaction authorized by you?*”, and follow up with context aware security question. The tool can verify suspicious transaction and flag it. With this feature, the tool continuously learns from interactions and improves the detection accuracy.

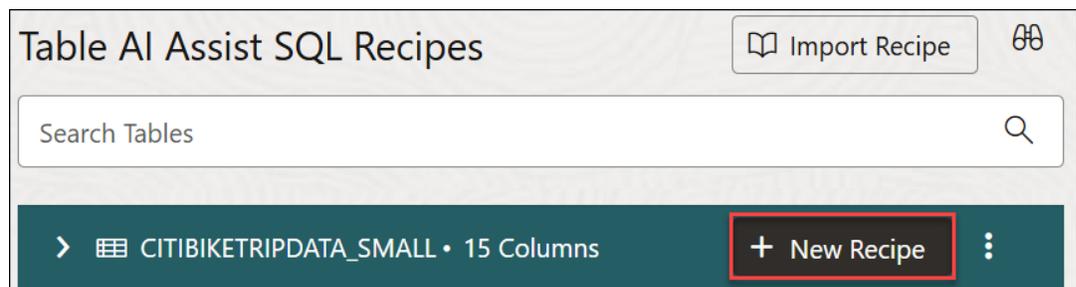
Working of the Conversational AI in the Table AI Assist Tool

When using Table AI Assist, you're essentially creating a natural language prompt that translates into a new column in your table. This new column contains the SQL expression that corresponds to your prompt.

You ask a question, prompting data retrieval model to access the table. The retrieval model queries your autonomous database and responds with a refined prompt that augments your original query with supplementary contextual details.

In the following example, we shall check if the bike trip started from Manhattan:

1. On the Table AI Assist page, the tables that you have created so far are displayed. Click + **New Recipe** beside the *CITIBIKETRIPDATA_SMALL* Table.



You can view the Edit Recipe page.

2. Select **Add or Replace Column** from the + **Add Step** drop-down field.

- On the Add Step dialog, enter the following prompt in the **Ask AI to generate SQL** text box area:

Does the trip start in Manhattan?

You can also select **+ Include Column Details** to perform any of the following actions:

- View All Columns:** If you are working on adding a step and forget the names of your columns, you can click this button to see a list of all your column names.
- Basic Prompt Building:** You can click on column names to automatically add them to your prompt, so you do not have to type them out. However, you still need to complete the prompt by explaining what you want to do with those columns.

Add Sample Data to Your Prompt: For example, if you have a `VARCHAR` column that contains dates and you want to add one year to those dates, simply asking Add one year to `my_varchar_date` column might not work because the AI does not know the date format (like `YYYY/MM/DD` or `DD/MM/YYYY`). By default, we only send column names and data types to the model, not the actual data.

If you use the **+ Include Column Details** option, you can enter the column name and check **Include Sample Values**. This adds example data to your prompt, like:

```
Add one year to my_varchar_date column. my_varchar_date sample values:
1995/4/5, 2014/12/31
```

With this added context, the tool understands how to handle the date strings properly. Sending sample data can also help in other cases, such as redacting sensitive information like phone numbers, credit cards, or emails.

4. Click **Ask AI**.

Selecting **Ask AI** opens a right pane in the Add Step dialog that displays the newly generated column name, the equivalent SQL code and the new generated `TRIP_STARTS_IN_MANHATTAN` column is displayed in the Data Preview section. You can also rename this column if you want.

The screenshot shows the 'Add Step' dialog with the following content:

Chat Prompt: Does the trip start in Manhattan?

Generated SQL:

```
SELECT
CASE
WHEN "START_STATION_LATITUDE" BETWEEN 40.7856 AND 40.9125
AND "START_STATION_LONGITUDE" BETWEEN -74.8467 AND -73.99074142
THEN "YES"
ELSE "NO"
END
```

Data Preview:

	START_STATION_LONGITUDE #	TRIP_STARTS_IN_MANHATTAN
1	-73.98915076	YES
2	-73.98915076	YES
3	-73.99595065	NO
4	-73.98758561	YES
5	-74.01677685	YES
6	-73.98338988	YES
7	-73.99074142	YES
8	-74.00197139	VFS

5. If you want to learn more about the trip, for example, whether it began in `BROOKLYN`, you can continue the conversation in the **Refine the returned response or save the step** text field.

Enter the following prompt in the chat area:

Brooklyn?

Add Step

Does the trip start in Manhattan?

```
SELECT
CASE
WHEN "START_STATION_LATITUDE" BETWEEN 40.7056 AND 40.9125
AND "START_STATION_LONGITUDE" BETWEEN -74.0467 AND -73.7562
THEN 'Yes'
ELSE 'No'
END
```

This query checks if the trip starts in Manhattan by verifying if the "START_STATION_LATITUDE" and "START_STATION_LONGITUDE" fall within the geographical boundaries of Manhattan. The latitude range for Manhattan is approximately between 40.7056 and 40.9125, and the longitude range is between -74.0467 and -73.7562. If the coordinates are within these ranges, the query returns 'Yes' for the "TRIP_STARTS_IN_MANHATTAN" column; otherwise, it returns 'No'.

Extract YEAR from STARTTIME Extract MONTH from STOPTIME Calculate difference between STARTTIME and STOPTIME

Brooklyn

Clear + Include Column Details Ask AI

New Column Replace Column

TRIP_STARTS_IN_MANHATTAN

```
SQL
CASE WHEN "START_STATION_LATITUDE" BETWEEN 40.7056 AND 40.9125
AND "START_STATION_LONGITUDE" BETWEEN -74.0467 AND -73.7562
THEN 'Yes'
ELSE 'No'
END
```

Data Preview

	START_STATION_LATITUDE #	START_STATION_LONGITUDE #
1	40.7423543	-73.98915076
2	40.7423543	-73.98915076
3	40.69512845	-73.99595065
4	40.73524276	-73.98758561
5	40.70569254	-74.01677685
6	40.75038009	-73.98338988
7	40.73454567	-73.99074142
8	40.7454973	-74.00197139

Save Cancel

Click **Ask AI**.

Add Step

Does the trip start in Manhattan?

```
SELECT
CASE
WHEN "START_STATION_LATITUDE" BETWEEN 40.7056 AND 40.9125
AND "START_STATION_LONGITUDE" BETWEEN -74.0467 AND -73.7562
THEN 'Yes'
ELSE 'No'
END
```

This query checks if the trip starts in Manhattan by verifying if the "START_STATION_LATITUDE" and "START_STATION_LONGITUDE" fall within the geographical boundaries of Manhattan. The latitude range for Manhattan is approximately between 40.7056 and 40.9125, and the longitude range is between -74.0467 and -73.7562. If the coordinates are within these ranges, the query returns 'Yes' for the "TRIP_STARTS_IN_MANHATTAN" column; otherwise, it returns 'No'.

Brooklyn?

Extract YEAR from STARTTIME Extract MONTH from STOPTIME Calculate difference between STARTTIME and STOPTIME

Refine the returned response or save the step

Clear + Include Column Details Ask AI

New Column Replace Column

TRIP_STARTS_IN_BROOKLYN

```
SQL
CASE WHEN "START_STATION_LATITUDE" BETWEEN 40.5513 AND 40.7377
AND "START_STATION_LONGITUDE" BETWEEN -74.8415 AND -73.7447
THEN 'Yes'
ELSE 'No'
END
```

Data Preview

	START_STATION_LONGITUDE #	TRIP_STARTS_IN_BROOKLYN A
1	-73.98915076	No
2	-73.98915076	No
3	-73.99595065	Yes
4	-73.98758561	Yes
5	-74.01677685	Yes
6	-73.98338988	No
7	-73.99074142	Yes
8	-74.00197139	No

Save Cancel

Note

Observe that the tool understands the background of the chat from earlier parts of the conversation and responds accordingly, rather than treating each prompt as isolated and displays the modified CITIBIKETRIPDATA_SMALL table with newly generated TRIP_STARTS_IN_BROOKLYN column

- Click **Save** to save the modified CITIBIKETRIPDATA_SMALL with newly added TRIP_STARTS_IN_BROOKLYN column.

Note

To save the previously modified tables, be sure to click **Save**; otherwise, your updates will be lost.

You can then proceed further by creating a recipe from the above steps. See [Prepare Data Using Table AI Assist Tool](#) for more information on creating a recipe.

19

The Jobs Feature

The Jobs feature automates the tasks run by the Data Studio suite of tools. It can be scheduled to run at specific intervals or triggered by certain events.

The different types of tasks the Jobs feature of the Data Studio tool run are:

- Data Load using the Data Load Tool
- Augment data with Table AI Assist Tool
- Run PL/SQL procedures
- Run REST Calls

Note

You need the `CREATE JOB` privilege to create jobs in your own schema and `CREATE ANY JOB` to create jobs in other schemas.

The following pages describes the Jobs feature:

Topics:

- [How to Access the Jobs Feature?](#)
Navigate to the **Oracle Autonomous Database Data Studio**, launch the **Jobs** feature, and start running it.
- [Conceptual Diagram of the Jobs Feature](#)
The following diagram displays the conceptual workflow diagram of the Jobs feature:
- [Create a Job](#)
The Jobs feature automates the steps that were performed in the prerequisites section.
- [Manage and Monitor the Job](#)
Once the job is scheduled and created, you can manage the job and monitor each step you run for the job via the **Report** feature.
- [DBMS_DATA_TOOLS_JOBS Package Reference](#)
This chapter provides information about the packages you use with the Jobs feature in Data Studio. The Summary of `DBMS_DATA_TOOLS_JOBS` Subprograms topic also covers the procedures included in the `DBMS_DATA_TOOLS_JOBS` package.

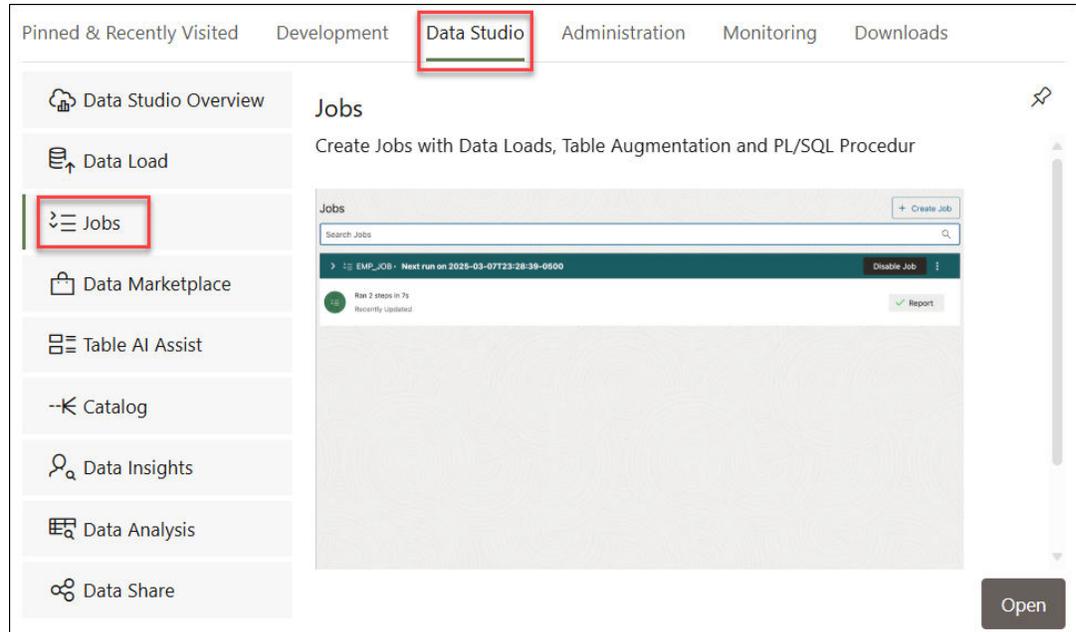
How to Access the Jobs Feature?

Navigate to the **Oracle Autonomous Database Data Studio**, launch the **Jobs** feature, and start running it.

Follow these steps:

1. Define the tasks you want to run:
 - Create a Data Load Job

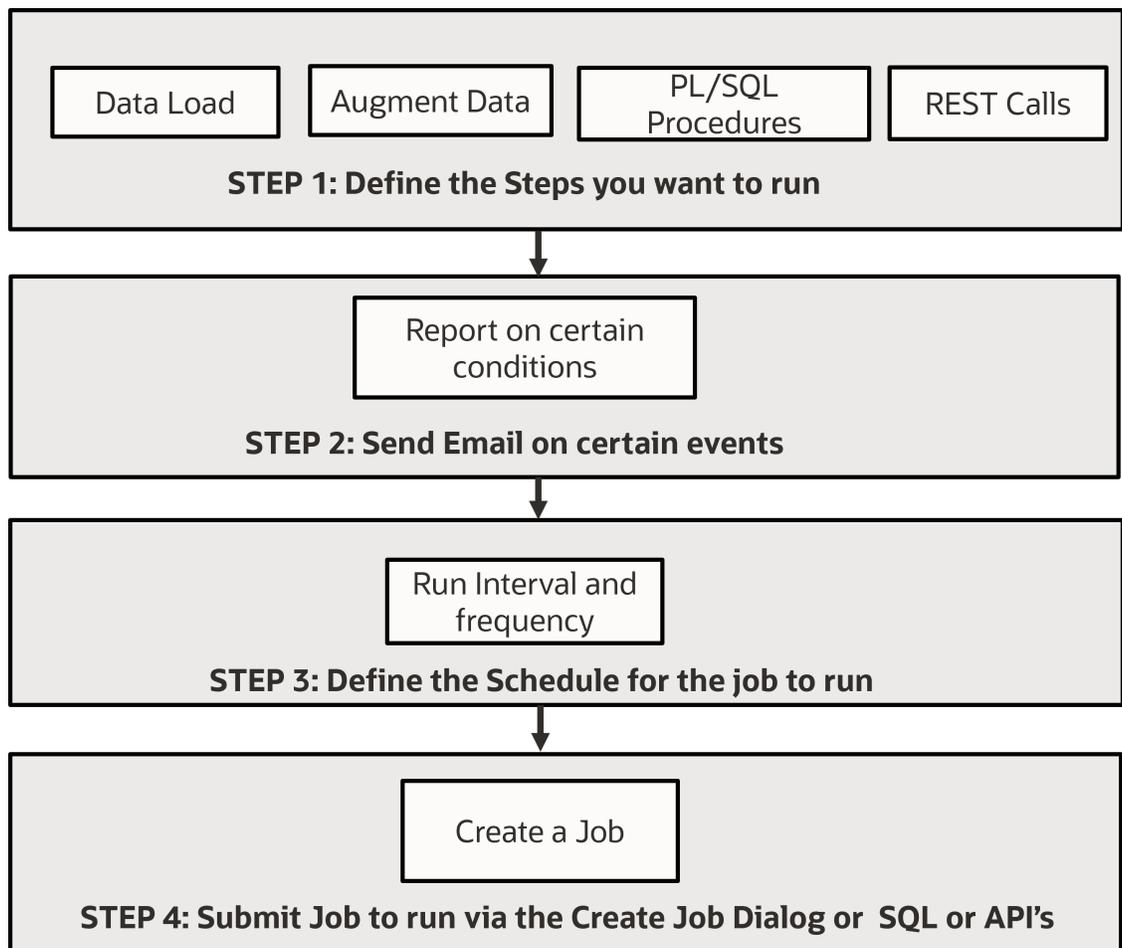
- Create a Table AI Assist Recipe
 - Create the PL/SQL Procedure you want to run
 - Obtain the parameters to run a REST call
2. On the Launchpad, click **Data Studio** and select the **Jobs** menu.



You can automate the tasks you define in Step 1 with Jobs feature.

Conceptual Diagram of the Jobs Feature

The following diagram displays the conceptual workflow diagram of the Jobs feature:



Prerequisites

Before using the Jobs feature, you must decide the tasks you want the Jobs feature to automate. For each kind of job step that you want you must:

- Create a Data Load Job
- Create a Table AI Assist Recipe
- Create a PL/SQL Procedure

Note

You can choose any of these prerequisites based on the task you want to perform. You can either create a Data Load Job, a Table AI Assist recipe or create a PL/SQL procedure.

Create a Data Load Job

1. On the Data Load menu of Data Studio navigation, select **Load Data**.
2. Click **Cloud Store** and select the **Cloud Store Location** from the drop-down.

3. Select the format of the file you want to load from the drop-down. For example, to load a `csv` file, select `*.csv` from the drop-down. Drag and drop the folder with CSV files to the cart area.
4. Click **Load Name** to tag your load to assist in identifying your job during monitoring or Data Transforms. Enter the name of the load on the **Name** field. In this example, you specify **DATA_LOAD**. Click **OK**.
5. Select **Start** in the data load cart menu bar to create a table you define and load it with data from the `csv` file. A Start Load from Cloud Store message is displayed. Click **Run**. The target table is created.
6. You can view the status of the newly loaded table on the Data Load Dashboard. For more information on Loading Data from a Cloud Store, see [Loading Data from Cloud Storage](#).

Create a Table AI Assist Recipe

1. In the Navigation menu on the left, click **Table AI Assist**.
2. On the **Table AI Assist Recipes** page, select **+ New Recipe** besides the table you have created on the previous step. The Edit Recipe page is displayed.
3. Click **Add Step**. The Add Step panel is displayed.

Add Step

Column Action

Add Or Replace
Remove
Rename

Enter a prompt in natural language or specify SQL for column

extract month from UPDATED_TIMESTAMP

💡
Extract YEAR from
UPDATED_TIMESTAMP

Calculate OVERTIME_RATE as
OVERTIMEPAY / BASEPAY

Determine if EMPLOY
based on JOBTITLE
>

⚡
Ask AI

?

Save

Clear

Cancel

4. In the **Enter a prompt in natural language or specify SQL for column** text field, provide some natural language prompt of what you want to do such as

Extract Month from UPDATED_TIMESTAMP

Next, click **Ask AI**.

5. A new generated `MONTH` column is displayed in the Data Preview section. The new column shows the month extracted from the `UPDATED_TIMESTAMP` column. Note the generated

SQL that performs this operation which you can also edit and run if you are familiar with SQL.

Add Step

⚡ Ask AI

New Column Replace Column

MONTH

SQL

```
EXTRACT(MONTH FROM "UPDATED_TIMESTAMP" )
```

Data Preview ↻

	UPDATED_TIMESTAMP	MONTH #
1	2025-03-17T09:25:52.996128Z	3
2	2025-03-17T09:25:52.996128Z	3
3	2025-03-17T09:25:52.996128Z	3
4	2025-03-17T09:25:52.996128Z	3
5	2025-03-17T09:25:52.996128Z	3

? Save Clear Cancel

Click **Save** to add the step.

- The **Edit Recipe** page is re-displayed. Scroll to the right of the table to display the newly added month column.

Table AI Assist > Edit Recipe

ADP_LIVE_FEED_RECIP1

Preview Data Columns

	YEAR	UPDATED_TIMESTAMP	MONTH	#
1	2013	2025-03-17T09:25:52.996128Z		3
2	2013	2025-03-17T09:25:52.996128Z		3
3	2013	2025-03-17T09:25:52.996128Z		3
4	2013	2025-03-17T09:25:52.996128Z		3
5	2013	2025-03-17T09:25:52.996128Z		3
6	2013	2025-03-17T09:25:52.996128Z		3
7	2013	2025-03-17T09:25:52.996128Z		3
8	2013	2025-03-17T09:25:52.996128Z		3
13	2013	2025-03-17T09:25:52.996128Z		3
14	2013	2025-03-17T09:25:52.996128Z		3

Source Table Name: ADP_LIVE_FEED

Target Type: Create Table

Target Name: ADP_LIVE_FEED_1

extract month from UPDATED_TIMESTAMP
Added MONTH

Show code

Create Table

- Let us save the modified **customer** table as a new table named **LIVE_FEED**.
- Click the **Target Type** drop-down list and select **Create Table**. Change the table name in the **Target Name** field to **LIVE_FEED** and then click **Create Table**. A **Create Table** dialog box is displayed. Click **Yes**.

A **Confirmation** message box is displayed. The table is created.

For more information, see [The Table AI Assist Tool](#)

Create a PL/SQL Procedure

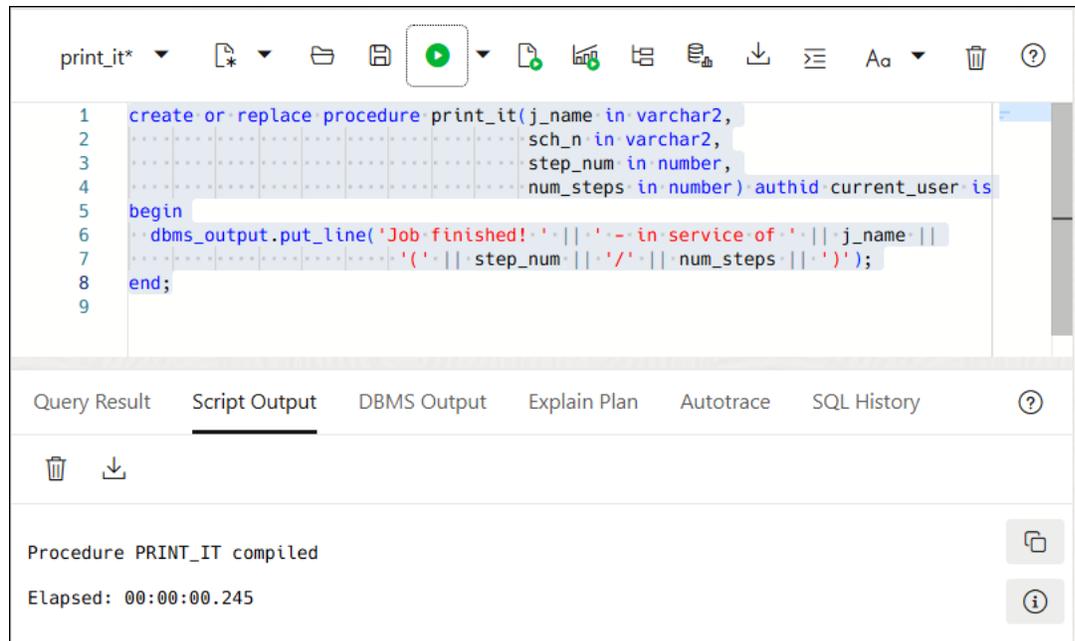
You can create a PL/SQL Procedure with SQL worksheet editor.

- Click the **Selector**  icon to display the navigation menu. Under **Development**, select **SQL**. You can use SQL and PL/SQL statements in the worksheet to create a procedure.
- Copy and paste the following code snippet on the editor area to print few lines of text:

```
create or replace procedure print_it(j_name in varchar2,
                                   sch_n in varchar2,
                                   step_num in number,
                                   num_steps in number) authid
current_user is
begin
  dbms_output.put_line('Job finished! ' || ' - in service of ' || j_name ||
                      '(' || step_num || '/' || num_steps || ')');
end;
```

Note

This is just a sample code. You can use `DBMS_CLOUD` procedures to load data into existing Autonomous Database table and perform other operations. Most of the operations you perform in Data Studio have a **Show code**. You can copy and run this PL/SQL code in the worksheet to perform the same action.



3. Run this code in the worksheet using the **Run** icon. The Query Result tab displays the statements that you print in the output.

You have successfully created the procedure.

Create a Job

The Jobs feature automates the steps that were performed in the prerequisites section.

Here is step-by-step example of creating a job using the **Jobs** feature.

1. On the Jobs feature of the Data Studio tool, select **Create Job**.

On the Create Job wizard, in the general settings specify the following field values:

- **Job Name:** Enter the name of the Job. For example, *MY NEW JOB*.
- **Job Description:** Enter the description. This field is optional. Click **Next** to proceed to the **Steps** tab.

2. **Add Steps**

On the Steps tab, you will add the steps to automate your process.

You will add the data load step, add the Table AI Assist Recipe you created, run a PL/SQL procedure and run a REST call as steps to your job.

Note

You have already defined the tasks in the [Prerequisites section](#).

The screenshot shows the 'Create Job' interface with five tabs: General, Steps, Reporting, Schedule, and API Details. The 'Job Steps' section is active, displaying 'Add steps to the Job.' A '+ Add Step' dropdown menu is open, showing options: 'Add Data Load', 'Add Table AI Assist Recipe', 'Add Procedure', and 'Add REST Call'. The 'Add Data Load' option is highlighted with a red box. At the bottom, there are buttons for 'Back', 'Next', 'Create', and 'Cancel', along with a 'Show code' toggle.

Add Data Load Step:

- On the **+Add Step** drop-down, select **Add Data Load**.
- Specify the following field values on the Add Data Load dialog:
Select Data Load: Select the Data Load from the drop-down. In this example, it is *DATA_LOAD*.

Load Name: Enter the Load name. This field will be automatically populated. This is the name by which you tag your load.

Select **Stop Job On Errors**.

Click **Add Data Load**.

The 'Add Data Load' dialog box contains the following fields and controls:
 - 'Select Data Load' dropdown menu with 'DATA_LOAD (CSV/S)' selected.
 - 'Load Name' text input field containing 'DATA_LOAD'.
 - A checked checkbox for 'Stop Job On Errors'.
 - A help icon (question mark) on the bottom left.
 - 'Add Data Load' and 'Cancel' buttons on the bottom right.

You will see your new step appear in the list of steps for your Job. Click the **pencil** icon besides the step to edit the step. Click the **remove** icon besides the step to remove the selected step.

Add Table AI Assist Recipe:

- On the **+Add Step** drop-down, select **Add Table AI Assist Recipe**.
- Specify the following field values on the **Table AI Assist Recipe** dialog:

Select Table AI Assist Recipe: Select the recipe you create from the drop-down.

Select **Stop Job On Errors**.

Click **Add Table AI Assist Recipe**.

You will see this new step get added to your list of Steps, below your existing Data Load step. Click the pencil icon besides the step to edit the step. Click the remove icon besides the step to remove the selected step.

Add Procedure

- On the **+ Add Step** drop-down of the Steps tab, select **Add Procedure**.
- Specify the following field values on the **Add Procedure** dialog:

PL/SQL Procedure: Select any one of the PL/SQL Procedures you create from the drop-down. In this example, you select *PRINT IT*.

PL/SQL Procedure Arguments: Specify a comma delimited list of arguments in this field. The values you specify in this field will be passed to the actual parameters to the procedure accepting the default values for any parameters you omit. These arguments will be passed to your PL/SQL procedure. Your arguments can be numbers, string, booleans or one of the special values starting with \$. In this case, \$job_name will be changed to My New Job when calling the procedure.

For example, to replace \$job_name with PLSQL_JOB in the output, you must pass PLSQL_JOB parameter in this field.

Select **Stop Job On Errors**.

Click **Add Procedure**.

Add Procedure

PL/SQL Procedure *

PRINT_IT

PL/SQL Procedure Arguments

\$job_name

Specify comma delimited list of arguments that can be a value or one of \$job_name, \$scheduler_job_name, \$step_number, \$number_of_steps

Stop Job On Errors

Add REST Call:

- On the **+Add Step** drop-down, select **Add REST Call**.

i Note

You can use REST calls to perform a variety of operations. These may be against your own custom applications, your cloud provider, or other third party vendors. For instance, Oracle Cloud Infrastructure (OCI) provides a variety of endpoints, comprehensively documented at: <https://docs.oracle.com/en-us/iaas/api/>. It should be noted that this scope is **not** confined solely to OCI.

In this example, you will call OCI's object store REST calls to access their endpoint that reads a file from the object store.

- Specify the following field values on the **Add REST** dialog to must set up a Job step:
Method: *GET*.

This method retrieves data from the Oracle database. The *GET* method is used to fetch data, such as rows from a table or the result of a query. It does not modify any data.

You can select any of the following methods depending on the operation you want to perform:

- **POST:**
This method inserts new data or invokes a process that changes data. It submits data to the Oracle database, typically to create new records or run procedures that modify data.
- **DELETE:**
Removes data from the Oracle database. This method deletes specified resources, such as rows in a table, identified by parameters or path variables.
- **PATCH:** Partially updates existing data. This method applies partial modifications to a resource, such as updating specific columns of a row without replacing the entire record.

Url: Specifies the object storage location of the REST service you want to access, directing your request to the correct resource on the server. In this example, enter `https://objectstorage.us-ashburn-1.oraclecloud.com/n/*****/**/**/****.csv`.

Credential: Select any one of the credentials to authenticate your request, ensuring that you are authorized to access the REST API resources securely.

Note

There is no credential selected in this example, since the **Url** is a public bucket.

Body: This field specifies the data sent to the server, typically in **POST** or **PUT** requests. **GET** and **DELETE** requests do not have a body according to HTTP standards.

Headers: The headers should be supplied as a JSON object where each key/value pair of the JSON object is used as a header and value in the request. For example, `{"X-My-Header": "My header value"}`. This field specifies additional information about the request or the client to the server. These headers can specify metadata such as content type, authorization credentials, custom parameters, and other control information necessary for processing the request. Headers are typically used in **PUT**, **PATCH**, **POST**, and **DELETE** methods to convey such details.

Select **Stop Job On Errors**.

Click **Add Step**.

Add REST

Method *

GET

Url *

https://objectstorage.us-ashburn-1.oraclecloud.com/n

Credential

Body

Headers

Stop Job On Errors

Add Step Cancel

- **On Failure:** You will receive an email update or Slack notification when the job has failed.
- **On Exceeded Runtime:** You will receive an email update or Slack notification when the job has run longer than the maximum allowed or configured time limit and was stopped or timed out as a result.
- **On Rejected Rows:** You will receive an email update or Slack notification when the rows of data were not accepted or processed successfully during the data loading operation.

Note

- For the Jobs feature to send notifications, you need to configure the Simple Mail Transfer Protocol (SMTP) Settings in the Data Studio Settings.
- Refer to the [Send Email on Autonomous Database](#) chapter for more details.
- See [Data Studio Settings](#) for more information on configuration.

If you want to report via Email, specify the following field values:

- **Send Using:** Select *Email* from the drop-down.
- **Send to:** *abc@example.com*. Enter the email address where you want to receive the notifications.
Select the events on when you want to receive the notifications.

If you want to report via Slack, specify the following field values:

- **Send Using:** Select *Slack* from the drop-down.
- **Send to:** *******. Enter the channel ID of the Slack channel where you want to receive the notifications. It specifies the Channel ID in a `String` value. This parameter is mandatory. The Channel ID is a unique ID for a channel and is different from the channel name. In Slack, when you view channel details you can find the Channel ID on the `About` tab.
Select the events on when you want to receive the notifications.

Note

If you are running parallel jobs, you must ensure that the job notification is closed only after every job completes otherwise you'll get incorrect notifications.

Click **Next** to progress to the **Schedule** tab.

4. Create a Job Schedule:

- On the Schedule tab, you will define when to run the job and at what frequency.

Enable for Scheduling: Select this option to set up a schedule for running the job feature; that is, to poll the data source on a regular basis:

- In the time interval fields, enter a number, and select a time type and the days on which to poll the bucket for new or changed files. For example, to poll every two hours on Monday, Wednesday, and Friday, enter **2**, select **Hours**. You can select **All Days, Monday to Friday, Sunday to Thursday**, or **Custom** from **Week Days**

drop-down. The Custom field enables you to select **Monday, Tuesday, Wednesday, Thursday and Friday** in the appropriate fields.

- Select a start and end date with start and end time. If you don't select a start date, the current time and date are used as the start date. The end date is optional. However, without an end date, the live feed will continue to poll.

Click **Next** to progress to the **API Details** tab.

5. Run Jobs Manually:

You can run the Jobs feature after its created with the `DBMS_SCHEDULER` or API 's.

On the API Details tab of the Create Job dialog, you can view the following panes on the left navigation:

- **Run Job:** Run the specific job by copying the statements under REST API and submitting a GET request to the REST resource using cURL.

```
curl -u admin:<password>
      'https://*****.*****.oraclecloud.com:****/ords/
admin/_/db-api/latest/data-tools/jobs/run'
      -X POST -H 'Content-Type: application/json;charset=UTF-8' -d
      '{ "job_name": "My New
Job", "owner": "admin" }'
```

This is a sample code.

You can also run the job on the SQL worksheet by copying and pasting the following sample code you view under the SQL section:

```
begin c#***$service.dbms_data_tools.run_job('My New Job'); end;
```

The above sample SQL code runs the specific job feature. You can run this code on SQL worksheet to run the Jobs feature.

- **Job Activity Detail:** This option displays the REST APIs and SQL query to check the status of any ongoing job run. For example, if your job takes 10 hours, you can view if the job is in progress or not, it failed or not.

Note

Every time the job is run the entire flow of every step is started from scratch.



Create Job

1 General 2 Steps 3 Reporting 4 Schedule 5 API Details

After the Job is created, you can use the REST endpoint or SQL to invoke them

Run Job

REST API

Job Activity Detail

```
curl -u www:ab-<password> https://www.oracle.com:8443/o db-api/latest/data-tools/jobs/run/ -X POST -H 'Content-Type: application/json;charset=UTF-8' -d '{"job_name": "Test_job", "owner": "www"}'
```

SQL

```
begin c# www.run_job('Test_job'); end;
```

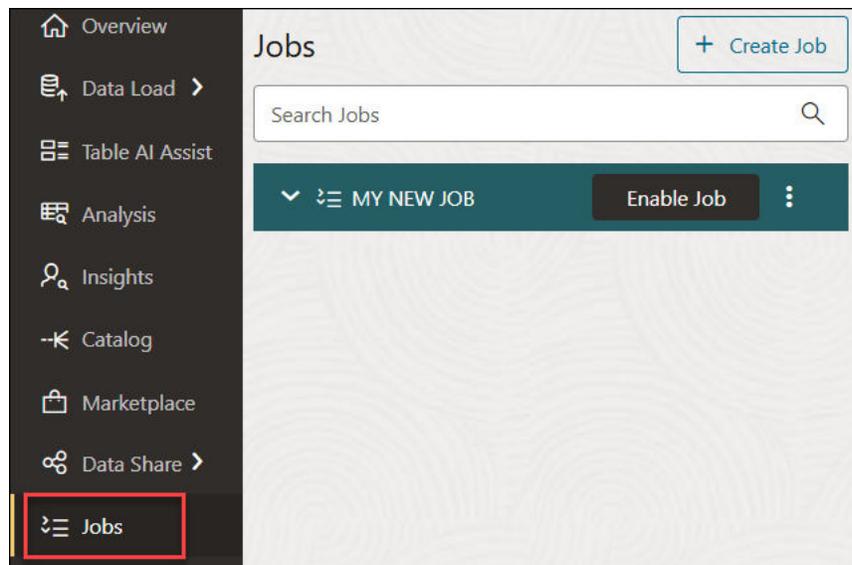
Show code

Back Next Create Cancel

Click **Create**.

You will receive a confirmation message saying that **My New Job** was created.

The newly created job will be listed on the **Jobs** page.



If you have configured Slack with the Jobs feature, you will also receive notification in Slack on the events you have configured in the above steps.

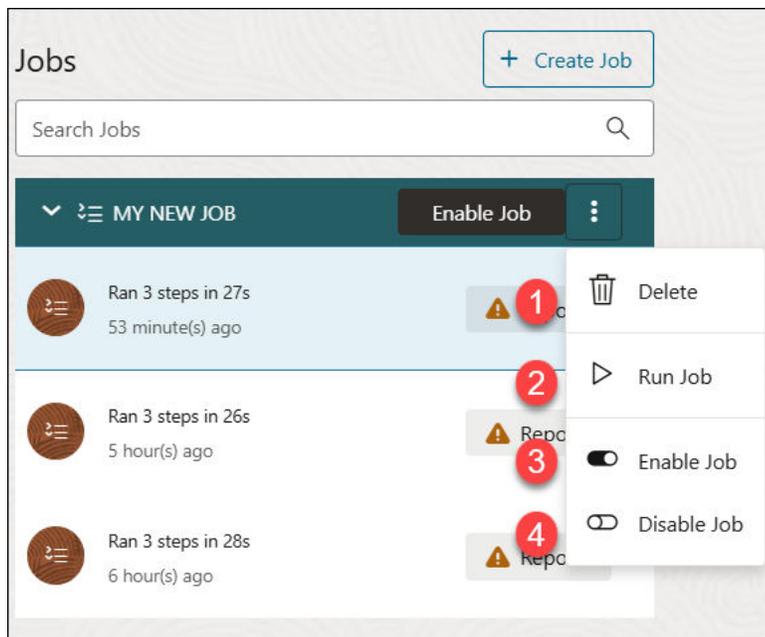
For more details on Slack configuration refer to the [DBMS_CLOUD_NOTIFICATION Chapter](#).

Manage and Monitor the Job

Once the job is scheduled and created, you can manage the job and monitor each step you run for the job via the **Report** feature.

Manage the Job

On the Jobs page, click the **Actions** icon besides the Job you created and perform the following actions:



- **Delete:** Select **Delete** to delete the job.
- **Run Job:** Select **Run Job** to run the job. This will drop the table you loaded previously, re-run the data load, and create a new table with the same data, re-run the Table AI Assist recipe and run the procedure again.
- **Enable Job:** Select **Enable Job** to enable the scheduling of the job you run. A confirmation message will be displayed that asks you if you want to run the job using the scheduler.
- **Disable Job:** Select **Disable Job** to disable the scheduling of the job you run.

Report Jobs

After creating a job, you can monitor its status and execution history using the **Report** feature.

Select **Report** besides each step to view details about its execution status.

You can view the following parameters with their values on the Report table:

Table 19-1 Report Parameters and their Descriptions

Report Parameters	Description
LOG_DATE	This field refers to the date and time when the specific job was logged in the system. It is stored in ISO-8601 format: YYYY-MM-DDTHH:MM:SSZ
ACTION	This field refers to specific actions or states related to job execution. Here are its values: <ul style="list-style-type: none"> RUN JOB: It refers to running a job immediately, regardless of its predefined schedule. START JOB: It refers to enabling a job so that it can run according to its schedule. By default, jobs are created in a disabled state and must be enabled using the Enable Job icon. Once enabled, the Scheduler will automatically start the job at its next scheduled time or when triggered by an event. FINISHED JOB: It refers to the state of a job after it has completed execution.
STEP TYPE	This displays the step you have added in the Job feature. It can be Data Load, Smart Table or a Procedure.
STEP NUMBER	This field displays the step number.
ROWS LOADED	This field refers to the number of data rows successfully inserted into a table or processed during the execution of a job.
ROWS REJECTED	This field displays the data records that were not successfully loaded or processed during a job execution due to various reasons such as data errors, constraint violations, SQL errors and insert or update failures.
ERROR MESSAGE	This field provides detailed information about issues encountered during job execution. These messages are crucial for troubleshooting and resolving problems that may have caused a job to fail or complete with warnings.
DETAILS	This field displays comprehensive overview of each job's execution.

DBMS_DATA_TOOLS_JOBS Package Reference

This chapter provides information about the packages you use with the Jobs feature in Data Studio. The [Summary of DBMS_DATA_TOOLS_JOBS Subprograms](#) topic also covers the procedures included in the DBMS_DATA_TOOLS_JOBS package.

Topics:

[Summary of DBMS_DATA_TOOLS_JOB Subprograms](#)

- [DBMS_DATA_TOOLS_JOB Procedures](#)

- [Summary of DBMS_DATA_TOOLS_JOB Subprograms](#)
The DBMS_DATA_TOOLS_JOB package provides a collection of scheduling functions and procedures you use with the Jobs feature in the Data Studio suite of tools.

Summary of DBMS_DATA_TOOLS_JOB Subprograms

The DBMS_DATA_TOOLS_JOB package provides a collection of scheduling functions and procedures you use with the Jobs feature in the Data Studio suite of tools.

The following table lists the DBMS_DATA_TOOLS_JOB subprograms and briefly describes them.

DBMS_DATA_TOOLS_JOB Package Subprograms

Table 19-2 DBMS_DATA_TOOLS_JOB Names and description for the DBMS_DATA_TOOLS_JOB Subprograms

Subprogram	Description
Create Job procedure	This procedure creates a new job.
Delete Job procedure	Deletes the job from the scheduler.
Rename Job procedure	Renames the job with the desired name.
Update Job procedure	Modifies the attributes of an existing job.
Run procedure	Runs a job immediately regardless of the schedule you define.
Stop Job procedure	Stops a running job. This terminates a job gracefully, and if necessary, forcibly stop it.

CREATE_JOB Procedure

This procedure creates a single job. If you create the job as enabled by setting the enabled attribute to TRUE, the Scheduler automatically runs the job according to its schedule. If you create the job disabled, the job does not run until you enable it with the Enable Job feature.

Syntax

```
dbms_data_tools_job.create_job(
    job_name          in varchar2,
    description       in varchar2,
    steps             in clob,
    owner             in varchar2 default null,
    start_date        IN TIMESTAMP WITH TIME ZONE DEFAULT
NULL,
    repeat_interval   IN VARCHAR2                DEFAULT
NULL,
    end_date          IN TIMESTAMP WITH TIME ZONE DEFAULT
NULL,
    job_class         IN VARCHAR2                DEFAULT
NULL,
    enabled           IN BOOLEAN                DEFAULT
FALSE,
    reporting         in clob default null);
```

CREATE_JOB Procedure Parameters

Table 19-3 CREATE_JOB Procedure Parameters

Parameter	Description
Job Name	The name to assign to the job. If <code>job_name</code> is not specified, an error is generated.
Description	This refers to additional information that is set for jobs.
Steps	<p>This refers to various tasks the Data Studio tool performs that comprises of a job. Steps is a JSON array. Each array element is a JSON object. Here are the JSON elements which apply to all objects:</p> <ul style="list-style-type: none"> <code>type</code> - string, declares the type of step that is performed: <code>data_load</code>, <code>smart_table</code>, <code>procedure</code>, <code>http</code>. <code>on_errors</code>- string <code>stop</code>: If an error is encountered, the job is stopped and the error is raised. <code>continue</code>: the error is logged, but processing continues with the next job step. <code>step_name</code> - descriptive name of the step to appear in logs <p>Each object type has some additional elements depending on the type.</p> <p><code>data_load</code>:</p> <ul style="list-style-type: none"> <code>data_load_name</code> - (mandatory) the name of a previously created data load to run <p><code>smart_table</code>:</p> <ul style="list-style-type: none"> <code>recipe_name</code> - (mandatory) the name of a previously created smart table recipe to run. <code>url</code> - (mandatory) REST URL to connect to. <code>method</code> - GET (default), POST, PUT, DELETE, or some other HTTP verb <p><code>procedure</code>:</p> <ul style="list-style-type: none"> <code>procedure</code> - (mandatory) name of procedure to execute. It may optionally be schema qualified, and may be a bare procedure or a package method. <code>arguments</code> - array of string, number or boolean arguments to supply to the procedure.
Owner	It is the schema in which the job is created. If you create a job without specifying a schema, the owner will be the user executing the <code>CREATE_JOB</code> procedure.

Table 19-3 (Cont.) CREATE_JOB Procedure Parameters

Parameter	Description
Start Date	<p>This attribute specifies the first date and time on which this job is scheduled to start. If <code>start_date</code> and <code>repeat_interval</code> are left null, then the job is scheduled to run as soon as the job is enabled.</p> <p>For repeating jobs that use a calendaring expression to specify the repeat interval, <code>start_date</code> is used as a reference date. The first time the job runs is the first match of the calendaring expression that is on or after the current date and time.</p> <p>The Scheduler cannot guarantee that a job executes on an exact time because the system may be overloaded and thus resources unavailable.</p>
Repeat Interval	<p>This attribute specifies how often the window repeats. It is expressed using the Scheduler calendaring syntax. See "Calendaring Syntax" for more information.</p> <p>A PL/SQL expression cannot be used to specify the repeat interval for a window.</p> <p>The expression specified is evaluated to determine the next time the window opens. If no <code>repeat_interval</code> is specified, the window opens only once at the specified start date.</p>
End Date	<p>This attribute specifies the date and time after which the job expires and is no longer run.</p> <p>The value for <code>end_date</code> must be after the value for <code>start_date</code>. If <code>end_date</code> is less than <code>start_date</code>, then an error will be generated. If <code>end_date</code> is the same as <code>start_date</code>, then the job will not execute and no error will be generated.</p> <p>If no value for <code>end_date</code> is specified, the job repeats forever unless you <code>stop job</code>.</p>
Job Class	The class this job is associated with.
Enabled	<p>This attribute specifies whether the job is created enabled or not. The possible settings are <code>TRUE</code> or <code>FALSE</code>. By default, this attribute is set to <code>FALSE</code> and, therefore, the job is created as disabled. A disabled job means that the metadata about the job has been captured, and the job exists as a database object. However, the Scheduler ignores the job and the job coordinator does not pick it for processing. In order for the job coordinator to process the job, the job must be enabled. You can enable a job by selecting the <code>Enable Job</code> icon.</p>

Table 19-3 (Cont.) CREATE_JOB Procedure Parameters

Parameter	Description
Reporting	<p>This attribute specifies job execution details via Report button in the Jobs feature. This is a JSON object similar to the reporting argument for <code>dbms_live_feed</code>. It can have the following elements:</p> <ul style="list-style-type: none"> <code>completed</code>: Targets to notify when a job is completed. <code>errors</code>: Targets to notify when some step of a job encountered errors. <code>failed</code>: Targets to notify when a job had some fatal error outside of the context of a step. <code>long</code>: Targets to notify when a job ran long. <p>Each of these four elements can specify an array of email address in the <code>smtp</code> element or slack channels in the <code>slack</code> element. The <code>long</code> element can have a <code>max_runtime_seconds</code> element which determines how long a job has to run before it is considered to have run too long.</p>

DELETE_JOB Procedure

This procedure deletes a job.

Syntax

```
dbms_data_tools_job.delete_job(job_name in varchar2, owner in varchar2
default null);
```

DELETE_JOB Procedure Parameters**Table 19-4 DELETE_JOB Procedure Parameters**

Parameter	Description
Job Name	The name to assign to the job. If <code>job_name</code> is not specified, an error is generated.
Owner	It is the schema in which the job is created. If you create a job without specifying a schema, the owner will be the user executing the <code>CREATE_JOB</code> procedure.

RENAME_JOB Procedure

This procedure renames a job.

Syntax

```
dbms_data_tools_job.rename_job(old_job_name in varchar2,
                               new_job_name in varchar2,
                               owner          in varchar2 default null);
```

RENAME_JOB Procedure Parameters

Table 19-5 RENAME_JOB Procedure Parameters

Parameter	Description
Old Job Name	The name of the existing job.
New Job Name	The name of the new job.
Owner	It is the schema in which the job is created.

UPDATE_JOB Procedure

This procedure updates any of the existing job attributes.

Syntax

```

dbms_data_tools_job.update_job(job_name in varchar2,
                                owner          in varchar2 default null,
                                description    in varchar2 default null,
                                steps         in clob default null,
                                start_date    IN TIMESTAMP WITH TIME ZONE DEFAULT
NULL,
                                repeat_interval IN VARCHAR2                DEFAULT
NULL,
                                end_date     IN TIMESTAMP WITH TIME ZONE DEFAULT
NULL,
                                job_class   IN VARCHAR2                DEFAULT
NULL,
                                enabled      IN BOOLEAN                  DEFAULT
NULL,
                                reporting    in clob default null);

```

UPDATE_JOB Procedure Parameters

Table 19-6 UPDATE_JOB Procedure Parameters

Parameter	Description
Job Name	The name of the Job to update.
Owner	It is the schema in which the job is to be updated.
Description	The updated description to apply.
Steps	The new steps to update to.
Start Date	The new start date.
Repeat Interval	The updated repeat interval.
End Date	The updated end date.
Job Class	The updated job class.
Enabled	It could be enabled or disabled.
Reporting	The updated reporting attribute.

Run Procedure

This procedure runs a job immediately.

If a job is enabled, the Scheduler runs it automatically. It is not necessary to call `RUN_JOB` to run a job according to its schedule. Use `RUN_JOB` to run a job outside of its normal schedule.

Syntax

```
dbms_data_tools_job.run(job_name          in varchar2,
                        owner              in varchar2 default null,
                        use_current_session in number   default null,
                        options            in clob      default null);
```

RUN Procedure Parameters

Table 19-7 RUN Procedure Parameters

Parameter	Description
Job Name	<p>A job name or a comma-separate list of entries, where each is the name of an existing job, optionally preceded by a schema name and dot separator.</p> <p>If you specify a multiple-destination job, the job runs on all destinations. In this case, the <code>use_current_session</code> argument must be <code>FALSE</code>.</p>
Owner	It is the schema in which the job is run.
Use Current Session	<p>This specifies whether or not the job run should occur in the same session that the procedure was invoked from.</p> <p>When <code>use_current_session</code> is set to <code>TRUE</code>:</p> <ul style="list-style-type: none"> You can test a job and see any possible errors on the command line. <code>RUN</code> can be run in parallel with a regularly scheduled job run. <p>When <code>use_current_session</code> is set to <code>FALSE</code>:</p> <ul style="list-style-type: none"> You need to check the job log to find error information. All relevant fields in <code>scheduler_jobs</code> are updated. <code>RUN</code> fails if a regularly scheduled job is running.
Options	

STOP_JOB Procedure

This procedure stops currently running jobs or all jobs in a job class.

After stopping the job, the state of a one-time job is set to `STOPPED`, whereas the state of a repeating job is set to `SCHEDULED` or `COMPLETED`, depending on whether the next run of the job is scheduled.

Syntax

```
dbms_data_tools_job.stop_job(job_name in varchar2,  
                             owner      in varchar2 default null,  
                             force     in boolean  default false);
```

STOP_JOB Procedure Parameters

Table 19-8 STOP_JOB Procedure Parameters

Parameter	Description
Job Name	Name of a job to stop. It is the name of an existing job, optionally preceded by a schema name and dot separator.
Owner	It is the schema in which the job is to be stopped.
Force	<p>If <code>force</code> is set to <code>FALSE</code>, the Scheduler tries to gracefully stop the job using an interrupt mechanism. This method gives control back to the slave process, which can update the status of the job in the job queue to stopped. If this fails, an error is returned.</p> <p>If <code>force</code> is set to <code>TRUE</code>, the Scheduler immediately terminates the job slave. Oracle recommends that <code>STOP_JOB</code> with <code>force</code> set to <code>TRUE</code> be used only after a <code>STOP_JOB</code> with <code>force</code> set to <code>FALSE</code> has failed.</p> <p>Use of the <code>force</code> option requires the <code>MANAGE_SCHEDULER</code> system privilege.</p>

The Data Share Tool

The Data Share tool allows you to share Oracle data and metadata with other databases and non-database tools. The Data Share page displays information about the different types of shares available in the Oracle Autonomous Database.

- [Overview of the Data Share Tool](#)
Oracle Autonomous AI Database enables you to create shares using the share tool.
- [Access and Enable the Data Share Tool](#)
To share or consume data using the Data Share tool, navigate to the Data Studio, launch the Data Share tool, and enable sharing.
- [Provide Share](#)
Use the Provide Share page to view the number of shares of the database, and the number of share recipients, search for a share entity, search for a recipient, create a new share, and create a new recipient.
- [Consume Share](#)
Once the providers share the objects, there are a few steps the recipients need to follow to consume the share.
- [Manage Shares with DBMS_SHARE](#)
In addition to the Data Share tool, Autonomous Database supports a fully scriptable workflow via the PL/SQL `DBMS_SHARE` package.
- [Data Share Limitations on Autonomous Database](#)
This section summarizes the limitations of Data Share (both Versioned and Live Share) included in the Oracle Autonomous Database.
- [Live Share Feature Slow Performance](#)
- [FAQs on the Data Share Tool](#)
The following are frequently asked questions on the Data Share tool.

Overview of the Data Share Tool

Oracle Autonomous AI Database enables you to create shares using the share tool.

Sharing objects requires two steps. The Provider provides data share for access and the Consumer consumes (or receives) access to the published shares. The provider creates a share with the objects to share in the desired cloud object location. The provider also adds the recipient. The recipient accepts and receives the configured shared objects from the provider for consumption.

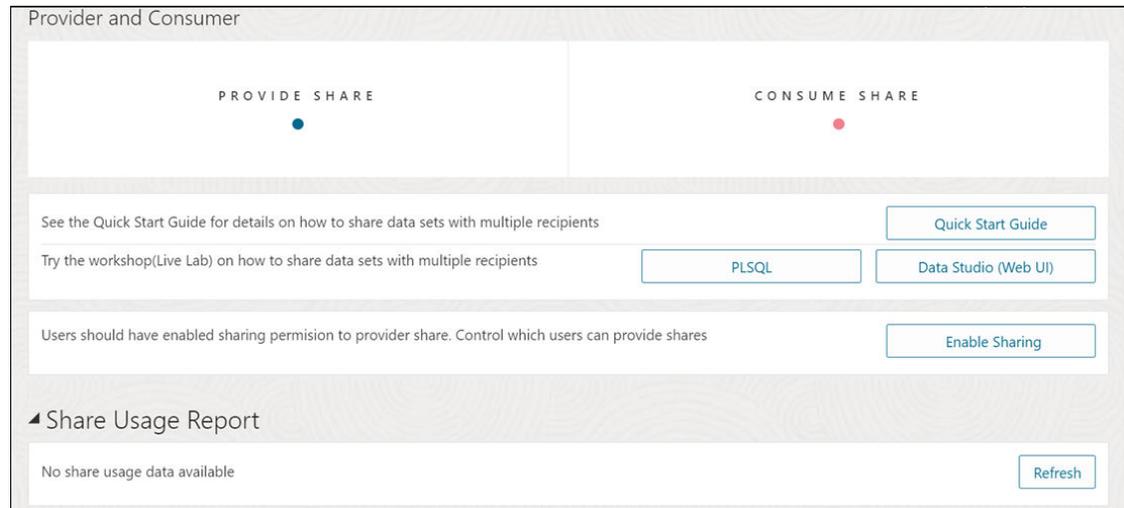
Note

You must have the correct privileges to create or consume a data share. In case the Data Share card is disabled, click on the tool tip and follow the steps for the administrator to grant you the required privilege.

Select the **Data Share** menu from the Data Studio suite in the Database Actions homepage to access this tool. This opens the **Data Share** homepage. It consists of widgets that enable you to provide and consume share objects.

Note

This is the homepage you view after you have enabled sharing and set Provider identification details.



Note

If you do not see the Data Share tool card, then your database user is missing the required DWROLE role.

Click [Quick Start Guide](#) to familiarize yourself with the Data Share tool.

Click [PLSQL](#) or [Data Studio \(Web UI\)](#) to try Data Sharing with PL/SQL or Data Studio without creating an account on Oracle Cloud tenancy.

Click **Enable Sharing** to grant sharing permission to you as a provider. See [Access and Enable the Data Share Tool](#) for more details.

The widgets are defined in the following sections:

- [Provide Share](#)
- [Consume Share](#)

Share Terminology

Provider: The Autonomous AI Database Serverless enables the provider to share existing objects. The share can contain a single table, a set of related tables, or a set of tables with some logical grouping. It could be a person, an institution, or a software system that shares the objects.

Example: An institution, such as NASA, that makes a data set available via data.gov.

Recipient: A Share recipient is an entity that associates an individual, an institution or a software system that receives a share from a provider. A recipient can have access to multiple shares. If you remove a recipient, that recipient loses access to all shares it could previously access.

Example: An external system, such as Microsoft Power BI, that supports the Delta Sharing REST API.

Share: A Share is a named entity in the provider's instance. It can be a group of datasets shared as a single entity.

Example: A SALES table that needs to be shared within an organization.

Overview of Providers and Recipients

A Data Share is the logical container that contains objects (such as tables) that share recipients will get access to a share and all tables within this share. A Data Share also implements security mechanisms on a high object level which simplifies the authorization for a set of individual objects. A provider creates and publishes share of a versioned type and a live share type. The recipient is given access to a share. The provider can modify shares (both data and metadata) after the provider publishes the share to the recipients.

Use Case of Data Share

A marketing agency can share sales information with multiple interested parties. The Data Analysis tool analyses the data, generates insights and then the application shares the information with interested parties.

How the Data Share tool works?

Data is made accessible by the data sharing provider (that is an Oracle Autonomous AI Database) to the data sharing recipient at query time in parquet format for a versioned share. A live share uses cloud links and can only be consumed in an Oracle database. The provider can only share data which they have access to when they log into an Autonomous AI Database instance.

As a data provider, you create a share and select other additional entities to share. The Oracle Data Sharing for general recipients is based on the open delta sharing standard protocol, providing a simple REST-based API to share data in parquet format. For near real-time access to shared data, customers can use Live Shares accessed using the consumer's ADB-S instance.

The Autonomous AI Database Serverless Versioned Sharing protocol works as follows:

- The provider creates and publishes a share that can be shared with one or multiple recipients. Every recipient will get a personal activation link to download their own JSON profile with the necessary information to access their share.
- The versioned share recipient registers with the share server by entering the URL for the end point along with a client ID, secret key and a bearer token.
- The versioned share recipient retrieves data from the share by calling the `/shares/./tables/./query` endpoint to obtain a list of URLs. The recipient then sends a GET request on these URLs to obtain the parquet files.

The Autonomous AI Database Serverless Live Sharing protocol works as follows:

- For a Live Share, the intended recipient will copy the sharing ID from the consumer page and publish the share that can be shared with recipients. This is the case when provider shares to only one database.
- A provider can share also share to **ALL_REGIONS**, **ALL_TENANCY**, or **ALL_COMPARTMENTS**.

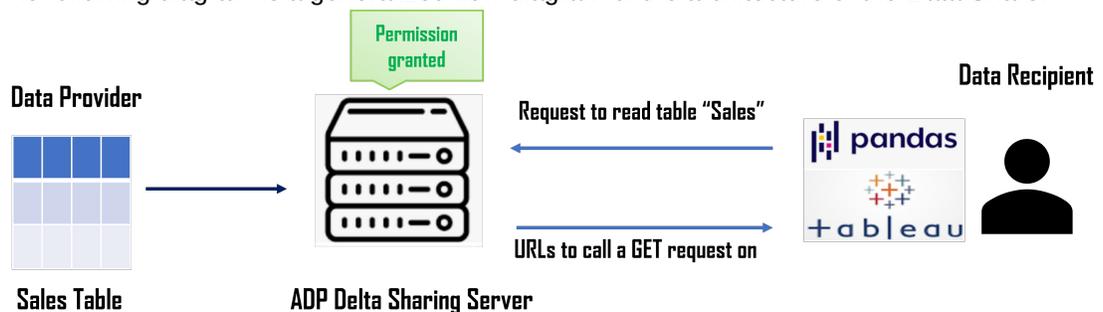
Features of Autonomous AI Database Serverless Share

With Autonomous AI Database Share you can:

- Share objects easily across Autonomous AI Database and all tools or APIs that support the open delta sharing protocol.
- Share versioned data with many recipients without data replication for all recipients.
- Establish secure and centrally managed data sharing and collaboration within and across organizations.

Share Architecture

The following diagram is a generalized flow diagram of the architecture of the Data Share.



Prerequisites for Share Providers

Here are some prerequisites for a share provider to use the share tool:

- For a versioned share, you must have read and write access to a bucket to store or cache your shares.
- The schema you wish to use to create and publish shares must be enabled by an `ADMIN` user.

Prerequisites for Share Recipients

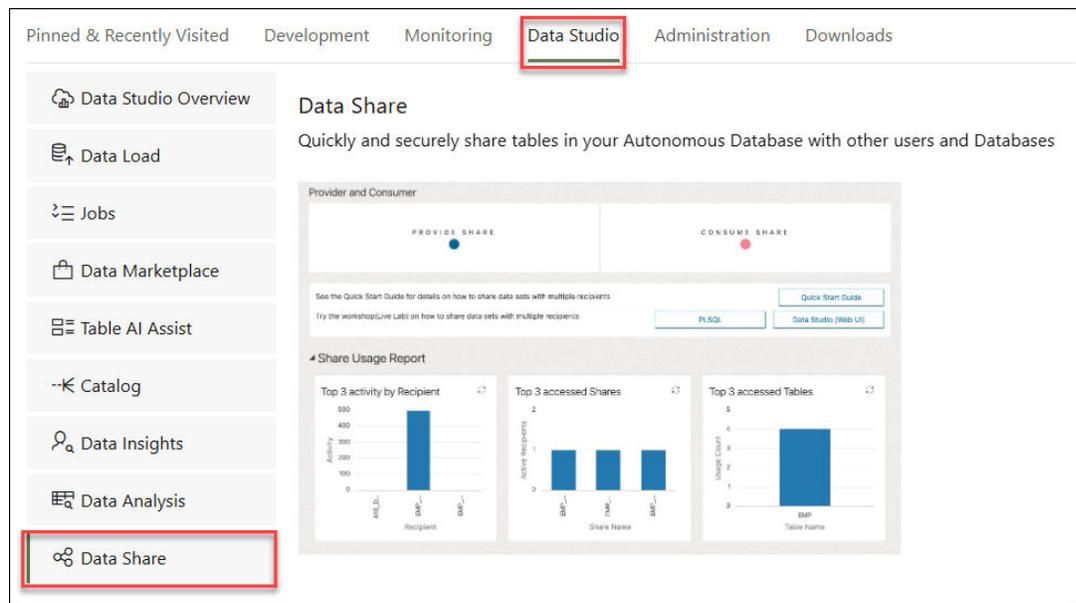
The share recipient must have a valid email address a provider can use to register the recipient to use the share tool. Oracle Data Share allows to share the information about a recipient's activation link by email.

Access and Enable the Data Share Tool

To share or consume data using the Data Share tool, navigate to the Data Studio, launch the Data Share tool, and enable sharing.

Follow these steps:

1. Navigate to the **Launchpad** on your Database Actions instance.
2. Click the **Data Studio** tab and select the **Data Share** pane.



3. Clicking the Data Share pane opens a Provider and Consumer page where you can view **Provide Share** and **Consume Share** widgets.



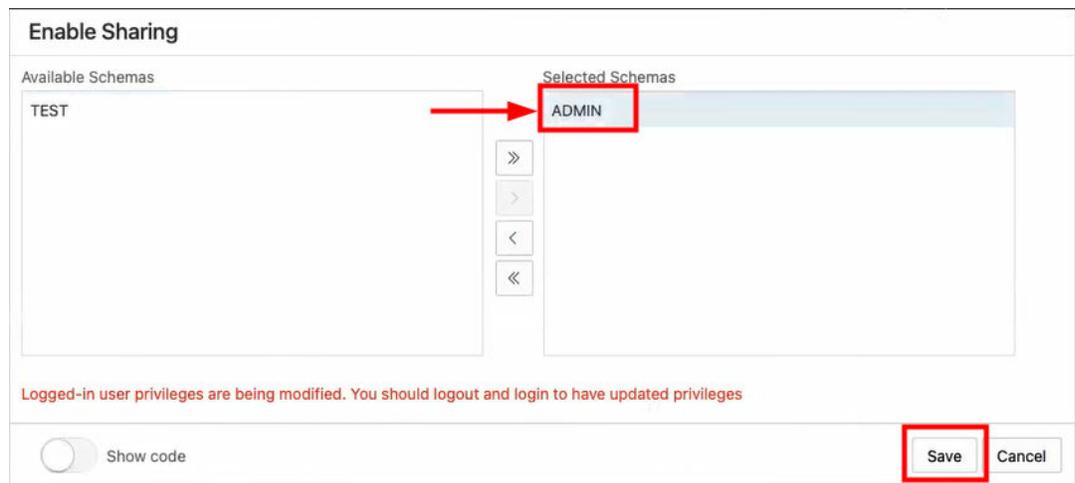
Click **Provide Share**.

4. On the **Provide Share** page, click **Enable Sharing**.

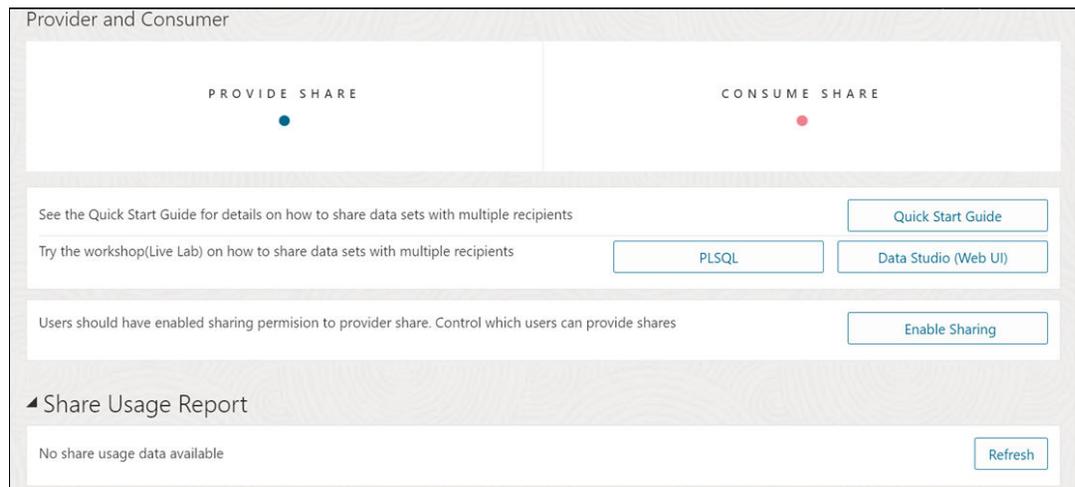


Clicking the **Enable Share** icon takes you to the Enable Sharing screen where you must select the user schema you want to enable and move it from the Available Schemas column to the Selected Schemas column.

5. Click **Save**.



6. Log out and log in again to update privileges.
7. You now view the following **Provider** and **Consumer** page.



You have now enabled the share. Click **Provide Share**. On the Provide Share page, click **Provider Identification** to create a Provider ID. This provides information to the recipient on how to identify you.

See [Data Studio Settings](#) on configuring details of the provider.

You can now start sharing your data.

Provide Share

Use the Provide Share page to view the number of shares of the database, and the number of share recipients, search for a share entity, search for a recipient, create a new share, and create a new recipient.

The Provide Share page consists of widgets Shares and Recipients and icons to create Shares and Recipients. The page is described in the following sections:

- [Provide Share Overview](#)
Use the Provide Share to create and publish shares.

- [Provide Versioned Share](#)
In this type of share, data is shared and published as well-defined, known, as-of-snapshots. At the time of publication, the tool generates and stores the data share as parquet files in the specified bucket. The recipient can directly access the share in the object store.
- [Provide Live Share](#)
In this type of share the recipient accesses data directly from the Oracle table or view. This share gives the recipient the latest data. Live Share works by querying data over a cloud link, which is implemented on a database link. The database link allows you to query objects over the network on a foreign machine and schema. The advantage of this mode is that the data is up to date as of the time of query.
- [View Share Entity Details](#)
Use the **Actions** icon at the right of the share entity entry to view details about the live share or delta share entity you create.
- [Create Share Recipient](#)
Use the **+ Create Share Recipient** icon on the Provide Share page to create a consumer of the data share. When there is already a recipient created, use the **+ Create Recipient** icon on the Recipients widget of the Provide Share page to create a consumer of the data share.
- [View Share Recipient Entity Details](#)
Use the **Actions** icon at the right of the share recipient entity entry to view details about the live share or delta share recipient entity you create.
- [Provide Share Overview](#)
The Provide Share page enables you to view, create, edit, and access information on the shares and recipients you create from this page.
- [Provide Versioned Share](#)
You can share data as a set of distinct versions, such as at the end of each day. The recipient will only see changes to the data when you publish a new version. As a versioned share Provider, you must create an OCI native credential and associate the bucket's URL with the credential.
- [Provide Live Share](#)
You can share data as of the latest database commit with Autonomous Databases in the same region. The recipient will always see the latest data.
- [View Share Entity Details](#)
To view details about an entity, click the **Actions** icon at the right of the Share entity entry, then click **View Details**.
- [Create Share Recipient](#)
On the Provide Shares page, click the **Recipients** widget and select **+ Create Recipient** to create a new Share Recipient.
- [View Share Recipient Entity Details](#)
Click the **Actions** icon at the right of the Share Recipient entity entry, then click **View Details**.

Provide Share Overview

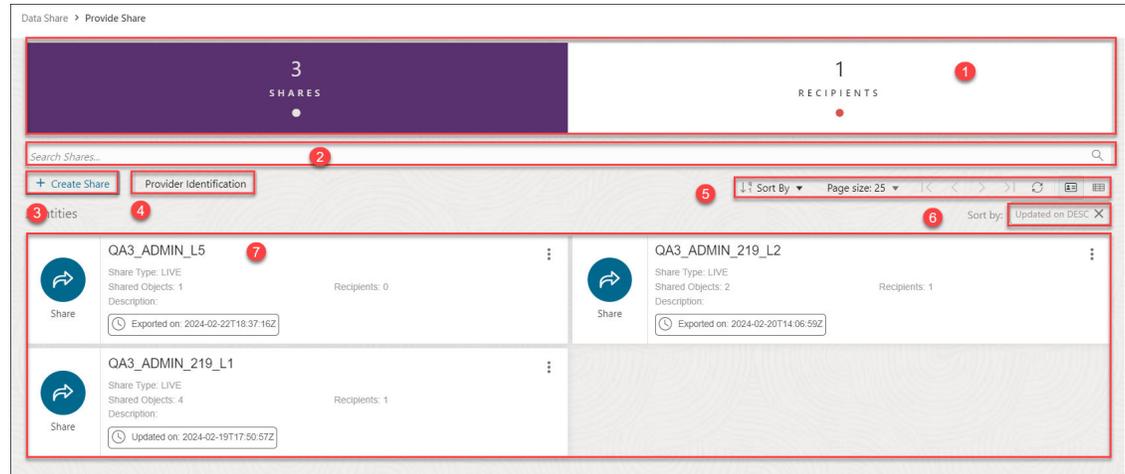
The Provide Share page enables you to view, create, edit, and access information on the shares and recipients you create from this page.

Click the **Provide Share** menu under Data Share in the Data Studio menu to reach the Provide Share page. You can switch the display of the Provide Share screen based on the widget you select from the Shares and Recipients section.

The search field and the display area of the **Provide Share** page varies when you switch between the **SHARES** and **RECIPIENTS** widgets.

Shares Page

The following image displays the Provide Share page when you select the **SHARES** widget.



The Provide Share page when you select the Shares widget consists of the following:

1. The top section of the Provide Share page displays Shares and Recipients section with two widgets. The widgets are defined in the following sections:
 - **SHARES**- Select this widget to view, search and perform actions on the Shares you create.
 - **RECIPIENTS**- Select this widget to view, search and perform actions on the Share Recipients you create.
2. **Search Shares** field
Selecting Shares widget enables you to search for the share you create. You can click the field and type or paste the name of the share or the recipient you are looking for. Click the magnifier icon to return the search input. Click **X** to clear your searches in the Search Shares field.
3. **+ Create Share** button
You can use the **Create Share** wizard to define data shares based on object storage data and associate them with recipients. A single data share can consist of multiple shared objects that are shared with one or multiple recipients.

You can create a new share and a new recipient from this button. Refer to the [Provide Live Share](#), [Provide Versioned Share](#) and [Create Share Recipients](#) section to explore more on what each button does.
4. **Provider Identification** button : See [Data Studio Settings](#) to view more about this icon for a first time user. If you have already configured SMTP, you can edit the Provider ID or set SMTP using this button.
5. **Toolbar**
The toolbar consists of the following buttons:
 - **Sort By**
To select sorting values, click the **Sort By** button to open the list of options. Then click the **Ascending** or **Descending** icon next to one or more of the sorting values.

For example, if you select the **Ascending** icon next to **Entity name** and the **Descending** icon next to **Entity type**, the entities will be sorted in alphabetical order by entity name and then in reverse alphabetical order by entity type.

Click **Reset** in the list to clear the choices in the list.

The sorting values you choose are listed next to the **Sort by** label beneath the toolbar. Click the **X** icon on a sorting value to remove it.

- **Page size**
By default, up to 25 entities are displayed on the page. If you want more entities on a page, select a number from this list.
- **Previous and Next**
If the search results are displayed on multiple pages, click these buttons to navigate through the pages.
- **Refresh**
Click to refresh the entities shown on the page, based on the current search string.
- **Entity view options**
Choose one of these three options to set how entities are displayed on the page.

Click **Open Card view** to display entities as card arranged into one or two columns; click **Open Grid View** to display entities as rows in a table; or click **Open List View** to display entities in a single column of panels.

6. Sort by settings

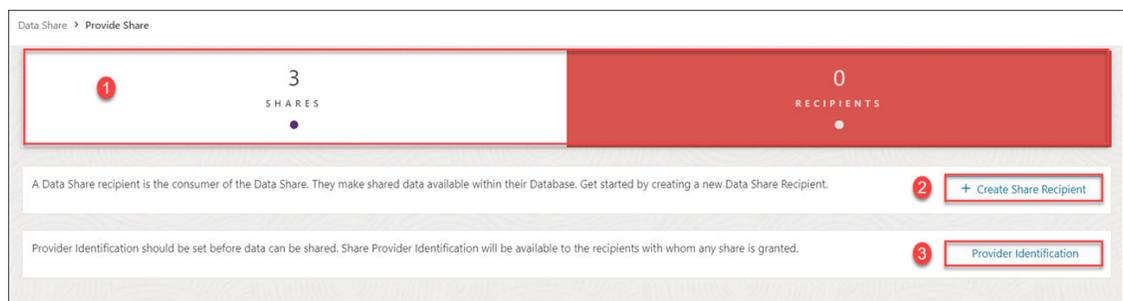
When you set sorting values by using the Sort By control in the toolbar, the settings are displayed in small boxes beneath the toolbar. You can delete a setting by clicking the X icon in the box. Or you can change the settings by returning to the Sort By control in the toolbar.

7. Display area

The area below the Search field displays the entities returned by a search. If you select the Shares widget, you will view the Share entities in the display area. If you select the Recipient widget, you will view the Share Recipient entities in the display area. This area shows all providers you are subscribed to.

Recipients Page

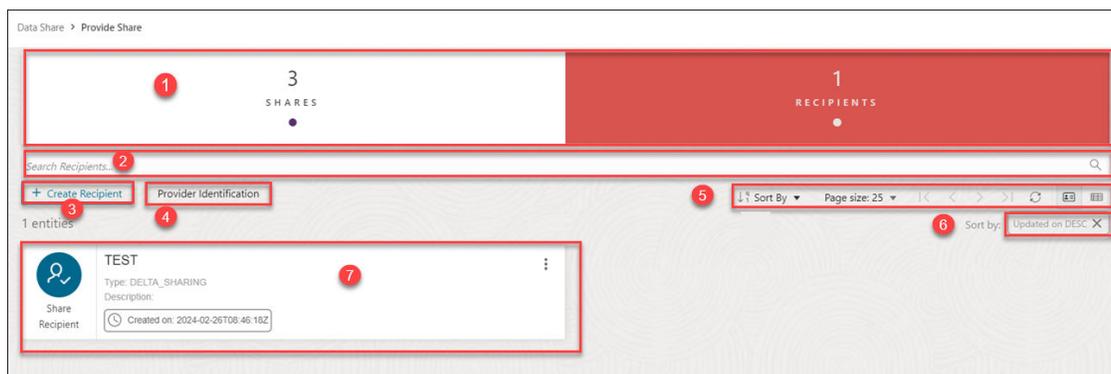
The following image displays the Provide Share page when you select the RECIPIENTS widget and **there is no RECIPIENT created**



Click **+ Create Share Recipient** to create a share recipient. See [Create Share Recipient](#) for more details.

Click **Provider Identification** to specify the name and description to identify how the recipients will view you. You can also configure SMTP settings.

The following image displays the Provide Share page when you select the RECIPIENTS widget and *there is minimum one RECIPIENT already created*



The Provide Share page consists of the following:

1. The top section of the Provide Share page displays Shares and Recipients section with two widgets. The widgets are defined in the following sections:
 - **SHARES**- Select this widget to view, search and perform actions on the Shares you create.
 - **RECIPIENTS**- Select this widget to view, search and perform actions on the Share Recipients you create.
2. **Search Recipients** field
Searching for a share or recipient depends on the widget you select from the Shares and Recipients section. If you select the Recipients widget, you can search for the recipients you create. You can click the field and type or paste the name of the share or the recipient you are looking for. Click the magnifier icon to return the search input. Click **X** to clear your searches in the Search Shares or Search Recipients field.
3. **+ Create Recipient** button
You can create a new share and a new recipient from this button. Refer to the [Provide Live Share](#), [Provide Versioned Share](#) and [Create Share Recipients](#) section to explore more on what each button does.
4. **Provider Identification** button: See [Data Studio Settings](#) to view more about this icon for a first time user. If you have already configured SMTP, you can edit the Provider ID or set SMTP using this button.
5. **Toolbar**
The toolbar consists of the following buttons:
 - **Sort By**
To select sorting values, click the **Sort By** button to open the list of options. Then click the **Ascending** or **Descending** icon next to one or more of the sorting values.

For example, if you select the **Ascending** icon next to **Entity name** and the **Descending** icon next to **Entity type**, the entities will be sorted in alphabetical order by entity name and then in reverse alphabetical order by entity type.

Click **Reset** in the list to clear the choices in the list.

The sorting values you choose are listed next to the **Sort by** label beneath the toolbar. Click the **X** icon on a sorting value to remove it.
 - **Page size**
By default, up to 25 entities are displayed on the page. If you want more entities on a page, select a number from this list.

- **Previous and Next**
If the search results are displayed on multiple pages, click these buttons to navigate through the pages.
 - **Refresh**
Click to refresh the entities shown on the page, based on the current search string.
 - **Entity view options**
Choose one of these three options to set how entities are displayed on the page.

Click **Open Card view** to display entities as card arranged into one or two columns; click **Open Grid View** to display entities as rows in a table; or click **Open List View** to display entities in a single column of panels.
6. **Sort by settings**
When you set sorting values by using the Sort By control in the toolbar, the settings are displayed in small boxes beneath the toolbar. You can delete a setting by clicking the **X** icon in the box. Or you can change the settings by returning to the Sort By control in the toolbar.
7. **Display area**
The area below the Search field displays the entities returned by a search. If you select the Shares widget, you will view the Share entities in the display area. If you select the Recipient widget, you will view the Share Recipient entities in the display area.

Provide Versioned Share

You can share data as a set of distinct versions, such as at the end of each day. The recipient will only see changes to the data when you publish a new version. As a versioned share Provider, you must create an OCI native credential and associate the bucket's URL with the credential.

On the Provide Share page, select **+ Create Share** from the Create menu. This brings up the **Create Share** wizard.

Note

You do not need to configure the Provider Identification details to create a versioned share.

1. On the Create Share wizard, in the **Name** field of the General tab, enter a name for the Share. For example: *My_Share*.
In the **Description** field, enter a description for the link. For example: *Weekly Sales report*.
Select **Next** to progress to the Publish Details tab of the Create Share wizard. You can alternatively click on the Publish Details tab to progress to the Publish Details screen of the wizard.

Create Share

1 General 2 Publish Details 3 Select Tables 4 Recipients

Name
My_Share

Description

Back Next Create Cancel

2. In the Publish Details Tables tab of the wizard, select **SHARE VERSIONS USING OBJECT STORAGE**.

Create Share

1 General 2 Publish Details 3 Select Tables 4 Recipients

SHARE VERSIONS USING OBJECT STORAGE
Share data as a set of distinct versions, such as end of each day. The recipient will only see changes to the data when you publish a new version.

SHARE LIVE DATA USING DIRECT CONNECTION
Share data as of the latest database commit with Autonomous Databases in the same region. The recipient will always see the latest data.

Choose an OCI native credential based cloud location where the share will be published *

testsci
You have access to Read, Write and Delete Create Cloud location

Enable for Scheduling

Every 12 Hours on all Days

Start Date
09/18/2025 17:34

End Date
09/25/2025 17:34

Back Next Create Cancel

Select a cloud location where you want the share to be published. If you already have an OCI cloud storage location available, use the **Select Cloud location** drop-down and select any available Cloud location from the drop-down. You will view a successful message if you have access to read, write, or delete in that cloud location.

Note

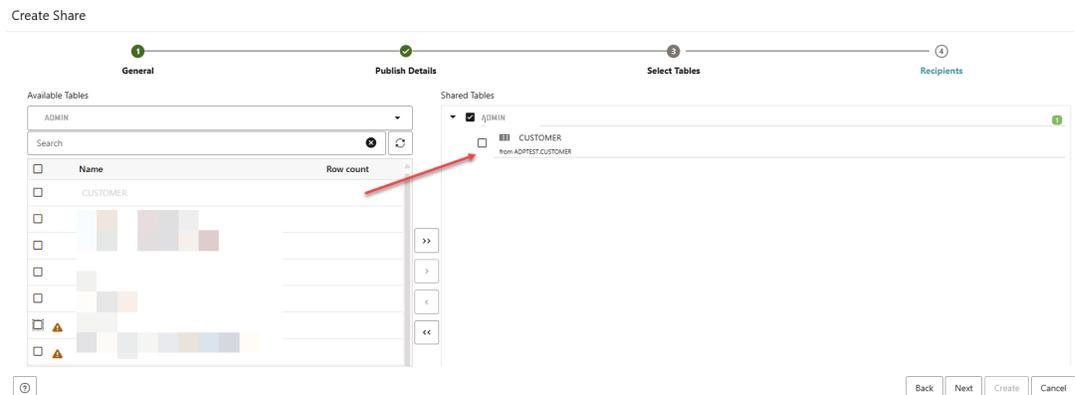
Select **Create Cloud location** and follow the steps described in the [Create Oracle Cloud Infrastructure Native Credentials](#) if you do not have a cloud location available.

For sharing versioned data, you can enable the schedule for share. Select **Enable For Scheduling** option to set up a schedule for sharing. In the time interval fields, enter a number, and select a time type and the days on which to poll the bucket for new or changed files. For example, to poll every two hours on Monday, Wednesday, and Friday, enter **2**, select **Hours**, and then select **Monday, Wednesday, Friday** in the appropriate fields. You can select **All Days, Monday to Friday, Sunday to Thursday**, or **Custom** from the Week Days drop-down. The Custom field enables you to select Monday, Tuesday, Wednesday, Thursday and Friday in the appropriate fields.

Select a start and end date. If you don't select a start date, the current time and date are used as the start date. The end date is optional. However, without an end date, the share will continue to poll. This is an optional step.

Click **Next** to progress to the Select Tables tab of the Create Share wizard.

- On the **Select Tables** tab of the wizard, select the schema from the drop-down menu and click the table you wish to share from the Available Tables. Choose any of the available options:
 - >: This option enables you to move the table to Shared Tables.
 - <: To remove the selected table from Shared Tables select this option.
 - >>: This option allows you to move all the tables to the Shared Tables screen.
 - <<: To remove all the selected tables from Shared Tables select this option.



Click **Next** to proceed to the Recipients tab of the Create Share wizard.

Note

You may find it convenient to click on the specific tab to move back and forth. You can update changes on any screen this way.

- On the Recipients tab of the Create Share wizard, select available Recipient from the drop-down.



Select **Notify Recipients When Available** when you want to notify the recipients when they are available. This can be enabled when SMTP is configured to send an email notification to the recipient.

- In case you want to create a new recipient, select **New Recipient**. Selecting New Recipient opens a **Create Share Recipient** wizard. See [Create Share Recipient](#).

Note

Ensure there are no spaces in the name of the recipient. This field does not support duplicate recipient names. If you try creating a recipient with a duplicate name, the creation of the share fails with the "Name is already used by an existing object" error.

Click **Create** to finish creating the share recipient. The newly added recipient is displayed in the list of recipients. Click **Cancel** to cancel the ongoing process of recipient creation.

6. After you create a new share recipient, in the granted recipient's list you can:
 - Select **copy profile activation link to clipboard** icon to copy the activation link.
 - Select **Email recipient profile download link** icon to mail the profile link to the recipient you select in this step. Once you select it, the Share tool triggers an activation mail with the profile link to the recipient's email address. Click **Send** to share the profile link with the recipient. See *About the Consume Share Page* to know more about the activation mail and how to proceed as a recipient once you receive the activation mail.
 - Click **deselect Recipient** icon to remove the selected recipient.

Select **Create** to create the share. Note that the status of the share changes from unpublished to published depending on the size of the table in the share. You will receive a successful share notification message at the top right of the display that says the share is successful. The Provide Share page displays the share along with its details such as the entity type, owner, shared objects and the recipients.

Note

The Data Share tool can create only two delta share profiles for the same recipient. If you try to create a third profile, the tool invalidates the first profile, and the newly created profile becomes active.

Note

A versioned share will show different colors and states depending upon what state the share is in.

Provide Live Share

You can share data as of the latest database commit with Autonomous Databases in the same region. The recipient will always see the latest data.

To provide access to live shared data, you must define data shares and grant recipients access to consume them. Use the **Create Share** wizard to create and publish shares for live data.

Note

You must configure the Provider Identification details to create a live share.

On the Provide Share page, select **+ Create Share** from the Create menu. This brings up the Create Share wizard.

1. On the Create Share wizard, in the **Name** field of the General tab, enter a name for the Share. For example: *My_Share*.
In the **Description** field, enter a description for the data you are sharing. For example: *Weekly Sales report*.

Select **Next** to progress to the Publish Details tab of the Create Share wizard. You can alternatively click on the Publish Details tab to progress to the Publish Details screen of the wizard.

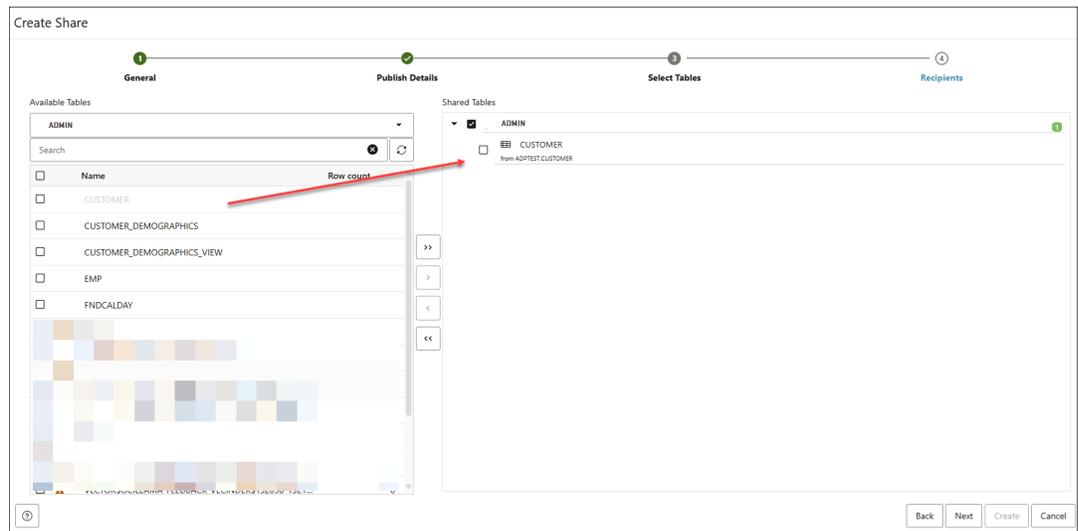
2. In the Publish Details Tables tab of the wizard, select **SHARE LIVE DATA USING DIRECT CONNECTION**.

If you have not configured the provider information details, click **Set Provider Identity** to configure the details of the provider before you share the data. See the **Share provider Identity** tab of the [Data Studio Settings](#) wizard for more information on configuration.

Click **Next** to progress to the Select Tables tab of the Create Share wizard.

3. On the **Select Tables** tab of the wizard, select the schema from the drop-down menu and click the table you wish to share from the Available Tables. Choose any of the available options:
 - **>**: This option enables you to move the table to Shared Tables.
 - **<**: To remove the selected table from Shared Tables select this option.

- >>: This option allows you to move all the tables to the Shared Tables screen.
- <<: To remove all the selected tables from Shared Tables select this option.



Note

If a live share data provider is sharing multiple tables which would normally be joined together, it is recommended that the producer create a view that performs the joins and then only shares the view.

Click **Next** to proceed to the Recipients tab of the Create Share wizard.

Note

You may find it convenient to click on the specific tab to move back and forth. You can update changes on any screen this way.

- On the Recipients tab of the Create Share wizard, select available Recipients from the drop-down. The list of recipients available in the drop-down depends on the scope of the share. You can select any or all the following values depending on with whom you intend to share the data:
 - **MY_REGION**: Access to the data share is allowed for databases in the same region as the data share provider.
 - **MY_TENANCY**: Access to the data share is allowed for databases in the same tenancy as the data share provider.
 - **MY_COMPARTMENT**: Access to the data share is allowed for databases in the same compartment as the data share provider.

- In case you want to create a new recipient, select **New Recipient**. Selecting New Recipient opens a **Create Share Recipient** wizard. Enter the recipient's Name, Description (optional), Sharing ID, Email in their respective fields.

Note

- Ensure there are no spaces in the name of the recipient. This field does not support duplicate recipient names. If you try creating a recipient with a duplicate name, the creation of the share fails with the "Name is already used by an existing object" error.
- A Sharing ID is a unique provider for your Autonomous Database. The Data Share tool uses it to share data with you. Copy the Sharing ID from the Consume Share page to the clipboard and paste it in the Sharing ID field of the **Create Share Recipient** wizard.

Click **Create** to finish creating the share recipient. The newly added recipient is displayed in the list of recipients. Click **Cancel** to cancel the ongoing process of recipient creation.

- Select **Create** to create the share. You will receive a successful share notification message at the top right of the display that says the share is successful. The Provide Share page displays the share along with its details such as the entity type, owner, shared objects and the recipients.

Click the name of the entity or click **Action** to view details about the share entity. See [View Share Entity Details](#).

After share is created, you can view the new Share entity in the Provide Share page.

View Share Entity Details

To view details about an entity, click the **Actions** icon at the right of the Share entity entry, then click **View Details**.

For all entities, the details include General, Selected Tables, Publishing details, Recipients, Versions, Log Details, Lineage and Impact sections.

General

The General tab displays the Name and description of the share.

Objects

You can view the Selected Tables tab that displays the list of tables you wish to share.

Publish Details

This tab displays the publishing details you configure during creation of the share. It displays whether you select live data share or share versioned data.

Recipients

You can review the list of recipients with whom you intend to share objects. It displays the name of the recipient with their mail address.

Jobs

See the Jobs option below to view information about it.

Log Details

You can view the detailed status of the share in this tab. It provides information such as, the time, Status, Log Level, and Details of each activity that takes place during the running of the job, i.e., creation of the share. It also displays if any errors are encountered during the creation of the share.

Lineage

Lineage displays all known information about the dependencies of the entity, and therefore how the entity was created and how it is linked to other entities.

For example, for a share you create in the database, the lineage is just the cloud location. For a share that you create by adding files from cloud storage, the lineage includes the cloud location of the file you share.

Pointing to the name of an item in the lineage displays the share name, the application that creates it, the type of entity, the path to it, and the schema it is in.

Arrows point from an entity to the entity that it derives from. For example, for a share you create, an arrow points from the cloud location to the share. If you point to an arrow, then a Links Information box appears that shows information about the relationship between the two entities.

To view more details about an item, click the **Actions** icon for the item, then click **Expand**. For a cloud location, the files in the location are displayed. Pointing to the name of the file displays the name, application, type, path, and schema of the file. To collapse the display, click the Actions icon, then click Collapse.

Impact

Impact shows all known information about the downstream use of an entity, and therefore how a change in the definition of an entity may affect other entities that depend on it.

For a specific share entity, you can perform the following actions using the Actions context menu:

- **View Details:** See [View Share Entity Details](#).
- **Manage Versions:** The version information of the selected share is displayed. Each version has the following characteristics:
 - Version name
 - Publication Date
 - Num Objects
 - Num new Objects
 - Status: CURRENT and RETIRED
 - Actions: Delete the version or Make it current

You can also **Delete unused versions**.

- **Objects:** Opens Select Tables dialog of the specific share where you can remove tables from the list of Shared Tables list or add tables from the list of Available Tables list to the list of Shared Tables.
- **Recipients and Profiles:** Opens the Recipients and Profiles dialog which enables you to create a new share recipient. You can also revoke the recipient right from the share by clicking X next to the recipient name. The Recipients and Profiles dialog enables you to update the list of recipients, copy profile activation link, send an activation mail to the selected recipient.
- **Jobs:** Opens the Jobs table and displays information related to it. A job is created whenever something needs to happen to a share. This includes the following:
 - PUBLISH VERSIONED SHARE
 - PUBLISH LIVE SHARE
 - DROP SHARE VERSIONS

Only one job can run at a time on a single share. E.g. if the user calls `publish_share` while a `PUBLISH` job is already running, then the second job will have to wait. The `STATUS` column reflects the same.

The `JOB TYPE` column displays the method of parquet files generation (only for a versioned share). It's values can be `ODI` or `DBMS_CLOUD`.

- **Scheduling:** This tab displays the scheduling details you configure during creation of the share.
- **Publish:** Select **Publish** to publish the updated share.
- **Unpublish:** Select **Unpublish** to unpublish the share. This option prevents all recipients from accessing a single share, but lets them continue to access any other shares. You can restore access by publishing share again.
- **Delete:** Deletes the Share Provider Entity.

Create Share Recipient

On the Provide Shares page, click the **Recipients** widget and select **+ Create Recipient** to create a new Share Recipient.

Share Recipients are those with whom we intend to share objects with. You can share objects securely with users outside of Oracle workspace. Apart from creating recipients from the Create Share wizard, you can also create a share recipient from the Create Recipient icon on the Provide Share page and verify the details you configure while its creation.

1. On clicking the **+ Create Recipient** icon, a Create Share Recipient dialog box appears.

2. On the General tab, specify the following fields:

- **Name:** Enter the name of the Share Recipient. For example, *Marketing_team*.

Note

Ensure there are no spaces in the name of the recipient. This field does not support duplicate recipient names. If you try creating a recipient with a duplicate name, the creation of the share fails with the "Name is already used by an existing object" error.

- **Email:** Enter the email ID of the recipient. For example, *marketing@oracle.com*.
- **Description:** Add a description. This is optional.

Under **Direct Database Connect** section you can select any of the available options:

- **Select Database or Compartment:** You can select the Database or Compartment from the drop-down.
- **Sharing ID:** An intended Recipient should copy this ID from the consumer page and send it to the Producer to be used here. The Share Provider wont have access to the consumer page. **This field is required for Live Share Recipients only.**

On the User token lifetime section, enter any value in the `TOKEN_LIFETIME` field for the share recipient. This property specifies for how long the generated token will be valid after which the recipient loses access to the data share and must request a new token. Enter 60 in the first text field. Select Minutes from the drop-down menu.

Click **Next** to progress to the Shares tab of the Create Share Recipient dialog box.

3. On the Shares tab, select an available share from the drop-down. As you select the share, it lists under the Granted Shares section. You have the choice to share multiple shares to the recipient.

Select **Create** to create the Share recipient.

Once you have created the Share recipient, you will be able to view the newly created Share Recipient entity in the display area of the Provide Share page. The new recipient card displays details such as the entity type and the owner.

Click the name of the entity or click **Action** to view details about the share recipient entity. See [View Share Recipient Entity Details](#).

View Share Recipient Entity Details

Click the **Actions** icon at the right of the Share Recipient entity entry, then click **View Details**.

For all entities, the details include general details, Log details, Granted Shares, Lineage and Impact.

Overview

The General tab displays the Name and Email address of the Share Recipient, the date when the Share Recipient was created, type of the share recipient and granted shares.

Granted Shares

The Granted Shares tab displays the list of shares granted to the recipient.

Log Details

The Log details tab displays the Time, Status, Log Level, Share Name, Request, End Point, Remote address, User Agent and Details of the Recipient. The details depend on the log level settings you select in the **Edit Share Recipient** dialog.

Lineage

Lineage displays all known information about the dependencies of the entity, and therefore how the entity was created and how it is linked to other entities.

For example, for a share you publish, the lineage includes the files in cloud storage along with the cloud location of the files you share.

To view more details about an item, click the **Actions** icon for the item, then click **Expand**. For a cloud location, the files in the location are displayed. Pointing to the name of the file displays the name, application, type, path, and schema of the file. To collapse the display, click the **Actions** icon, then click **Collapse**.

Impact

Impact shows all known information about the downstream use of an entity, and therefore how a change in the definition of an entity may affect other entities that depend on it.

Click **Actions** in the Share Recipient entity card to open the context menu. The actions available are:

- **Edit:** Opens Edit Share Recipient dialog where you can edit the share recipient to update details of the share recipient. You can modify the following field values:
 - **Name:** You cannot modify the name of the Share Recipient.
 - **Description:** Enter the description to add or update description of the Share Recipient.
 - **Log level:** You can select any of the available options such as Basic, Detail and Errors Only based on the details you want to view in the **Log Details** tab.
 - * Basic: This option logs basic information each time the recipient calls an endpoint.
 - * Errors Only: This option logs only errors.
 - * Detail: This option logs detailed information each time the recipient calls an endpoint.
 - **User token lifetime:** You can select the validity period of bearer tokens utilized by the recipient to authenticate access to metadata endpoints. The maximum user token lifetime you can select is 200 minutes, 6 hours and 90 days.
 - **Par lifetime:** You can select the duration for which the pre-authenticated request URLs remain valid for the recipient to access data files.
Click **Update** to modify the values you updated in the wizard.
- **View Details:** See [View Share Recipient Entity Details](#).
- **Granted Share:** Opens Manage Granted Shares dialog where you can grant access to a share from list of available shares or you can revoke access from the current selection of share.

- **Get Bearer Token:** Opens an Auth Token dialog to authenticate the recipient. Use this option when your token expires. This enables you to generate a new token. A Bearer Token is part of the JSON file.
- **Send Activation Mail:** Opens the mail with the profile link.
- **Copy Profile Activation Link to Clipboard:** Copies the JSON profile activation link to clipboard.
- **Rename:** Select this option to rename the Recipient.
- **Delete:** Select this option to delete the recipient you created entirely. By doing so, access to all shares will end instantly and any existing delta credentials will become invalid.

Consume Share

Once the providers share the objects, there are a few steps the recipients need to follow to consume the share.

Use the Consume Share page to perform the following operations:

- [Consume Share Overview](#):
To consume data shares, you need to subscribe to them and create views of tables included in the share.
- [Consume Versioned Share](#):
As a recipient, you will need to download your share profile, subscribe to the data share provider, register the shares and create external tables on top of your shares. The Data Share tool authorizes the access using the JSON profile sent to the recipient using the activation mail. After the access is granted, the Data Share tool links the shared objects with the Data link tool where the consumer can run the data link job and access the objects shared by the provider.
- [Consume Live Share](#):
This lets you as a recipient to consume live data from the database.
- [View Share Provider Entity Details](#):
Use the **Actions** icon at the right of the live share or delta share provider entity entry to view details about the live share or delta share provider entity you create.
- [Consume Share Overview](#)
The Consume Share page provides an overview of the list of share providers, with options to search for and add share providers.
- [Consume Versioned Share](#)
You must follow these steps to make shared versioned data available to you within Oracle Autonomous Database. Data shared with you through Delta Sharing is not automatically available and discoverable in your Autonomous Database.
- [Consume Live Share](#)
Live data shared with you through data sharing is not automatically available for consumption.
- [View Share Provider Entity Details](#)
To view details about the Share Provider entity, click the **Actions** icon at the right of the Share Provider entity entry, then click **View Details**.

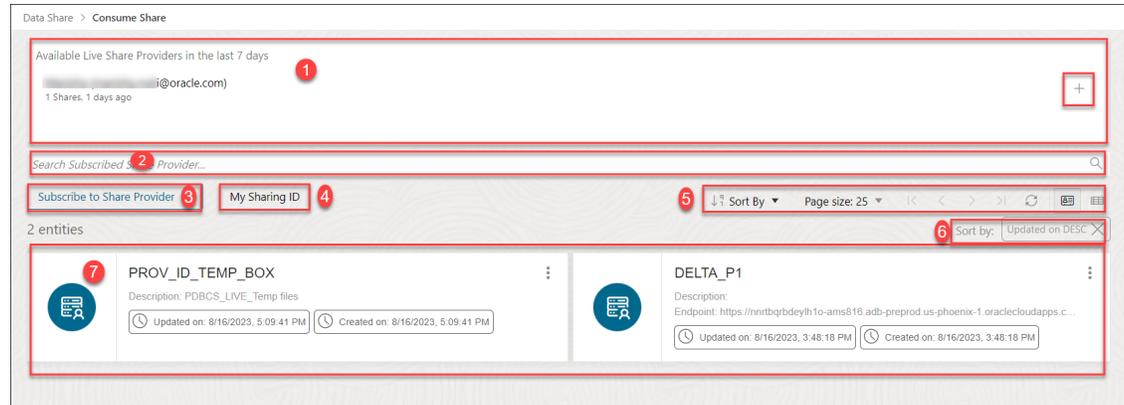
Consume Share Overview

The Consume Share page provides an overview of the list of share providers, with options to search for and add share providers.

To navigate to the Consume Share page, do either of the following:

- On the Data Studio menu, select **Consume Share** under the Data Share menu.
- On the Data Share page, click the **Consume Share** widget present in the Provider and Consumer section.

The Consume Share page contains:



1. Available Live Share Providers in the last 7 Days

This area displays the list of available Live Share Providers you create in the last 7 days. You can update any of the fields as per your wish. You can also subscribe to Live Share using the + sign to the right of Shares listed in **Available shares in the last 7 days**.

2. Search Subscribed Share Provider field

You can search for the Share Recipient you create by entering the name of the Subscribed Share Provider. Enter the name of the Subscribed Share Provider, for example, *REVIEW_PROVIDER* and click the **magnifier** icon to complete the search. The Share tool displays the search results in the display area.

3. Select **+ Subscribe to Share Provider** to subscribe to a new Share Provider. See [Subscribe to Share Provider](#) for exploring this icon.

4. My Sharing ID

A Sharing ID is a unique provider for your Autonomous Database. Copy this ID to the clipboard and paste it in the Sharing ID field of the **Create Share Recipient** wizard. This enables a Live Share to be shared with a share provider.

5. Toolbar

The toolbar consists of the sort by, page size, refresh and entity view options.

6. Sort by settings

When you set sorting values by using the Sort By control in the toolbar, the settings are displayed in small boxes beneath the toolbar. You can delete a setting by clicking the **X** icon in the box. Or you can change the settings by returning to the Sort By control in the toolbar.

7. Display area

The area beneath the Search Consumer Share Providers field displays the entities returned by a search and that match the filter criteria set in the Filters panel. You can sort the entities by clicking the **Sort By** button and then setting sort values.

Consume Versioned Share

You must follow these steps to make shared versioned data available to you within Oracle Autonomous Database. Data shared with you through Delta Sharing is not automatically available and discoverable in your Autonomous Database.

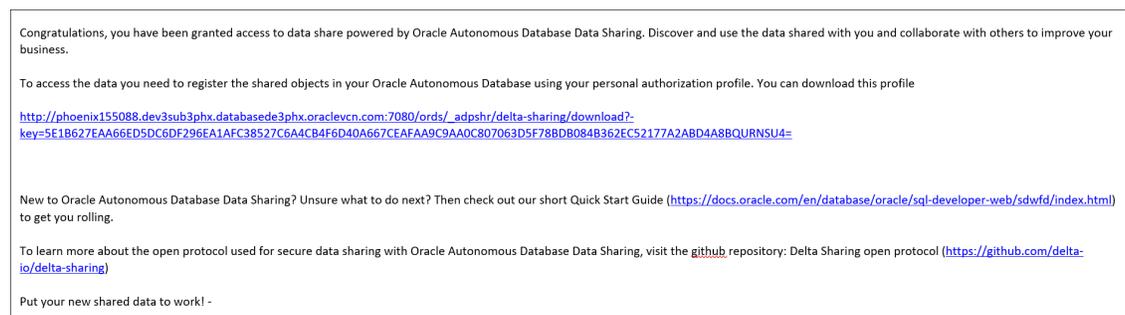
You need to perform several basic steps to subscribe and access the provided data shares. Here are the steps you need to perform:

- Download the JSON profile.
- Subscribe to the data share provider.
- Register shares made available to you.
- Create external tables on top of your shares.

About JSON Profile

Profile files are JSON files containing a user's credentials to access a Delta Sharing Server. This enables you to authenticate yourself with the delta sharing server and to discover the data shares you are eligible to access. Download the profile using the URL included in the invitation email sent by the data share provider.

Once the Share tool creates a share object, the recipient receives an activation mail with subject titled *Oracle Autonomous Database Data Share*.



To access the share, you need to register the shared objects using personal authorization JSON profile.

You can click on the profile link to download the JSON profile. Clicking the profile link takes you to a new screen in the browser with a **Get Profile Information** button as shown below:



Select **Get Profile Information** to download the JSON profile to connect to the share provider.

Note

You can click on the **Get Profile Information** button only once and also view the link to the [Quick Start Guide](#). The share tool does not allow you to select **Get Profile Information** twice. Clicking on it twice brings up a screen which displays the list of causes of the failure to download the profile.

The below profile is an example of the JSON profile you download.

```
{ "
  shareCredentialsVersion ": 1,
  "endpoint": "https://myhost.us.example.com/ords/prov/
_delta_sharing/",
  "tokenEndpoint": "http://myhost.us.example.com:1234/ords/pdbdba/oauth/
token",
  "bearerToken": "-xxxxxxxxxxxxxxxxxxxxxxxx",
  "expirationTime": "2023-01-13T07:53:11.073Z",
  "clientId": "xxxxxxxxxxxxxxxxxxxxxxxx..",
  "clientSecret": "xxxxxxxxxxxxxxxxxxxxxxxx.."} }
```

The profile stores the credentials in an encrypted format. The parameters with their description are:

- **shareCredentialsVersion:** The version of the share you publish.
- **endpoint:** Specifies the share endpoint.
- **tokenEndpoint:** Specifies the token endpoint. The Share tool client use the token endpoint to refresh the timeout on your bearer token if you are consuming the share using Oracle.
- **bearerToken:** This is a cryptic string which the authentication server generates in response to a login request.
- **expirationTime:** This is the time taken for the authentication to expire.
- **ClientID:** Specifies the public identifier the authentication server generates when you register the instance for authentication.
- **clientSecret:** Specifies a secret identifier the authentication server generates for authorization.

Copy the JSON content of the profile in a notepad. You will need this JSON below to subscribe your share provider.

Note

Ensure you copy the full JSON content profile, including the left brace and the right brace.

Security enhancements

As a share recipient, you must set up an access control list (ACL) to the share provider's machine by using the `APPEND_HOST_ACE` procedure as an ADMIN user, or another privileged user. This allows you to access the share via the Internet.

Note

This must be done before using the Add Share Provider Wizard to add an access control entry (ACE) to the access control list (ACL) of the host (i.e. Share provider). You can find the host name from the JSON profile you downloaded in the previous step.

For example, if you wish to allow a database user, `A_SHARE_USER` to access the endpoints on a host (Share provider) named, here is a sample of PL/SQL procedure you will need to run in the SQL worksheet editor as an admin. As a prerequisite, extract the host name from the endpoint property in the delta sharing JSON profile, as provided in the example above. The hostname from the example is `myhost.us.example.com`.

```
BEGIN
    dbms_network_acl_admin.append_host_ace(
        host => 'myhost.us.example.com',
        lower_port=>443,
        upper_port=>443,
        ace => xs$ace_type(
            privilege_list => xs$name_list('http', 'http_proxy'),
            principal_name => 'A_SHARE_USER',
            principal_type => xs_acl.p_type_db));
    COMMIT;
END;
```

/

Following are the parameters with their description:

- **host**- Specifies the name or the IP address of the host. The host or domain name is not case-sensitive.
- **lower port**- Specifies the lower port of an optional TCP port range.
- **upper port**- Specifies the upper port of an optional TCP port range.
- **ace** – The Access Control Entry.
- **privilege list**- Specifies the list of network privileges to be granted or denied.
- **principal_name**- It is the principal (database user or role) to whom the privilege is granted or denied. It is case-sensitive.
- **principal_type**- Specifies the type of principal you use.

Refer to the [PL/SQL Packages and Types Reference](#) document for more details on the `DBMS_NETWORK_ACL_ADMIN` package subprograms.

Grant an ACL to the user on the local ORDS endpoint. You will need this to generate bearer tokens on locally created shares.

```
PRIV_ORDS_ACL          CONSTANT PLS_INTEGER := 8;
```

In this process you will load the provider's JSON profile for configuration and credentials to enable access to the recipients.

1. Open the Consume Share page and click **+ Subscribe to Share Provider** to select **Subscribe to Delta Share Provider** from the drop-down. This opens the Subscribe to Share Provider dialog box.

- On the Provider Settings pane of the Register Share Provider dialog box, specify the following details:

The screenshot shows the 'Subscribe to Share Provider' dialog box with the 'Provider Settings' tab selected. The 'Share Source' section has 'Delta Share Provider JSON' selected. The 'Share Provider JSON' section has 'From File' selected, and a file named 'delta_share_profile (10).json' is selected for loading. The 'Provider Name' field contains 'Provider' and the 'Description' field is empty. The 'Next' button is highlighted.

- Provider Name:** *Provider*.
- Description:** Add a description. This field is optional.

Under Share Source section, choose **Delta Share Provider JSON**.

Under Share Provider JSON field, you can share the JSON profile in the following ways:

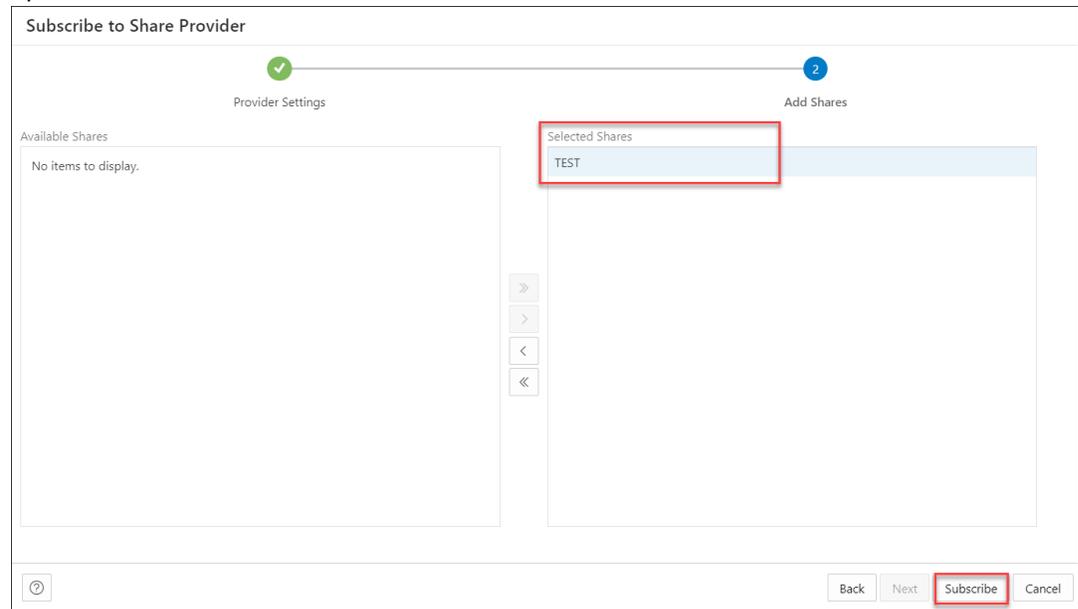
- From File:** Select this option and click on the drop area titled "Delta Share Profile JSON". Clicking on the area opens your local repository where you can select the JSON profile you had downloaded.
- JSON:** You can select this option and paste the JSON content of the profile you copy into the notepad.

Upload the JSON profile file and create a share provider subscription.

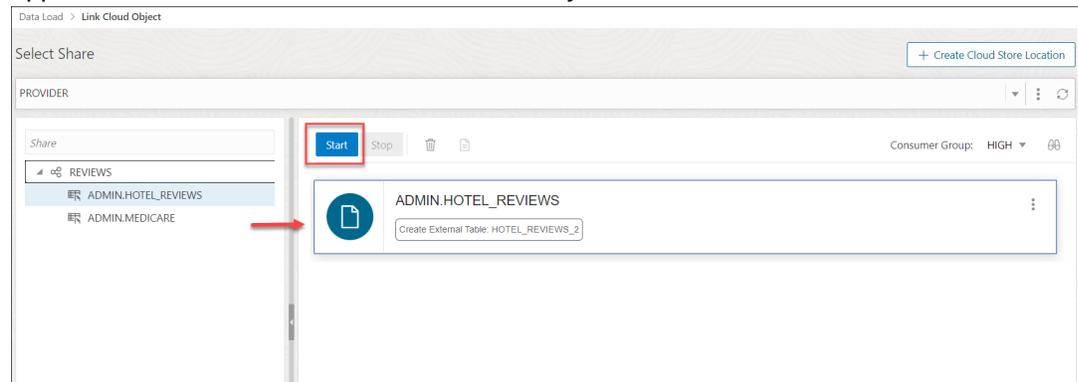
Click **Next** to progress to the Add Shares tab.

- On the Add Shares tab of the dialog, you will view the list of available shares. Click the share you wish to consume from the Available Shares and select any of the available

options:



- >: This option enables you to move the Available Share to Selected Shares.
 - <: Select this option to remove the selected share from Selected Shares.
 - >>: This option allows you to move all the shares to the Selected Shares.
 - <<: Select this option to remove all the selected shares from Selected Shares.
4. Click **Subscribe** to add the share. A confirmation prompt appears when the provider is created successfully. After successful creation of the provider, you will now view the Link Cloud Object screen of the Data Load page.
 5. You can view the name of the provider in the cloud storage location field. The share appears in the source file location with the files you add to the share.

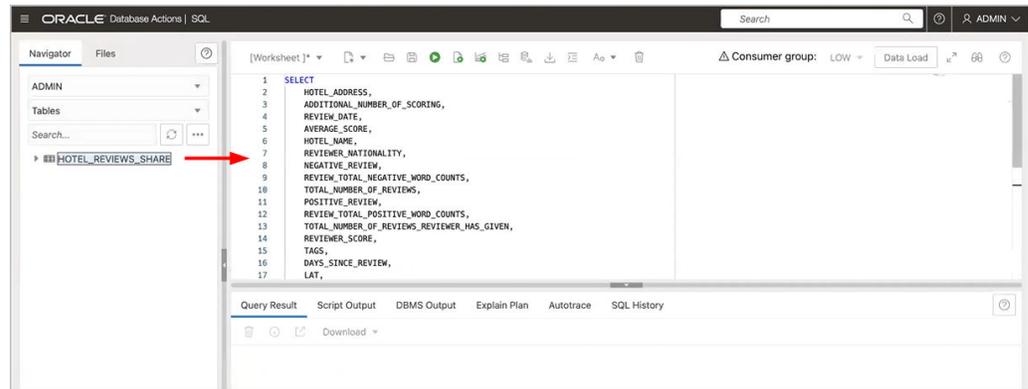


Expand the Share folder cart, drag and drop the file you share from the source to the Data Link cart.

Select **Start** in the Data link cart to run the data link job.

6. View the created tables from Database Actions.
 - Click on Database Actions, in the breadcrumb, to go back to the Database Actions launchpad.
 - Click on the SQL tile.
 - Select the external table, drag and drop it into the worksheet.

The SQL Select statement for the table appears. This SQL statement can be run to consume the shared data.



Consume Live Share

Live data shared with you through data sharing is not automatically available for consumption.

To consume live data shares, you need to subscribe to them and create views of tables included in the live share. The views can be queried using SQL scripts.

1. Open the Consume Share page and click **+ Subscribe to Share Provider** to select **Subscribe to Live Share Provider** from the drop-down. This opens the Subscribe to Share Provider dialog box.
2. On the Provider Settings pane of the Subscribe to Share Provider dialog box, specify the following details:

Subscribe to Share Provider

1
 Provider Settings

2
 Add Shares

Share Source *

Select from Live Share Providers
 Delta Share Provider JSON

Share Provider Details

Provider Name *

Provider

Description

?
Back
Next
Subscribe
Cancel

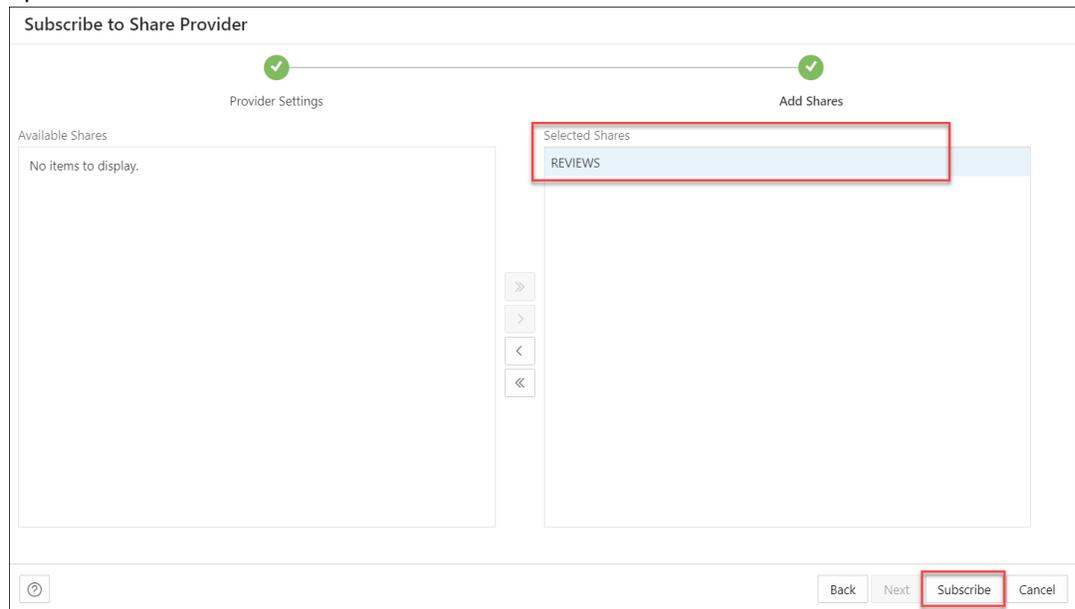
Under Share Source section, choose **Select from Live Share Providers** and select the provider from the drop-down.

Under Share Provider Details field, enter the following:

- **Provider Name:** Specify the name of the provider.
- **Description:** Enter a description of the Provider.

Click **Next** to progress to the Add Shares tab.

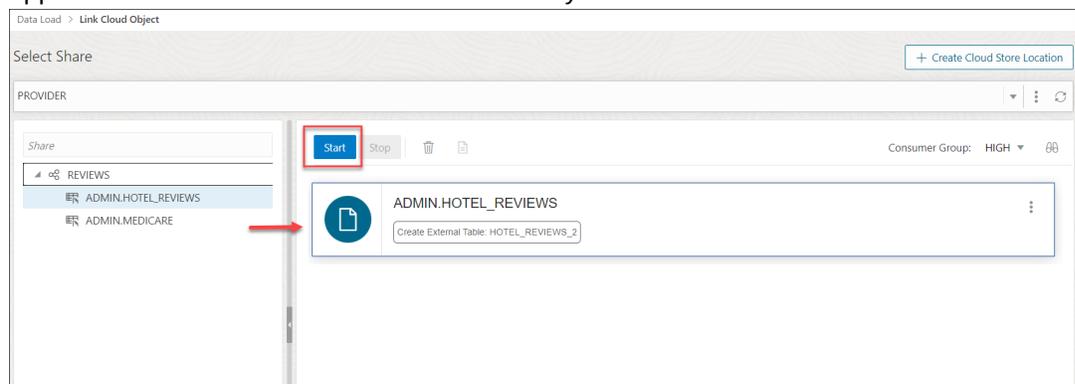
- On the Add Shares tab of the dialog, you will view the list of available shares. Click the share you wish to consume from the Available Shares and select any of the available options:



- >>: This option enables you to move the Available Share to Selected Shares.
- <: Select this option to remove the selected share from Selected Shares.
- >>>: This option allows you to move all the shares to the Selected Shares.
- <<<: Select this option to remove all the selected shares from Selected Shares.

Click **Subscribe** to add the share. A confirmation prompt appears when the provider is created successfully. After successful creation of the provider, you will now view the Link Cloud Object screen of the Data Load page.

- You can view the name of the Share provider in the cloud storage location field. The share appears in the source file location with the files you add to the share.



Expand the Share folder cart, drag and drop the file you share from the source to the Data Link cart.

Select **Start** in the Data link cart to run the data link job.

View Share Provider Entity Details

To view details about the Share Provider entity, click the **Actions** icon at the right of the Share Provider entity entry, then click **View Details**.

For all entities, the details include Lineage and Impact sections.

For a specific Share Provider entity, you can perform the following actions using the **Actions** context menu.

- **View Details:** See [View Share Provider Entity Details](#).
- **Manage Shares:** Opens a Manage Shares for Share provider dialog box. This lists the shares you select to share with the recipient. You can edit the list of shares you wish to share with the recipient. Click **OK** to save any changes or select **Cancel** to discard the process of editing.
- **Load to table:** You are directed to the Load Data page with the Share tab selected.
- **Link to table:** You are directed to the Link Data page with the Share tab selected to view and run the related link object. Drag and drop the shared data to add it to the data link job.
- **Rename:** Renames the Provider Name. Select **Yes** to make changes else click **No**.
- **Copy Endpoint:** Copies the endpoint.
- **Delete:** Removes the Share Provider Entity.

Manage Shares with DBMS_SHARE

In addition to the Data Share tool, Autonomous Database supports a fully scriptable workflow via the PL/SQL `DBMS_SHARE` package.

Using `DBMS_SHARE`, providers can create and publish shares, add or remove tables and views, register recipients, grant or revoke access, and manage share lifecycle; consumers can discover available shares and programmatically connect to them. The package also exposes producer and consumer views for monitoring and troubleshooting.

If you prefer a guided example, see [Implement Data Sharing with PL/SQL](#).

This LiveLab tutorial walks through enabling a schema for sharing, creating and populating a share, authorizing a recipient, publishing the share, and consuming it - all with `DBMS_SHARE` calls.

Data Sharing with Python Data Studio APIs for Oracle Autonomous Database

You can also share data using Python Data Studio API for Oracle Autonomous Database. See [Data Studio Data Share API functions](#) for more information.

Data Share Limitations on Autonomous Database

This section summarizes the limitations of Data Share (both Versioned and Live Share) included in the Oracle Autonomous Database.

Oracle Database Release Restriction

Only **Oracle Database 19c** supports both Versioned Share and Live Share. You cannot use Versioned Share and Live Share with any other database version.

Supported Column Types

Only the following list of column types are supported:

- `BINARY_DOUBLE`
- `BINARY_FLOAT`

- CHAR
- CLOB
- DATE
- FLOAT
- INTERVAL DAY TO SECOND
- INTERVAL YEAR TO MONTH
- NCHAR
- NCLOB
- NUMBER
- NVARCHAR2
- TIMESTAMP
- VARCHAR2

Live Share Data Provider

The following are limitations for using the Data Share tool as a Data Share provider:

1. A live share data provider can create shares with a maximum of four objects.
2. If the live share data provider is sharing multiple tables which would normally be joined together, it is recommended that the producer create a view that performs the joins and then only shares the view.

Live Share Feature Slow Performance

With an increase in ECPUs, does the performance of the Live share feature in the Data Share tool improve while running complex queries?

No, it does not. In such cases, you need to configure the **Degree Of Parallelism (DOP)** using Data Studio to improve the performance of complex running queries.

Configure the Degree Of Parallelism (DOP) using Data Studio

When the Live Share feature of the Database Actions instance runs on an increased value of ECPU, by default the performance of the complex running queries does not improve.

You can use the DOP option to improve the performance of the complex running queries. The DOP sets the number of processors employed to run a single statement, for each parallel plan execution. This section describes how to configure the `degree of parallelism (DOP)` configuration option in the Autonomous Database by using the Live Share feature of Data Studio.

In this example, you will use the Live share feature and share tables of an uncompressed size of 1TB.

Before you begin, here are a few things you need to consider:

- The option to configure the value of DOP can only be done by an `admin`.

Prerequisite:

- Log in to your Database Actions instance as an `admin` and under **Administration**, select **Set Resource Management Rules**.
- Note the default **concurrency value** and **DOP**.

To configure the degree of parallelism using Data Studio to improve the performance of complex running queries, you will:

- Create Share as a provider.
- Consume Share as a consumer and create external queries on top of your shares to record the query execution time.
- Increase the ECPUs to a value greater than the current value as provider and consumer and record the query execution time.
- Configure the DOP value for the MEDIUM consumer group.
- Consume the same share and record the query execution time of the same table.

Note

You must configure the ECPU count and DOP value as a **provider** with ADMIN privileges.

On the Database Actions launchpad, click **Selector** and select the **Data Share** tool from the **Data Studio** suite of tools.

1. Create Share as a provider.

- Click **Provide Shares** and select **+Create Share**.
- On the Create Share wizard, in the **Name** field of the General tab, enter a name for the Share. For example: *TEST*.

The screenshot shows the 'Create Share' wizard interface. At the top, there is a progress bar with four numbered steps: 1. General, 2. Publish Details, 3. Select Tables, and 4. Recipients. The 'General' tab is currently selected. Below the progress bar, there are two input fields: 'Name' (containing 'TEST') and 'Description' (empty). At the bottom right of the wizard, there are four buttons: 'Back', 'Next', 'Create', and 'Cancel'. A help icon (?) is located at the bottom left.

In the **Description** field, enter a description for the data you share. This field is optional.

Select **Next** to progress to the Publish Details tab.

- In the Publish Details Tables tab of the wizard, select **SHARE LIVE DATA USING DIRECT CONNECTION**.

Create Share

1 General 2 Publish Details 3 Select Tables 4 Recipients

SHARE VERSIONS USING OBJECT STORAGE
Share data as a set of distinct versions, such as end of each day. The recipient will only see changes to the data when you publish a new... version.

SHARE LIVE DATA USING DIRECT CONNECTION
Share data as of the latest database commit with Autonomous Databases in the same region. The recipient will always see the latest data.

Back Next Create Cancel

Click **Next**.

- d. On the **Select Tables** tab of the wizard, select the schema from the drop-down menu, drag and drop the `CUSTOMER` table from the Available Tables.

Create Share

1 General 2 Publish Details 3 Select Tables 4 Recipients

Available Tables

Search

Name Role

CUSTOMER

DBTOOLS\$EXECUTION_HIS...

HRRECORDS50K_TABLE1

I_INCIITEM

Shared Tables

TPUSER1

CUSTOMER
from TPUSER1.CUSTOMER

Drag and Drop

Back Next Create Cancel

Note

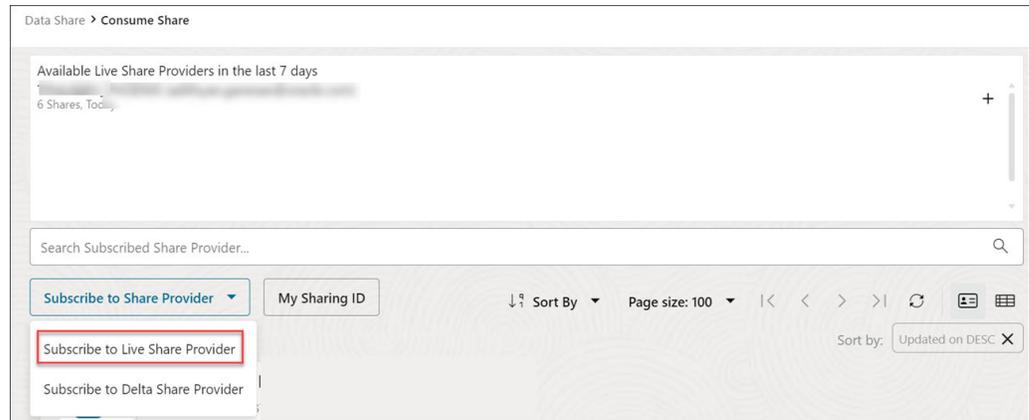
In this example, we are sharing a single table named `Customer` of large size.

Click **Next** to proceed to the Recipients tab of the Create Share Wizard.

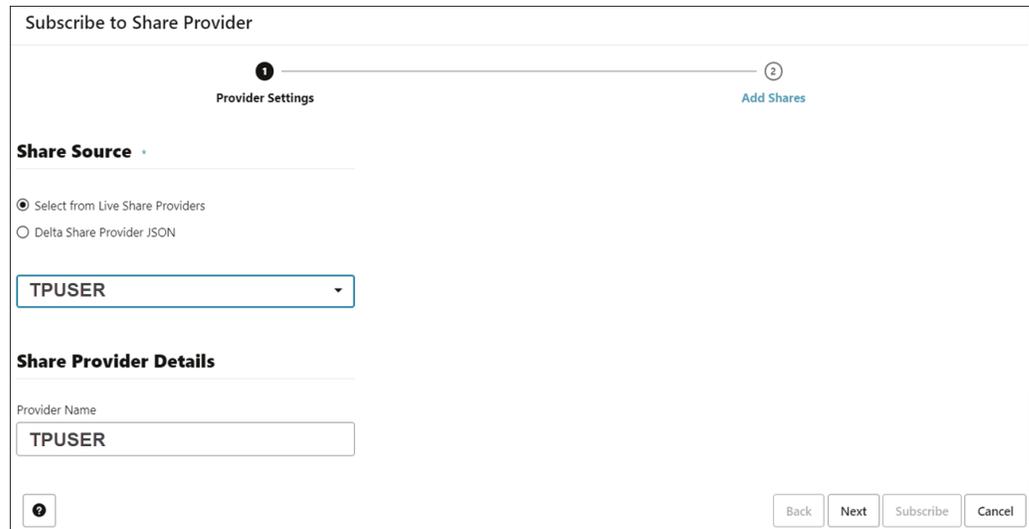
- e. On the Recipients tab of the Create Share wizard, select `MY_COMPARTMENT` from the drop-down.

- f. Select **Create** to create the share.

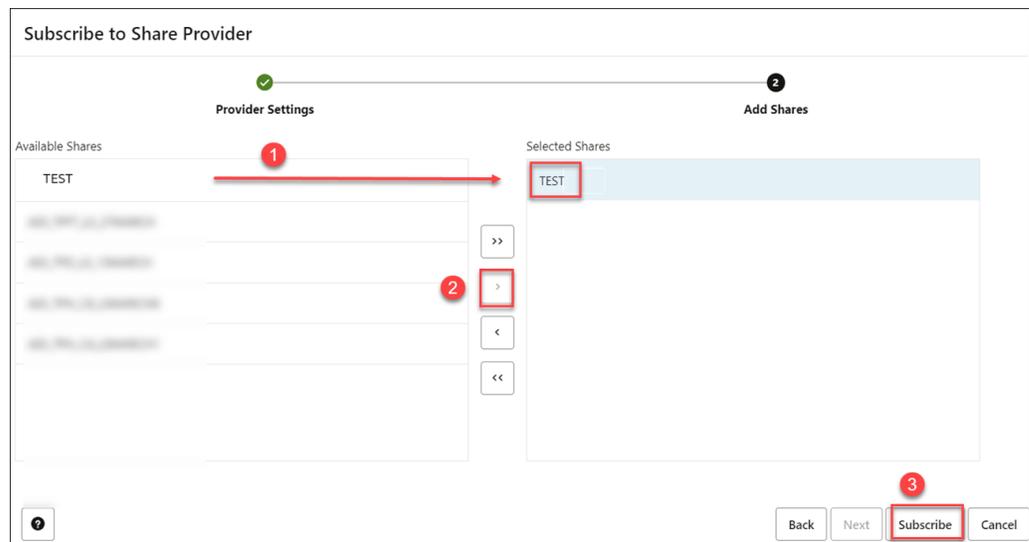
2. **Consume Share as a consumer and create external queries on top of your shares to record the query execution time.**
 - a. After you have created the share as a provider, you will consume the data shared by the provider.
See [Consume Share](#) for more details.
 - b. Log in as a **consumer** to subscribe to the data share provider and access the data in your share.
 - c. On the **Provider and Consumer** page of the Data Share tool, click the **CONSUME SHARE** tile to display the **Consume Share** page.
 - d. Click the **Subscribe to Share Provider** drop-down list, and then click the **Subscribe to Live Share Provider** option.



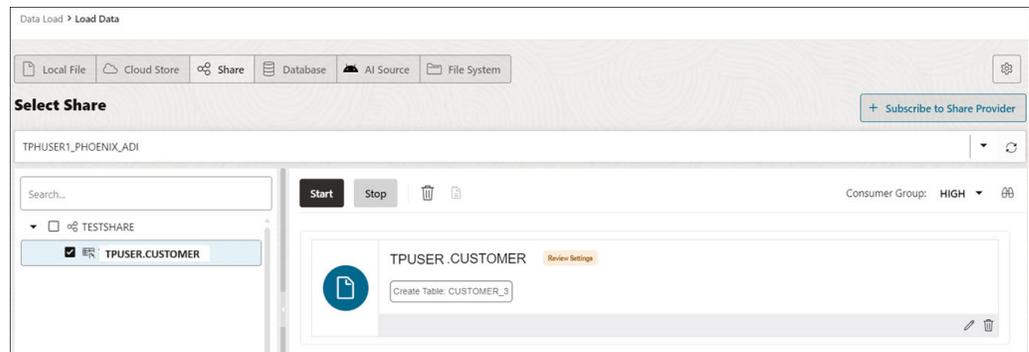
- e. Under Share Provider Details, select the **Provider Name**. In this example, TPUSER is the name of the provider.



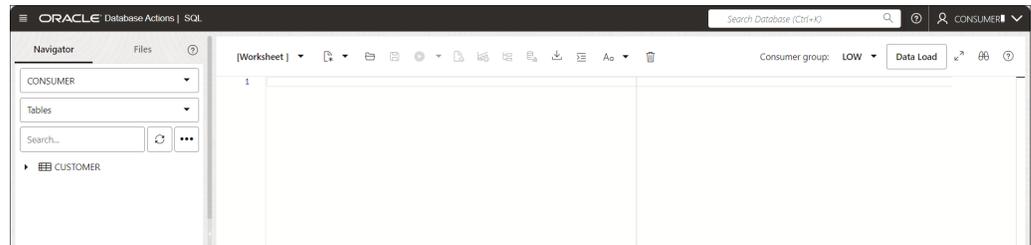
- f. On the **Add Shares** panel of the Subscribe to Share **Provider Settings** wizard page, click the **TEST** in the **Available Shares** section, and then click the (>) icon.



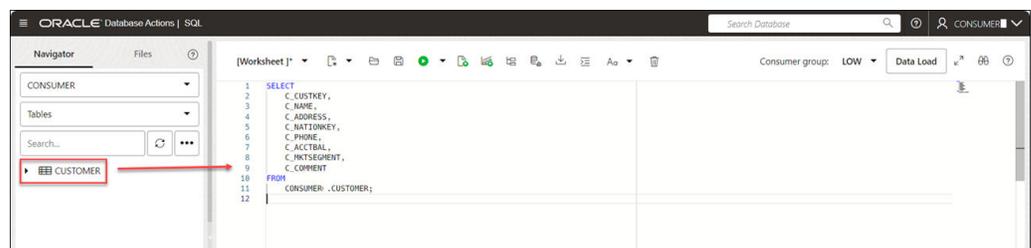
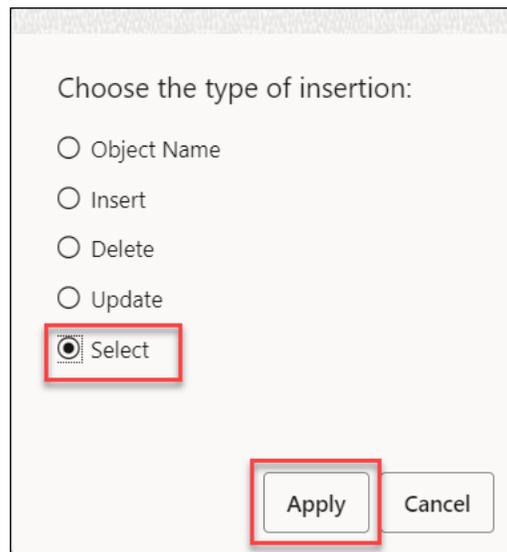
- g. The tool adds the share to the **Selected Shares** section. Click **Subscribe**.
- h. A TPUSER share provider was created successfully informational message is displayed. You are directed to the **Load Data** page with the **Share** tab selected.



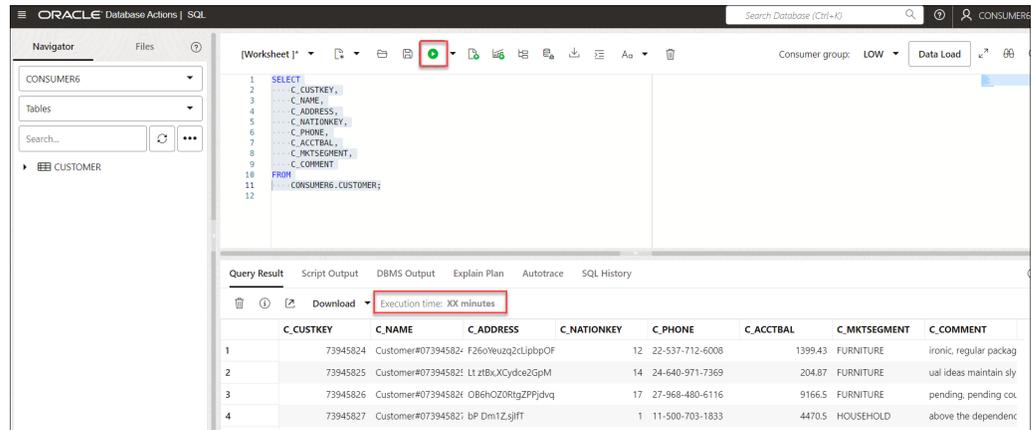
- i. Expand the TESTSHARE data share node to display the objects in it. Drag and drop the TPUSER.CUSTOMER shared table to add it to the data load job.
- j. The shared table is added to the data link job. When you run this job, a new CUSTOMER view will be created on top of the CUSTOMER table that was shared with you.
- k. Click **Start**. A **Start Link from Cloud Store** dialog box is displayed. Click **Run**. When the job completes successfully, a link icon is displayed. The CUSTOMER view is created.
- l. Click the **Database Actions | Data Share** banner. On the **Database Actions | Launchpad**, click the **Development** tab, and then click the **SQL** tab to display the SQL Worksheet.



- m. Drag and drop the `CUSTOMER` table to the SQL worksheet area.
- n. A dialog displays offering the types of available insertions.
- o. Click **Select**, then **Apply**.



- p. Click **Run**. You can view the automatically generated query displayed in the Worksheet.



The screenshot shows the Oracle Database Actions SQL interface. The query executed is:

```

1 SELECT
2   C.CUSTKEY,
3   C.C_NAME,
4   C.ADDRESS,
5   C.NATIONKEY,
6   C.PHONE,
7   C.ACCTBAL,
8   C.MKTSEGMENT,
9   C.COMMENT
10 FROM
11   CONSUMER.CUSTOMER;
12

```

The Query Result tab shows the following data:

	C.CUSTKEY	C.C_NAME	C.ADDRESS	C.NATIONKEY	C.PHONE	C.ACCTBAL	C.MKTSEGMENT	C.COMMENT
1	73945824	Customer#073945824	F26oYeuqz2ClpIpOP	12	22-537-712-6008	1399.43	FURNITURE	ironic, regular packag
2	73945825	Customer#073945825	Lt.ztBkXCydce2GpM	14	24-640-971-7369	204.87	FURNITURE	ual ideas maintain sty
3	73945826	Customer#073945826	OB6hOZ0RtgZPPjdq	17	27-968-480-6116	9166.5	FURNITURE	pending, pending co.
4	73945827	Customer#073945827	bP.Dm1Z.sjIT	1	11-500-703-1833	4470.5	HOUSEHOLD	above the dependenc

- q. Note the query execution time in the **Execution time** field displayed under Query Result tab.
3. **Increase the ECPUs as provider and consumer and record the query execution time.**
 - a. Increase the ECPUs of your Autonomous Database instance as a **provider** and a **consumer** from your OCI console.

Note

Make sure you have not enabled Auto Scaling.

In this example we have increased the ECPU count to **32**. You can upscale the ECPU count to any value of your choice.

- b. Repeat consuming the same `CUSTOMER` share and calculating the query execution time. There will be no change in the time to query the table.
4. **Configure the DOP value for the `MEDIUM` consumer group.**
 - a. Log in to your database actions instance as a **consumer** with `ADMIN` privilege.
 - b. Click **Selector** and under **Administration**, select **Set Resource Management Rules**.
 - c. Decrease the **Concurrency limit** of the `MEDIUM` consumer group to **8** and click **Save Changes**. Decreasing the concurrency limit increases the DOP.

Set Resource Management Rules

Run-away criteria	CPU/IO shares	Concurrency limit
		Degree of parallelism (DOP)
HIGH	disabled 3	disabled 16
MEDIUM	<input style="width: 50px; border: 1px solid red;" type="text" value="8"/>	disabled 4
LOW	disabled 2400	disabled 1

Your ECPU count is **32** and Auto Scaling is **disabled** for your instance.

i **Note**

The concurrency limit you set in this step should be lesser than the default value you noted in the prerequisite.

5. **Follow up step- After you configure the degree of parallelism option, repeat the process of consuming the share, creating external queries on top of your shares to record the query execution time.**

You will see the value of query execution time has decreased.

The improvement in the performance of the same complex SQL queries happens immediately without restarting the server.

FAQs on the Data Share Tool

The following are frequently asked questions on the Data Share tool.

1. When do recipients receive the Data Definition Language (DDL) updates to the source tables?
The recipients need to link objects again from the cloud storage location.
 - Navigate to the link cloud Object page from the Data Studio menu.
 - Select the table in the Data Link cart.
 - Click on the **Actions** icon on the table and select Settings. This opens the Link Data from Cloud Store Location dialog.
 - Click the **Settings** pane and select "Drop Table and Create New External Table" option from the drop-down.

You will receive the updated source table.

2. When do recipients receive the Data Manipulation Language (DML) updates to the source tables?

The provider selects the option to publish versioned data, changes will be reflected if the provider publishes a new version.

3. Does the Data Share tool support object path-based partitioning on the Data Provider end when Oracle Autonomous Database is the consumer?

No, the Data Share tool does not support object path-based partitioning on the Data Provider end. It supports consumption of only the values that are physically present within a file. The Oracle Autonomous Database consumers will not be able to view the partition values that are based on the file path.

Part IV

Administration

This part provides information about the following topics:

Topics:

- [The Database Users Page](#)
- [The APEX Workspaces Page](#)
- [The Data Pump Page](#)
- [The Database Users Page](#)
- [The APEX Workspaces Page](#)
- [The Data Pump Page](#)

21

The Database Users Page

Note

Available only if you signed in to an Oracle Autonomous Database as a user with administrator rights.

The Database Users page enables you to perform user management tasks such as create, edit, enable REST, and delete, and create and manage self service schemas. For user management, the actions available are based on the user privileges (CREATE, ALTER, DROP) granted to you.

Users with no assigned privileges can still access the Database Users page to browse all users. However, the only action available to them is changing their password.

To navigate to the Database Users page, do either of the following:

- In the Launchpad page, select the **Administration** tab and click **Database Users**.
- Click **Selector**  to display the navigation menu. Under Administration, select **Database users**.

The Database Users page consists of two parts: Current User and All Users.

Current User



Displays information about the current user such as user name, whether REST Enabled or not, REST Alias, account expiration (in days), and the last login date and time. The icon on the left displays the user status with one of the following colours: green (Open), blue (Locked), and red (Expired).

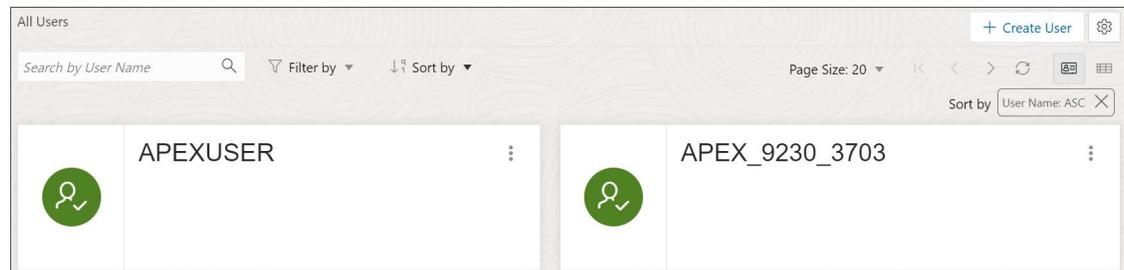
The URL at the bottom is displayed only if the user is REST Enabled. It provides the URL to the Database Actions user login page. Use **Copy to Clipboard**  to copy the URL to the user's clipboard.

Click **Actions**  to open the context menu. The actions available are:

- **Edit**: Opens the Edit User Dialog, where you can edit current user information. See [Creating or Editing a User](#).

- **Enable REST:** Enables REST for a user where disabled. When this option is selected the first time, it opens the Enable REST dialog. See [Enabling REST](#).
- **Disable REST:** Disables REST where enabled for a user.
- **Drop REST Services:** Removes the REST data for a user, such as REST Alias, Base Path and so on, which is stored in Oracle REST Data Services (ORDS).
- **Delete:** Opens the Delete User dialog, where you can delete the user. See [Deleting a User](#).

All Users



Displays information about all other users that have been created in the database. You can use the Search field, which is case insensitive, to search for users, or sort the users in ascending or descending order using the sort icons, or filter by user status or REST status.

To create a user, click **Create User** to open the Create User dialog. For more information, see [Creating or Editing a User](#).

There are two views available:



(Card View): Displays the user information in a card view. This is the default display view. Each user card provides details such as user status, password expiry, user name, and the context menu.



(Grid View): Displays the user information in a tabular format. The last column in each row contains the context menu icon.

- [Creating or Editing a User](#)
You can create a new database user or edit an existing database user.
- [Enabling REST](#)
You can enable REST for a user that has not been REST enabled.
- [Deleting a User](#)
Use this option to delete users.

Creating or Editing a User

You can create a new database user or edit an existing database user.

- To create a new database user, click **Create User** in the Database Users page.
- To edit an existing database user, select **Edit** from the context menu for the associated user.

The user properties are grouped under two tabs: User and Granted Roles.

User Tab

Specifies general properties for the database user.

- **User Name:** The user name string. For an existing user, this field is read-only. To change the name, you must drop the user and create a new user with the desired name.
- **Password:** Password string for the new user, or new password for an existing user. You must also type the same password string for **Confirm Password**.
- **Password Expired:** If this option is selected, the password is marked as expired, and the user must change the password before being permitted to connect to the database.
- **Account is Locked:** If this option is selected, the user will not be permitted to connect to the database until a DBA user unlocks the account associated with this user.
- **OML:** If this option is selected, Oracle Machine Learning is enabled for the user. This user is granted the following three roles: OML_DEVELOPER, DWROLE, PYQADMIN. The user card displays "OML Enabled".

Note

For the OML option to be available in the Add Create User slider, create an OML user named "OML\$PROXY" and grant the following roles in the SQL worksheet: OML_DEVELOPER, DWROLE, PYQADMIN.

Note

To remove the OML roles granted, disable **Enable OML**. You cannot remove a single role alone, all three roles need to be removed to disable the OML option.

- **Quota on tablespace USERS:** Enter or select to assign quota on the user's default tablespace. By default, the user has no quota on the default tablespace.
- **Web Access:** If this option is selected, the user is enabled for REST access. Expand **Web access advanced features** to specify the related fields: **Authorization required**, **REST Alias**, and **URL Mapping Type**.

Granted Roles Tab

Specifies roles to be granted to the user.

Use **Filter by role** to quickly locate the required roles.

For each role, you can select **Granted** to grant the role, **Admin** to permit the user to grant the role to other users, and **Default** to use the default settings for Granted and Admin. A new user is granted **CONNECT** and **RESOURCE** roles when Web Access is selected.

Show code: Select this option to view the PL/SQL code equivalent of the Create/Edit User slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create/Edit** in the Create/Edit User slider.

Enabling REST

You can enable REST for a user that has not been REST enabled.

In the Database Users page, select **Enable REST** from the user's context menu.

When you select Enable REST for a user for the first time, REST Enable User dialog is displayed. Subsequently, if you disable REST and then select Enable REST again, you receive a message stating that REST is enabled. In this case, the REST data previously provided is used for enabling REST. To enter new REST data, select **Drop REST Services** and then select **Enable REST** again.

Schema Alias: Enter the alias for the schema name to use in the URL.

URL Mapping Type: Select **BASE_PATH** or **BASE_URL**.

Authorization Required: For a schema, controls whether Oracle REST Data Services should require user authorization before allowing access to the Oracle REST Data Services metadata catalog of this schema.

Deleting a User

Use this option to delete users.

In the Database Users page, select **Delete** from the user's context menu to delete a user.

Note

The number of user's active sessions is displayed in the Delete User dialog window.

- **Cascade:** If this option is selected, all dependent objects are also deleted.
- **Drop REST Services:** If this option is selected, all user REST data is removed from ORDS.

Note

If you do not select this option when deleting a user, the next time you create a user with the same user name, it will still retain the REST-enabled property.

Click **Delete User** and a confirmation notification is displayed.

Note

An error notification is displayed, if the user has active sessions. In such cases, you must close the active sessions before you can delete the user.

22

The APEX Workspaces Page

Note

You can create and delete workspaces only if you have the **APEX_Administrator_Role** assigned.

You can create APEX workspaces using the APEX Workspaces page. For more information about APEX workspaces, see [Workspace and Application Administration](#) in the *Oracle Application Express Administration Guide*.

To navigate to this page, do either of the following:

- In the Launchpad page, select the **Administration** tab and click **APEX Workspaces**.
- Click **Selector**  to display the navigation menu. Under **Administration**, select **APEX Workspaces**.

For a specific workspace, you can perform the following actions using the  **Actions** context menu:

- **Create a Workspace:** See [Create a Workspace](#)
- **Delete a Workspace:** Removes the workspace and all associated applications from the database.
- **View Schema Details:** Navigates to the Database Users page to view the related schema information.
- [Create a Workspace](#)

Create a Workspace

Note

You require Administrator privileges to create a workspace.

To create an APEX workspace:

1. In the APEX Workspaces page, at the top right, click **Create Workspace**.
2. Enter the following fields:
 - **Workspace name:** Enter a name for the workspace.
 - **Database user:** Select a schema from the drop-down list, or type to enter a new schema.
 - **Workspace password:** Enter a password for the workspace.

- **APEX Administrator:** Enter the administrator user details.

Show code: Select this option to view the PL/SQL code equivalent of the Create Workspace slider. You can copy and execute this PL/SQL code in the worksheet to perform the same action that occurs when you click **Create** in the Create Workspace slider.

3. Click **Create.**

The new workspace is displayed in the APEX workspaces page. The workspace card includes details such as the number of applications in the workspace, the number of developers using the workspace, and the number of administrators responsible for the workspace. Depending on your role, the  **Actions** context menu provides options such as creating and deleting a workspace, and viewing schema details.

23

The Data Pump Page

Note

This feature is only available for Oracle Database 12.2 and later releases.

The Data Pump page enables you to monitor Data Pump jobs that were initiated through the available Database API endpoints, the `DBMS_DATAPUMP` package, or the SQL Developer Data Pump Export and Import wizards.

The section at the top displays the total number of executing jobs, stopped jobs, and completed jobs. Click a tile (example, STOPPED) to filter and view the corresponding list of STOPPED jobs in the default card format.

You can filter or sort the jobs and set the time period by which to refresh the data.

A job card displays the following details: Job name, import or export operation, percentage of completion and time elapsed, and links to dump files and logs. The status of the job is indicated by the colour of the icon on the left side of the card. Green indicates successful jobs, yellow indicates that the jobs need to be reviewed, and blue indicates that the jobs are in progress.

In a job card, you can:

- Use  **Download** to access dump files for completed jobs.
- Use  **Log** to access the log files.

- [Importing Data using Oracle Data Pump](#)
You can import data from Data Pump files into your cloud database.

Importing Data using Oracle Data Pump

You can import data from Data Pump files into your cloud database.

Note

Before you begin, you must have an export job. To create an export job using cURL, see [Create an Export Data Pump Job](#).

With Oracle Data Pump Import, you can load an export dump file set into a target database, or load a target database directly from a source database with no intervening files.

For more information about Data Pump import, see [Data Pump Import](#) in *Oracle Database Utilities*.

Topics:

- [Requirements](#)
- [Importing Data](#)
- [Requirements](#)
- [Importing Data](#)

This section provides the steps for importing data using Oracle Data Pump in Database Actions.

Requirements

You need to set up a resource principal or a cloud service credential to access the Oracle Cloud Infrastructure Storage Object.

- [Setting Up a Resource Principal](#)
- [Setting Up a Cloud Service Credential](#)

Setting Up a Resource Principal

1. Log in as the ADMIN user in Database Actions and enable resource principal for the Autonomous Database.

In the SQL worksheet page, enter:

```
EXEC DBMS_CLOUD_ADMIN.ENABLE_RESOURCE_PRINCIPAL( );
```

Optional: The following step is only required if you want to grant access to the resource principal credential to a database user other than the ADMIN user. As the ADMIN user, enable resource principal for a specified database user by using the following statement:

```
EXEC DBMS_CLOUD_ADMIN.ENABLE_RESOURCE_PRINCIPAL(username => 'adb_user');
```

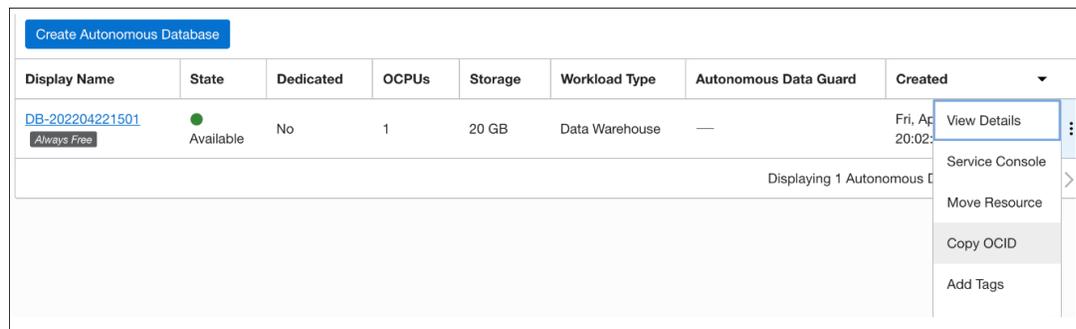
This grants the user `adb_user` access to the credential `OCI$RESOURCE_PRINCIPAL`. If you want the specified user to have privileges to enable resource principal for other users, set the `grant_option` parameter to `TRUE`.

```
BEGIN
DBMS_CLOUD_ADMIN.ENABLE_RESOURCE_PRINCIPAL(
username => 'adb_user',
grant_option => TRUE);
END;
```

For more information, see [Using Resource Principal](#) in *Oracle Cloud Using Oracle Autonomous Database Serverless*.

2. Obtain the resource.id.

In the Oracle Cloud Infrastructure console, select **Oracle Database** and then select **Autonomous Databases**. In the Database tab, click the **Actions** icon (three dots) and select **Copy OCID**. This is applicable for all database instances in all the compartments.



Display Name	State	Dedicated	OCPUs	Storage	Workload Type	Autonomous Data Guard	Created
DB-202204221501 <small>Always Free</small>	Available	No	1	20 GB	Data Warehouse	—	Fri, Apr 20:02:...

3. Create a dynamic group.

- a. In the Oracle Cloud Infrastructure console, click **Identity and Security** and click **Dynamic Groups**.
- b. Click **Create Dynamic Group** and enter all the required fields. Create matching rules using the following format for all your databases:

```
any { resource.id = 'here goes the OCID of your database 1',
resource.id = 'here goes the OCID of your database 2' }
```

Note

For managing dynamic groups, you must have one of the following privileges:

- You are a member of the **Administrators** group.
- You are granted the **Identity Domain Administrator** role or the **Security Administrator** role.
- You are a member of a group that is granted **manage identity-domains** or **manage dynamic-groups**.

4. Create a new policy.

The **ObjectStorageReadersPolicy** allows users in the `ObjectStorageReaders` group to download objects from any Object Storage bucket in the tenancy. You can also narrow the

scope to a specific compartment. The policy includes permissions to list the buckets, list objects in the buckets, and read existing objects in a bucket.

- a. In the Oracle Cloud Infrastructure console, click **Identity**, and then click **Policies**.
- b. Click **Create Policy**.
- c. For the policy name, enter **ObjectStorageReadersPolicy**.
- d. For the description, enter **Allow ObjectStorageReaders group to read objects and buckets**.
- e. From the **Compartment** list, select your root compartment.
- f. Add the following policy statement, which allows ObjectStorageReaders to read buckets:

```
Allow dynamic-group ObjectStorageReaders to read buckets in tenancy
```

- g. Add a second policy statement that allows ObjectStorageReaders to read objects in a bucket:

```
Allow dynamic-group ObjectStorageReaders to read objects in tenancy
```

- h. Click **Create**.

The screenshot displays the Oracle Cloud Infrastructure console interface for a policy named "ObjectStorageReadersPolicy". At the top, there are three buttons: "Edit Policy", "Add tags", and "Delete". Below these buttons, there are two tabs: "Policy Information" (selected) and "Tags". Under the "Policy Information" tab, the following details are shown:

- OCID:** ...d3f55kfa [Show](#) [Copy](#)
- Compartment:** jomadrigmx (root)
- Description:** Allow ObjectStorageReaders group to read objects and buckets
- Created:** Mon, Feb 13, 2023, 18:59:21 UTC

Below the "Policy Information" section, there is a "Statements" section. It contains a blue button labeled "Edit Policy Statements". Underneath this button, two policy statements are listed:

- Allow dynamic-group test_all to read buckets in tenancy
- Allow dynamic-group test_all to read objects in tenancy

At the bottom right of the "Statements" section, it says "Showing 2 Items".

Setting Up a Cloud Service Credential

1. In the Oracle Cloud Infrastructure console, click **Identity and Security** and click **Domains**.
2. Under List Scope, in the **Compartment** field, choose the root compartment from the drop-down list followed by the default domain.

3. In Default domain, click **Users** and then click **Create user**. In the Create user screen, select the **Administrators** group to have access to the buckets where the DMP files are stored.

Create user [Help](#)

First name *Optional*
DataPump

Last name
Test

Username
DataPumpTestCredentials

Email

Use the email address as the username

Groups *Optional*
Select groups to assign this user to.

Search...

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	All Domain Users	A group representing all users.
<input checked="" type="checkbox"/>	Administrators	Administrators

1 selected Showing 2 groups < Page 1 >

[Show advanced options](#) 

Create [Cancel](#)

4. After creating the user, in the left side of the new screen that appears, under **Resources**, select **API keys**.

Identity > Domains > Default domain > Users > DataPumpTestCredentials

DataPump Test

Edit user
Reset password
Reset factors
Edit user capabilities
More actions

User information
Tags
Capabilities

OCID: ...xyx57q Show Copy
Username: DataPumpTestCredentials
Prefix: -
First name: DataPump
Middle name: -
Last name: Test
Suffix: -

> User preferences
> Work information
> Other information

Resources

- Groups
- Integrated applications
- API keys

Assign user to groups
Remove user from group

<input type="checkbox"/>	Name
<input type="checkbox"/>	DataPumpTestCredentials

- Click **Add API key**.

Add API key [Help](#)

Note: An API key is an RSA key pair in PEM format used for signing API requests. You can generate the key pair here and download the private key. If you already have a key pair, you can choose to upload or paste your public key file instead. [Learn more](#)

Generate API key pair
 Choose public key file
 Paste a public key

Public key

Download the private key. It will not be shown again. After you download it, [change the file permissions](#) so only you can view it

Download private key
Download public key

Add
Cancel

- In the Add API Key screen, click **Download private key**. After downloading, click **Add**. The Configuration file preview screen appears.

Configuration file preview [Help](#)

Note: This configuration file snippet includes the basic authentication information you'll need to use the SDK, CLI, or other OCI developer tool. Paste the contents of the text box into your `~/oci/config` file and update the `key_file` parameter with the file path to your private key. [Learn more](#)

Select API key fingerprint

7a:10:c4:86:f8:6a:f6:9b:19:82:b9:d8:df:37:38:3f

Configuration file preview *Read-only*

```
[DEFAULT]
user=ocid1.user.oc1..aaaaaaaaav4fbtxlvrkzsjgsfitbtmeip6ukyxcjnwrxq45cnwqalxyx57q
fingerprint=7a:10:c4:86:f8:6a:f6:9b:19:82:b9:d8:df:37:38:3f
tenancy=ocid1.tenancy.oc1..aaaaaaaaatnfymioc5u5iht2u5fhngfirwixif44aiu4nh3xo5ybjbvsaobq
region=us-ashburn-1
key_file=<path to your private keyfile> # TODO
```

[Copy](#)
Paste the contents of the text box into your `~/oci/config` file.

[Close](#)

The **user**, **fingerprint** and **tenancy** information will be used later to create your user database.

7. Enter the database where you want to create the database user.

Overview > Autonomous Database > Autonomous Database details

ATP

test2

Always Free

Database actions ▼
Database connection
Performance hub

SQL

REST

Database Users

Data Load

View all database actions

Compartment: jomadrigmx (root)/Dynamic_G

OCID: ...kyfnla [Show](#) [Copy](#)

Created: Thu Nov 2 2023 17:04:23 UTC

8. The following code block, **DBMS_CLOUD.CREATE_CREDENTIAL**, is used to create the credential.

The Configuration file preview window information is used to generate the new credential. For the `private_key` attribute, you need to open the **.PEM** file that you downloaded in step 6, using any text editor. Generate a code block as shown in the following figure:

```

1 BEGIN
2   DBMS_CLOUD.CREATE_CREDENTIAL (
3     credential_name => 'NEW_TEST_CREDENTIAL',
4     user_ocid       => 'ocid1.user.oc1..aaaaaaaanv4fbtxlvrkzisjgsftbtmeip6ukyxcjnwrxq45cnwqalxyx57q',
5     tenancy_ocid   => 'ocid1.tenancy.oc1..aaaaaaaatnfyml0c5u5t2u5fhngftrwixlf44atu4nh3xo5ybjjvsaobq',
6     private_key    => 'MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwgg5jAgEAAoIBAQC98QaswRcPzbyS

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

PL/SQL procedure successfully completed.
Elapsed: 00:00:00.387

Importing Data

This section provides the steps for importing data using Oracle Data Pump in Database Actions.

1. In the Data Pump page, on the top right, click **Import**.

The Import wizard appears.

2. In the Source step:

Bucket Name Source:

- **Bucket List:** Allows you to select the bucket based on a bucket list after selecting a credential and the desired compartment.
- **Manual Search:** After selecting a valid credential, you must enter the exact name of a bucket to list its files.

After selecting the mode, enter the following fields:

- Credential Name:** Select a valid credential to access the information in the Object Storage Buckets.
- Compartment Name:** Select a compartment at any level within the tenancy (only available for Bucket List).
- Bucket Name:** Select the bucket that contains the dump files from the drop-down list. Selecting a bucket automatically prefills the associated dump files in the Bucket Objects field.
- Bucket Objects:** Select a dump file from the list.
- Import Pattern:** When you select a dump file, it is automatically entered in the Import Pattern field. You can modify the pattern, if needed. The dump files that match are displayed in the Dump Files field.
- Dump Files:** Select the dump files to import.

Click **Next**.

3. In the Import step, enter the following fields:

- **Import Name:** Enter a name for the import job.
- **Import Type:** Select the type of import. The options are Full, Tables, Schemas, and Tablespaces.

Note

If you select **Full**, you skip the Filter step in the wizard and directly go to the Mapping step.

- **Content:** Select **Data Only**, **DDL Only**, or **Data and DDL**.
- **Cloud Directory Name:** Select the directory to import to.
- **Encrypt:** Select if encrypted and enter an encryption password.

The screenshot shows the 'Import' wizard interface. At the top, a progress bar indicates the current step is 'Import' (step 2), with previous steps 'Source' (1) and 'Filter' (3) completed, and subsequent steps 'Mapping' (4), 'Options' (5), and 'Summary' (6) pending. Below the progress bar, the following fields are visible:

- Import Name ***: test
- Import Type ***: FULL (dropdown menu)
- Content ***: Data and DDL (dropdown menu)
- Cloud Directory Name ***: - Select - (dropdown menu)
 - DATA_PUMP_DIR (selected)
 - DBMS_OPTIM_ADMINDIR
 - DBMS_OPTIM_LOGDIR
 - EXT_DATA_DIR
 - HO_CLIENT_WALLETS_DIR

At the bottom left, there is a 'Show code' toggle switch which is currently turned off. At the bottom right, there are four buttons: 'Back', 'Next', 'Import', and 'Cancel'.

Click **Next**.

4. In the Filter step, depending on the import type, all the schemas, tables, or tablespaces for the import job are listed. Select the ones that apply. Click **Next**.
5. In the Mapping step, select the source schema and enter a new name for the target schema. If needed, do the same for tablespaces. Click **Next**.
6. In the Options step, enter the following fields:
 - **Threads:** Specify the maximum number of threads of active execution operating on behalf of the import job. The default is 1.
 - **Action on Table if Table Exists:** Specify the action needed if that table that import is trying to create already exists.
 - **Skip Unusable indexes:** Select to specify whether the import skips loading tables that have indexes that were set to the Index Unusable state.
 - **Regenerate Object IDs:** Select to create new object identifies for the imported database objects.

- **Delete Master Table:** Select to indicate whether the Data Pump control job table should be deleted or retained at the end of an Oracle Data Pump job that completes successfully.
- **Overwrite Existing Datafiles:** Select to indicate that if a table already exists in the destination schema, overwrite it.
- **Version:** Select the version of database objects to import.
- **Logging:** Select to create a log file. Enter the log directory and log file name.

Click **Next**.

7. The Summary step displays a summary of all the selections made in the previous steps. Select **Show Code** at the bottom to see the PL/SQL code equivalent of the form.

```

DECLARE
  L_JOB_STATE varchar2(1000);
  L_JOB_HANDLE number;
  L_HAS_FAILED VARCHAR2(100):= 'ERROR';
  L_SHOULD_TRY_TO_GET_STATUS number := 0;
  L_TABLE_EXISTS NUMBER;
BEGIN
  L_JOB_HANDLE := dbms_datapump.open( operation => 'IMPORT', job_mode => 'FULL', job_name => 'TEST_20_51_38', version => '(
  L_SHOULD_TRY_TO_GET_STATUS := 1;
  dbms_datapump.set_parallel( handle => L_JOB_HANDLE, degree => '1') );
  dbms_datapump.add_file( handle => L_JOB_HANDLE, filename => 'IMPORT-20_51_38.LOG', directory => 'DATA_PUMP_DIR', filetype =>
  dbms_datapump.set_parameter( handle => L_JOB_HANDLE, name => 'KEEP_MASTER', value => 1 );
  dbms_datapump.add_file( handle => L_JOB_HANDLE, filename => 'EXPDAT%u-2022-09-21-20_49_50.DMP', directory => 'DATA_PUMP_I
  dbms_datapump.set_parameter( handle => L_JOB_HANDLE, name => 'INCLUDE_METADATA', value => 1 );
  dbms_datapump.set_parameter( handle => L_JOB_HANDLE, name => 'DATA_ACCESS_METHOD', value => 'AUTOMATIC' );
  dbms_datapump.set_parameter( handle => L_JOB_HANDLE, name => 'REUSE_DATAFILES', value => 0 );
  dbms_datapump.set_parameter( handle => L_JOB_HANDLE, name => 'SKIP_UNUSABLE_INDEXES', value => 0 );
  dbms_datapump.start_job( handle => L_JOB_HANDLE, skip_current => 0, abort_step => 0 );
  dbms_datapump.wait_for_job( handle => L_JOB_HANDLE, job_state => L_JOB_STATE );
  dbms_datapump.detach( handle => L_JOB_HANDLE );
  L_HAS_FAILED := 'NO_ERROR';
EXCEPTION
  WHEN OTHERS THEN
    IF ( ( L_HAS_FAILED = 'ERROR' ) AND ( L_SHOULD_TRY_TO_GET_STATUS = 1 ) ) THEN
      DBMS_DATAPUMP.DETACH( L_JOB_HANDLE );
    END IF;

    :status_code := '400';
    HTP.PRN( '{ "errorCode": ' || SQLCODE || ', "errorDetails": ' || SQLERRM || ' }' );

EXECUTE IMMEDIATE
  'SELECT COUNT(*) FROM DBA_TABLES WHERE TABLE_NAME='||
  DBMS_ASSERT.ENQUOTE_LITERAL ( 'TEST_20_51_38' )
  INTO L_TABLE_EXISTS;

  IF L_TABLE_EXISTS <> 0 THEN
    EXECUTE IMMEDIATE
      'DROP TABLE' || DBMS_ASSERT.ENQUOTE_NAME ( 'TEST_20_51_38', FALSE );
  END IF;
END;
```

?
 Show code

Back
Next
Import
Cancel

Click **Import**.

The start of the job execution is displayed on the Data Pump page.

A

Supported SQL*Plus and SQLcl Commands in SQL Worksheet

The following sections list the SQL*Plus and SQLcl commands supported in the worksheet.

- [Supported SQL*Plus Commands](#)
The SQL worksheet supports most of the SQL*Plus commands except those statements that are related to formatting.
- [Supported SQLcl Commands](#)
The SQL worksheet supports many of the SQLcl commands.

Supported SQL*Plus Commands

The SQL worksheet supports most of the SQL*Plus commands except those statements that are related to formatting.

- / (slash)
- @@ { url | file_name[.ext] } [arg ...]
- ACC[EPT] variable [NUM[BER] | CHAR | DATE | BINARY_FLOAT | BINARY_DOUBLE] [FOR[MAT] format] [DEF[AULT] default] [PROMPT text | NOPR[OMPT]] [HIDE]
- ARCHIVE LOG LIST
- BRE[AK] [ON report_element [action [action]]] ...
- BTI[TLE] [printspect [text | variable] ...] | [ON | OFF]
- C[HANGE] sepchar old [sepchar [new [sepchar]]]
- CL[EAR] option ...
- COL[UMN] [{column | expr} [option ...]]
- COMP[UTE] [function [LAB[EL] text] ... OF {expr | column | alias} ...ON {expr | column | alias | REPORT | ROW} ...]
- COPY {FROM database | TO database | FROM database TO database} {APPEND | CREATE | INSERT | REPLACE} destination_table[(column, column, column, ...)] USING query
- DEF[INE] [variable] | [variable = text]
- DESC[RIBE] {[schema.]object[@connect_identifier]}
- DISC[ONNECT]
- EXEC[UTE] statement
- {EXIT | QUIT} [SUCCESS | FAILURE | WARNING | n | variable | :BindVariable] [COMMIT | ROLLBACK]
- GET [FILE] file_name[.ext] [LIST | NOLIST]
- HO[ST] [command]

- L[IST] [n | n m | n * | n LAST | * | * n | * LAST | LAST]
- PAU[SE] [text]
- PRINT [variable ...]
- PRO[MPT] [text]
- {QUIT | EXIT} [SUCCESS | FAILURE | WARNING | n | variable | :BindVariable]
[COMMIT | ROLLBACK]
- R[UN]
- SAV[E] [FILE] file_name[.ext] [CRE[ATE] | REP[LACE] | APP[END]]
- SET system_variable value
- SHO[W] [option]
- SHUTDOWN [ABORT | IMMEDIATE | NORMAL | TRANSACTIONAL [LOCAL]]
- STA[RT] { url | file_name[.ext] } [arg ...]
- STARTUP db_options | cdb_options | upgrade_options
- TIMI[NG] [START text | SHOW | STOP]
- TTI[TLE] [printspect [text | variable] ...] | [ON | OFF]
- UNDEF[INE] variable ...
- VAR[IABLE] [variable [type][=value]]
- XQUERY xquery_statement

Supported SQLcl Commands

The SQL worksheet supports many of the SQLcl commands.

- ALIAS
- APEX
- BRIDGE
- CTAS
- DDL
- FORMAT
- INFORMATION
- LOAD
- NET
- OERR
- RESERVED_WORDS
- SCRIPT
- SETERRORL
- SODA (See [SODA Commands](#))
- TNSPING
- TOSUB

- WHICH
- [SODA Commands](#)

SODA Commands

SODA (Simple Oracle Document Access) commands are supported in the SQL code editor. SODA allows schemaless application development using the JSON data model. The commands are:

- SODA create *<collection_name>* — Creates a new collection
- SODA list — Lists all the collections
- SODA get *<collection_name>* [-all | -f | -k | -klist] [{<key> | <k1> <k2> ... | <qbe>}] — Lists documents in the collection. Optional arguments:
 - all: Lists the keys of all documents in the collection
 - k: Lists documents matching the specific <key>
 - klist: Lists documents matching the list of keys
 - f: Lists documents matching the <qbe>
- SODA insert *<collection_name>* *<json_str | filename>* — Inserts a new document within a collection
- SODA drop *<collection_name>* — Deletes existing collection
- SODA count *<collection_name>* [*<qbe>*] — Counts number of documents inside collection. Optional parameter *<qbe>* returns number of matching documents
- SODA replace *<collection_name>* *<oldkey>* *<new_{str | doc}>* — Replaces one document with another
- SODA remove *<collection_name>* [-k | -klist | -f] {<key> | <k1> <k2> ... | <qbe>} — Removes documents from collection. Optional arguments:
 - k: Removes document in collection matching the specific <key>
 - klist: Removes document in collection matching the list <key1> <key2> ...
 - f: Removes document in collection matching <qbe>