

Oracle® Life Science AI Data Platform

User Guide



G41759-01
October 2025



Oracle Life Science AI Data Platform User Guide,
G41759-01

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

Primary Authors: (primary author), (primary author)

Contributing Authors: (contributing author), (contributing author)

Contributors: (contributor), (contributor)

Contents

About This Content

1 Explore Oracle Health Real-World Data

Sign in to Database Actions	2
Access the SQL Worksheet	2
Use the SQL Worksheet	2
Locate the Oracle Health Real-World Dataset	3
Run Queries on Oracle Health Real-World Data	5
Create New Datasets	8
Load New Data	8
Connect Using JDBC	8
Connect Using an Autonomous Database Wallet	9
Download an Autonomous Database Wallet	10
Make Data Available in Oracle Analytics Cloud	12
SQL Worksheet Guidelines	12

2 Analyze Health Data

Analyze Data Using Data Science Service Notebooks	2
Sign in to Data Science Service from OCI	2
Create Data Science Service Projects	3
Create Notebook Sessions Using JupyterLab	4
Configure Data Science Service Notebooks Using PySpark	4
Connect to Oracle Health Real-World Data from Notebooks	5
Authenticate with ADW Using a JDBC Connection String	5
Authenticate with ADW Using Resource Principal Authentication (Recommended)	6
Authenticate with ADW Using Credentials in Notebook (Not Recommended)	8
Install an ADW Wallet	9
Install Python Packages	11
FAQs for Data Science Service	12

3	Visualize Health Data	
	Sign In to Oracle Analytics Cloud	2
	View Existing Autonomous Database Connection	2
	Create a New Autonomous Database Connection	2
	View Existing Datasets	4
	Create New Health Datasets	5
	Enrich a Dataset	7
	View Workbooks	8
	Create a New Workbook	9
4	Oracle Health Real-World Data Assets	
	Oracle Health Real-World Data Core Model	1
	Real-World Data HIPAA Conformance for De-Identification	4
	Real-World Data Process Workflow and Real-World Data OMOP Common Data Model v5.4	4
	Real-World Data on Autonomous Data Warehouse	5
5	Appendix	
	Proprietary RWD Extension Data Model	1
	Index	

About This Content

Oracle Life Sciences AI Data Platform provides researchers and life sciences companies with a way to view deidentified patient data and create data visualizations.

Audience

This guide is intended for Oracle Life Sciences AI Data Platform users.

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.

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.

Introduction

Oracle Life Sciences AI Data Platform is a cloud-based platform that allows healthcare organizations, research institutes, and life sciences companies to find insights using large-scale, de-identified patient data sets.

Oracle Life Sciences AI Data Platform gathers clinical data from the Oracle Learning Health Network, which includes patient records contributed by many healthcare institutions. The data is ingested, normalized, standardized, and de-identified using the Expert Determination methodology as defined in HIPAA Privacy Rule § 164.514(b)(1), and made available in both the Oracle Core Data Model and the Observational Medical Outcomes Partnership (OMOP) Common Data Model, with mapping to standard clinical ontologies such as ICD-10 and SNOMED.

Access Oracle Life Sciences AI Data Platform through your organization's dedicated Oracle Cloud Infrastructure (OCI) tenant. Through the tenant, you can access Oracle Autonomous Data Warehouse (ADW) and the de-identified data. Then you can use the Oracle Data Science Service to access notebook interfaces to create SQL queries, create cohorts, and analytics using R, Python, and Spark. Once the notebooks have been created, you can use Oracle Analytics Cloud (OAC) to visualize and investigate the data and validate or disprove your research hypothesis.

You need to sign in to the following applications separately to use Oracle Life Sciences AI Data Platform. Currently, Oracle Life Sciences AI Data Platform does not support federated authentication:

Step	Tool
Explore Data	Database Actions
Visualize Data	OAC and Database Actions
Analyze and Model Data	Oracle Data Science Service

Oracle Analytics Cloud is the cloud-based platform that Oracle uses to enable customers to conduct analytics and business intelligence, analyze data, build dashboards, and apply analytics without needing to maintain physical infrastructure such as servers. While Oracle Analytics Cloud can connect to large datasets, performance and report response time could lag severely unless you curate the dataset in advance. For example, Real-World Data contains more than 30 billion records and queries could take hours or days without first filtering.

Oracle recommends using Oracle Analytics Cloud to:

- Prepare a cohort (or list) of patients in Database Actions, then gather the Real-World Data for those patients. That way you start with a more manageable dataset.
- Curate aggregates. Prepare summarized datasets grouping by attributes of interest instead of data at the most granular level.

Note

Oracle does not recommend connecting Oracle Analytics Cloud to the full Real-World Data dataset.

1

Explore Oracle Health Real-World Data

You can use Oracle Life Sciences AI Data Platform to query Oracle Health Real-World Data (ORWD).

- [Sign in to Database Actions](#)
Depending on how your organization has set up authentication, you can sign in to Database Actions by connecting to a single, default database or you can use a schema alias to connect to multiple databases. Sign in to Database Actions to view the data catalog and run queries.
- [Access the SQL Worksheet](#)
Use the SQL page in Database Actions to write and run SQL queries, then create database objects based on Real-World Data.
- [Use the SQL Worksheet](#)
Use the SQL page to interact with your database. You can create queries, and add or edit data. You can use SQL and PL/SQL statements in the worksheet to create tables, insert data, and select data from a table.
- [Locate the Oracle Health Real-World Dataset](#)
Each version of Oracle Health Real-World Data is stored in its own schema. Use the navigator to select the schema.
- [Run Queries on Oracle Health Real-World Data](#)
You can use the code editor on the SQL page to create and run SQL statements, PL/SQL scripts, and JavaScript code on the Oracle Health Real-World Data that your organization has access to.
- [Create New Datasets](#)
Real-World Data and Reference Dataset schema are read-only.
- [Load New Data](#)
If your organization has permission to do so, you can use the Data Load page in Database Actions to make that data available to your instance of Autonomous Database.
- [Connect Using JDBC](#)
You can use Java Database Connectivity (JDBC) to connect your data.
- [Connect Using an Autonomous Database Wallet](#)
Instead of using JDBC, you can use an ADW wallet to connect to other tools.
- [Download an Autonomous Database Wallet](#)
A wallet is a secure container that stores authentication credentials, such as secure socket layer certificates and keys for a secure connection to an Oracle database.
- [Make Data Available in Oracle Analytics Cloud](#)
Your Oracle Analytics Cloud (OAC) administrator can create a default connection to a shared ADW reporting schema (such as DW_CUST_REPORT), in Autonomous Data Warehouse (ADW).
- [SQL Worksheet Guidelines](#)
Adhering to operational standards ensures that your queries align with system performance, data security, and query efficiency. Consider the following guidelines when you use SQL Worksheet to develop SQL queries for an Autonomous Database Warehouse (ADW) instance.

Sign in to Database Actions

Depending on how your organization has set up authentication, you can sign in to Database Actions by connecting to a single, default database or you can use a schema alias to connect to multiple databases. Sign in to Database Actions to view the data catalog and run queries.

Database Actions allows you to query Real-World Data, which is housed in the Autonomous Data Warehouse. Database Actions allows you to use SQL to determine whether a study is possible with the data available in Real-World Data. Database Actions enables feasibility assessments. Clinical researchers who are familiar with SQL, data managers or biostatisticians who need feasibility insights, and research coordinators who support protocol design can use Database Actions to conduct feasibility and cohort discovery, and estimate the number of patients to see whether enough data is available for a study.

Note

Reach out to your database administrator for the Database Actions URL and your login credentials:

To sign in to Database Actions:

1. Access the Database Actions web address.
2. Enter the credentials provided by your administrator in the **Database Actions** sign-in screen.
3. Once successfully authenticated, access to the **Database Actions Launchpad**.
4. To change your password, open a SQL worksheet or connect using Oracle SQL Developer and enter the following statement: `alter user <your_user_name> identified by <new_password> replace <old_password>;`

See [Oracle Database Actions](#) for more information.

Access the SQL Worksheet

Use the SQL page in Database Actions to write and run SQL queries, then create database objects based on Real-World Data.

To access the SQL worksheet:

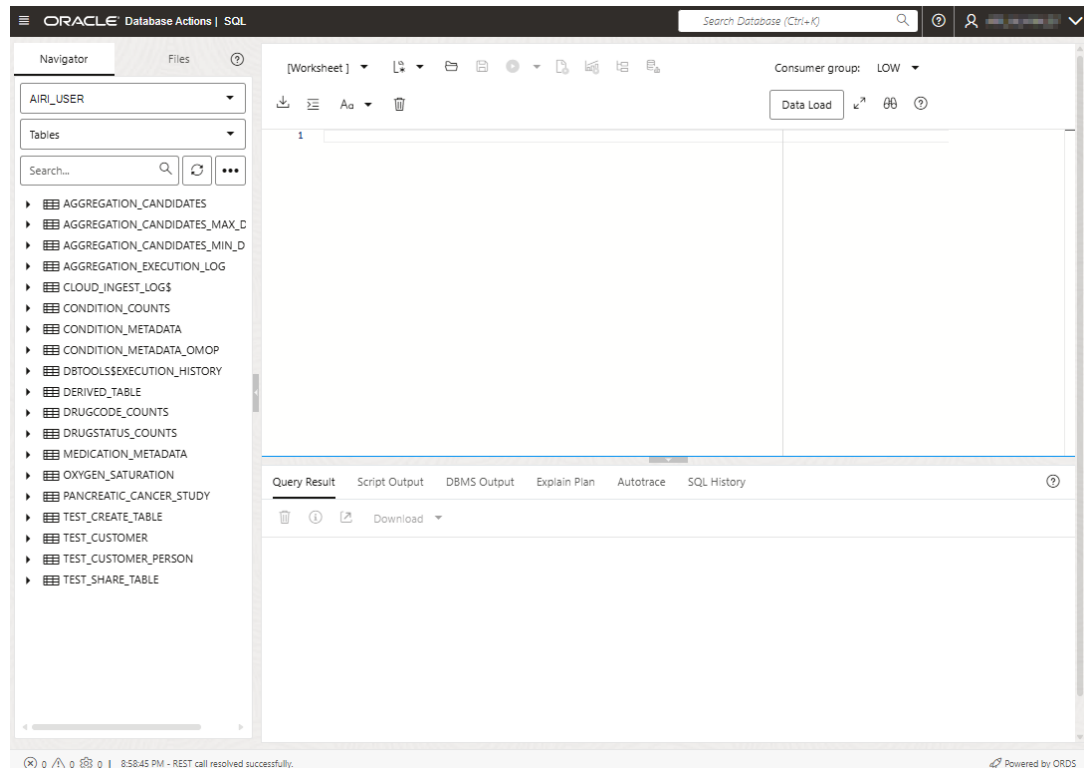
1. Sign in to Database Actions. The **Database Actions Launchpad** page is displayed.
2. Select the **Development** tab, select **SQL**, then select **Open**. Use the left pane to navigate between worksheets and objects, use the editor to execute SQL statements, and the output pane at the bottom to view results.

While it is beyond the scope of this document to provide details about creating SQL queries, see [The SQL Page](#) for more information.

Use the SQL Worksheet

Use the SQL page to interact with your database. You can create queries, and add or edit data. You can use SQL and PL/SQL statements in the worksheet to create tables, insert data, and select data from a table.

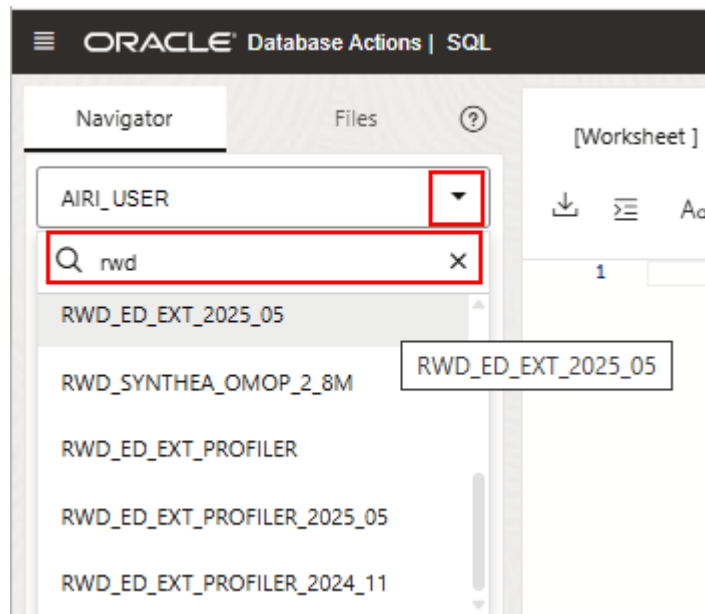
- The **Navigator** section allows you to view database objects including schema, tables, and so on.
- The **Worksheet** section allows you to enter your SQL and PL/SQL statements.
- The **Results** section displays information about statements after you run them.



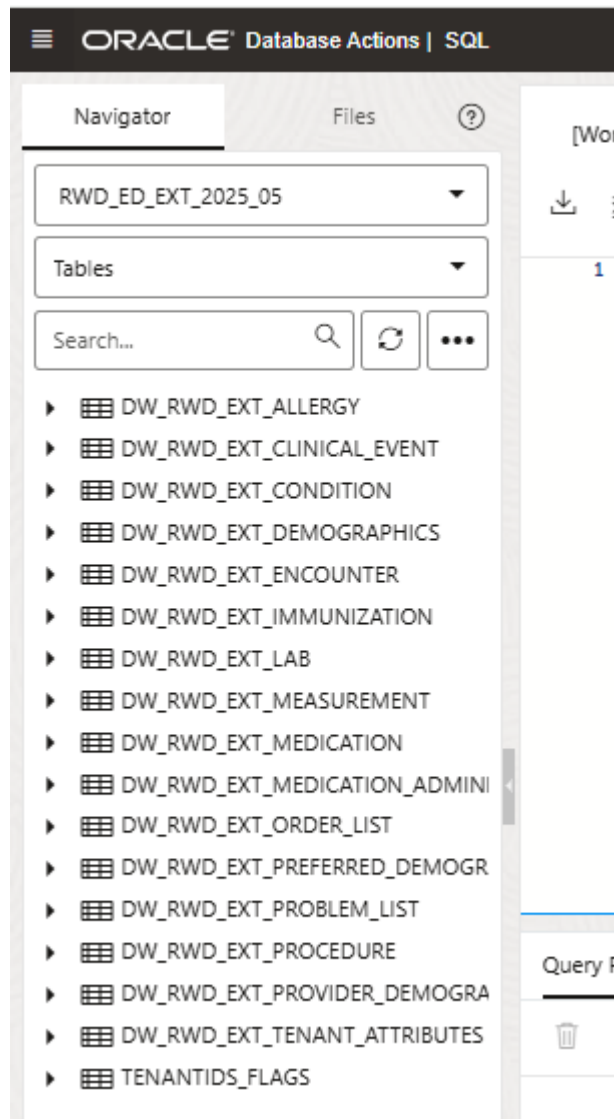
Locate the Oracle Health Real-World Dataset

Each version of Oracle Health Real-World Data is stored in its own schema. Use the navigator to select the schema.

1. Sign in to Database Actions and access the **SQL** page.
2. Select the **Navigator** tab.
3. Select the list next to your username and scroll or use the search to access the schema. Real-World Data schema starts with the prefix **RWD**.



Tables related to that schema are displayed.

**Note**

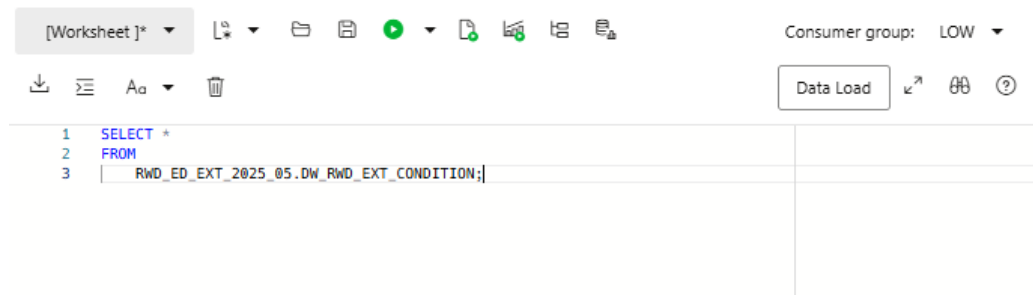
- You may see other schemas and usernames listed, but you can only access data if you are authorized to do so.
- Real-World Data and Reference Dataset schema are read-only.

Run Queries on Oracle Health Real-World Data

You can use the code editor on the SQL page to create and run SQL statements, PL/SQL scripts, and JavaScript code on the Oracle Health Real-World Data that your organization has access to.

1. Sign in to Database Actions, select the **Development** tab, then select **SQL**.
2. Access the Real-World Data catalog.
3. From the **New File** menu, select **Worksheet**.

4. Create the query in the code editor. See SQL Worksheet Guidelines for recommendations for creating SQL queries that perform well. The RWD database is large and could affect system performance if you do not follow the guidelines.
 - You can either write SQL statements directly in the worksheet editor, or you can drag objects from the **Navigators** section and drop them into the worksheet pane. If you drag and drop a table or view, you need to select Insert, Update, Select, or Delete. If you choose **Select**, the Select statement is constructed and includes all columns in the table or view. Then you can edit the statement by modifying the list of columns or by adding a WHERE clause.



- You can press **CTRL+SPACE** to view possible completions at the insertion point.
5. From the **Consumer Group** list, select the consumer group level. Oracle recommends developing in the Low consumer group until you have validated query performance because the real-world dataset contains more than two billion rows of data.
 6. When you have created a query that aligns with the guidelines, select **Run Script**. Depending on the amount of data returned, the query could take several minutes. The query results are displayed in the Query Result tab in the worksheet.

[Worksheet]* Consumer group: LOW

```

1 SELECT *
2 FROM
3   RWD_ED_EXT_2025_05.DW_RWD_EXT_CONDITION;

```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History

Download Execution time: 0.148 seconds

	CONDITIONID	PERSONID	ENCOUNTERID	CONDITIONCODE	EFFECTIVEDATE	ASSERTEDDATE	TYPE
1	01fb878f-8dc4-45...	eed3213-8735-4...	18a92e48-8dc1-4...	{"standard":{"id":"...	07/21/2022, 03:08...	07/21/2022, 03:08...	{"standard":
2	222c3309-1b50-4...	ebd3ce51-bc2b-4...	f6441b00-de22-4f...	{"standard":{"id":"...	12/15/2016, 01:26...	12/15/2016, 01:26...	{"standard":
3	09db929f-4be9-4...	a84708bd-f67e-48...	b40f3b2c-cb09-40...	{"standard":{"id":"...	(null)	(null)	{"standard":
4	98a0b913-2e4c-4...	19a59cb0-d573-4...	e2b9e43c-9baa-4...	{"standard":{"id":"...	(null)	(null)	{"standard":
5	2aafb733-94af-48...	336b8beb-b800-4...	2baa21c6-c9d4-4...	{"standard":{"id":"...	10/22/2024, 05:00...	10/22/2024, 12:40...	{"standard":
6	80d5b7d1-6732-4...	2dd7ba45-6eb9-4...	f4aa5164-fdc8-4d...	{"standard":{"id":"...	01/06/2025, 06:00...	01/06/2025, 02:04...	{"standard":
7	ced84f88-cb9e-4d...	7ab02f96-8f61-40...	9cafb67-f5f1-439...	{"standard":{"id":"...	11/12/2024, 06:00...	11/12/2024, 04:26...	{"standard":

7. Select from the following options:

- **Run Statement:** Run the selected statements or the statement at the pointer in the worksheet editor.
- **Run Script:** Runs all statements in the worksheet editor.
- **Compile:** Performs a PL/SQL compilation of the subprogram.
- **Create Chart:** Creates a chart for the corresponding SQL statement. You can also select a subquery to create a chart. A slider window is displayed to enter chart parameters. An error message is displayed if the SQL statement is incorrect or incomplete.
- **Explain Plan:** Generates the execution plan for the statement. The execution plan is displayed in the Explain Plan tab in the worksheet output pane.
- **Autotrace:** Runs the statement and collects run-time statistics and the actual execution plan.
- **Download Editor Content:** Downloads the worksheet as a SQL file to your 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:** Opens a guided tour of the worksheet.
- **Help:** Provides context-related assistance and a link to the documentation.

- **Open in Fullscreen:** Opens the editor in full screen mode.

For additional information about SQL syntax, see [SQL Language Reference](#).

Create New Datasets

Real-World Data and Reference Dataset schema are read-only.

To create derivatives of these datasets or to integrate your own data, use the default schema based on your username or a shared schema that you have write access to.

Load New Data

If your organization has permission to do so, you can use the Data Load page in Database Actions to make that data available to your instance of Autonomous Database.

You can load data from files or databases, from links to external databases, or from a live feed of data from cloud storage.

To access the Data Load page:

1. Sign in to Database Actions.
2. Select **Data Studio**.
3. Select **Data Load**.

The Data Load dashboard is displayed.

4. Use the available options to load data or link data. See [The Data Load Page](#) for more information.

Connect Using JDBC

You can use Java Database Connectivity (JDBC) to connect your data.

Note

Reach out to your database administrator for JDBC connection string details.

1. Ensure that you have access to Oracle SQL Developer and that Oracle JDBC drivers are installed.
2. In Oracle SQL Developer, select **New** then select **Connection**.
3. Select **Oracle** as the Database Type.
4. Enter your Autonomous Data Warehouse username and password.
5. Select **Custom JDBC** connection and paste the connection string:
`jdbc:oracle:thin:@(DESCRIPTION=(RETRY_COUNT=20)(RETRY_DELAY=3)
(ADDRESS=(PROTOCOL=tcps)(HOST=adb.<region>.oraclecloud.com)
(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=<service_name>.adb.oraclecloud.com))
(SECURITY=(SSL_SERVER_DN_MATCH=yes)))`

Connect Using an Autonomous Database Wallet

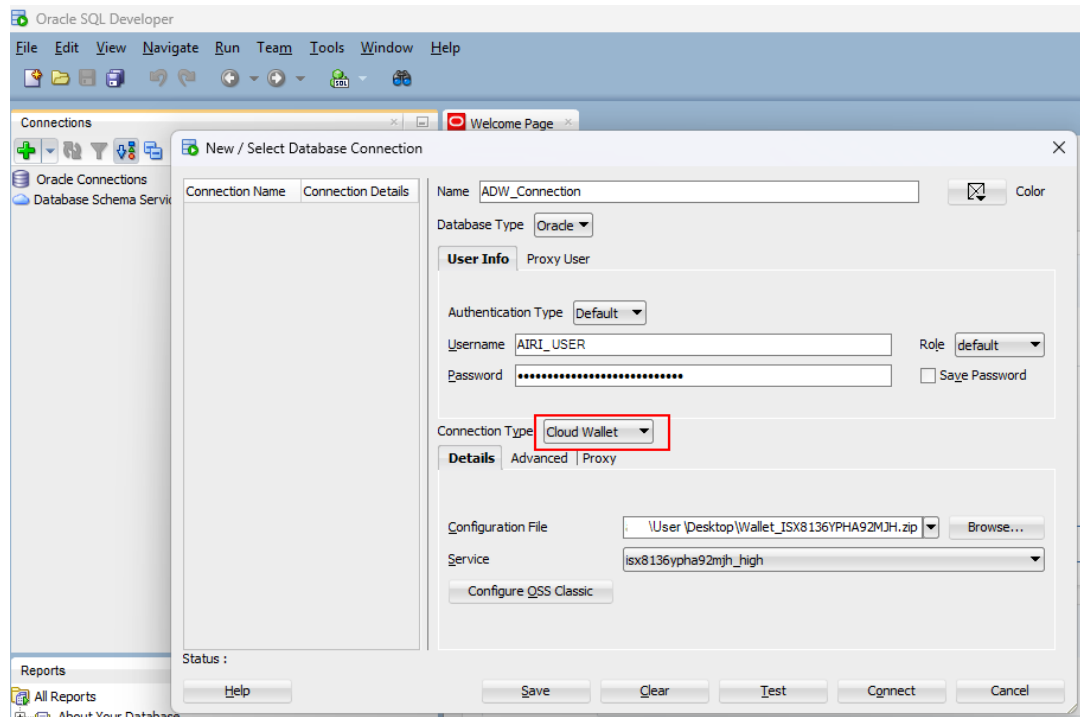
Instead of using JDBC, you can use an ADW wallet to connect to other tools.

Note

Reach out to your database administrator for ADW Cloud Wallet.

To connect using an ADW wallet:

1. In Oracle SQL Developer, select **New** then select **Connection**.
2. Enter your Autonomous Data Warehouse username and password.
3. Select the **Cloud Wallet** connection type.



4. Select **Browse** to access the wallet on your machine. See Download an Autonomous Database Wallet for more information.
5. Select an appropriate service level.
6. Select **Test** to test the connection.
7. Once the test is successful, select **Save**.

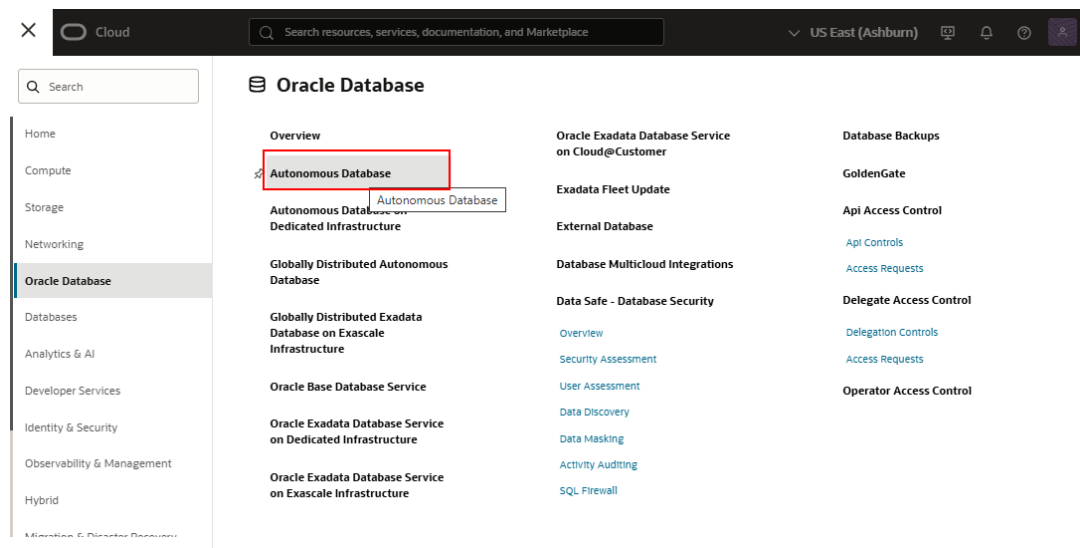
Download an Autonomous Database Wallet

A wallet is a secure container that stores authentication credentials, such as secure socket layer certificates and keys for a secure connection to an Oracle database.

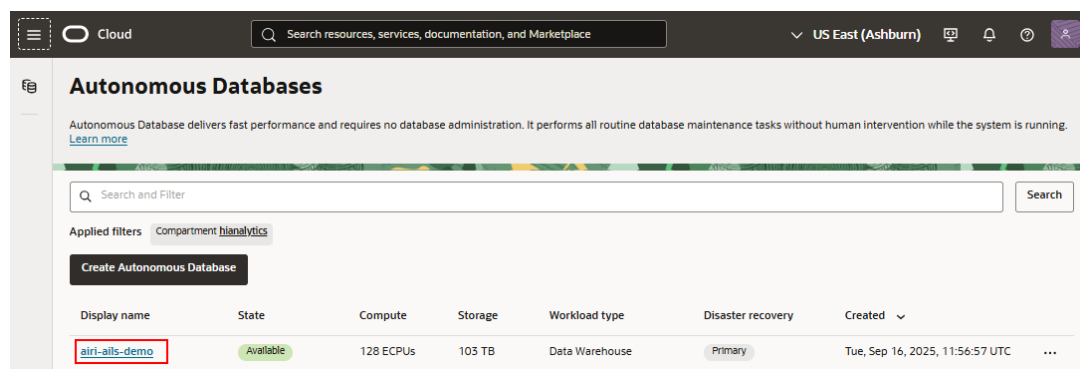
The wallet allows you to connect your notebook or code to the Autonomous Database over an encrypted channel using mutual TLS (mTLS) authentication without exposing passwords or private keys.

To connect to Autonomous Data Warehouse from Data Science Service or Database Actions, you need an ADW wallet.

1. In OCI Console, select the Navigation menu, select **Oracle Database**, then select **Autonomous Database**.



2. Select the Autonomous Database name. The Details page is displayed.



3. Select **Database Connection**, then select **Download Wallet**.

Search resources, services, documentation, and Marketplace

US East (Ashburn)

Database connection

Autonomous Database is preconfigured to support Oracle Net Services (a TNS listener is installed and configured to use secure TCPS). [Learn more](#)

Warning

Connections to your Autonomous Database are secured, and can be authorized using TLS or mTLS authentication options. TLS authentication is easier to use, provides better connection latency, and does not require you to download client credentials (wallet) if any of these is true for your connections:

- You are using JDBC Thin Client (version 12.2.0.1 or higher) with JDK 8(u163+) or higher.
- You are using the Python python-oracledb driver.
- You are using ODP.NET version 19.14 (or higher), or 21.5 (or higher).
- You are using an Oracle Call Interface based driver with Oracle Client libraries version 19.14 (or higher), or 21.5 (or higher).

[Learn more](#) about TLS authentication and how to enable it.

Download client credentials (Wallet)

To download your client credentials, select the wallet type, and click **Download wallet**. You then enter a password for the wallet. This client credential download only contains information for mTLS connections.

You do not need a wallet for TLS connections.

Wallet type

Instance wallet

Required

Download wallet

Rotate wallet

Wallet last rotated: -

Connection strings

Use the following connection strings or TNS names for your connections. See the [documentation](#) for details.

The TNS names indicate the service levels available.

These connection strings correspond with the TNS names for each service level.

TLS authentication

Mutual TLS

Required

Cancel

4. Create a wallet password and select **Download**.

Download wallet

Autonomous Database is preconfigured to support Oracle Net Services (a TNS listener is installed and configured to use secure TCPS). [Learn more](#)

Warning

Connections to your Autonomous Database are secured, and can be authorized using TLS or mTLS authentication options. TLS authentication is easier to use, provides better connection latency, and does not require you to download client credentials (wallet) if any of these is true for your connections:

- You are using JDBC Thin Client (version 12.2.0.1 or higher) with JDK 8(u163+) or higher.
- You are using the Python python-oracledb driver.
- You are using ODP.NET version 19.14 (or higher), or 21.5 (or higher).
- You are using an Oracle Call Interface based driver with Oracle Client libraries version 19.14 (or higher), or 21.5 (or higher).

[Learn more](#) about TLS authentication and how to enable it.

Please create a password for this wallet. Some database clients will require that you provide both the wallet and password to connect to your database (other clients will auto-login using the wallet without a password).

Required

Required

Cancel
Download

Make Data Available in Oracle Analytics Cloud

Your Oracle Analytics Cloud (OAC) administrator can create a default connection to a shared ADW reporting schema (such as DW_CUST_REPORT), in Autonomous Data Warehouse (ADW).

To make your data accessible to OAC without creating a new connection from OAC to ADW to an individual schema, you can create a new object in the DW_CUST_REPORT schema, which is visible to all OAC users.

You can use the tool of your choice to transform data (for example ADW SQL Worksheets, SQL Developer, or Data Science Service) with the data stored in the ADW schema.

SQL Worksheet Guidelines

Adhering to operational standards ensures that your queries align with system performance, data security, and query efficiency. Consider the following guidelines when you use SQL Worksheet to develop SQL queries for an Autonomous Database Warehouse (ADW) instance.

- **Query execution within SQL Worksheets is subject to a 5-minute timeout.** You can see if the query has timed out by looking at the indicator in the lower-left bar. If you are creating complex queries which are likely to take longer than five minutes, Oracle recommends:
 - Creating the query using DSS or [SQL Developer](#), which are not subject to timeout limitations.

- Use the EXPLAIN PLAN statement to generate the execution plan for the statement. Executing the EXPLAIN PLAN statement internally to identify opportunities for query optimization. The execution plan is automatically displayed in the Explain Plan tab in the worksheet output pane.
- Use the Consumer Group list to select the consumer group to run your SQL or PL/SQL code. ADW consumer groups are predefined resource management profiles like High, Medium, and Low and set different levels of CPU use. Oracle recommends LOW, but if that option is not feasible, then select **MEDIUM** to executing the statement. Do not use HIGH because it is reserved for administrative tasks.
- **Avoid large IN lists in the Where clause.** Optimize queries with large IN lists (for example, `SELECT * FROM RWD_ED_2024 WHERE person_id IN ('101', '102', '103' . . . )`) by joining to a table or using a CTE.
- **Avoid unnecessarily selecting all data.** Retrieve only the needed columns `SELECT <COLUMN_A> FROM . . .` instead of `SELECT * FROM . . .`
- Use `FETCH FIRST n ROWS ONLY` to limit the number of rows returned.
- **Share subsets of RWD data with OAC:** OAC is an enterprise cloud-based analytics and business intelligence (BI) platform, designed to let organizations analyze data, build dashboards and workbooks, and apply advanced analytics (AI/ML) without needing to manage heavy on-premises infrastructure. Oracle does not recommend connecting OAC to the full RWD dataset. Even though OAC can connect to large datasets (even datasets with hundreds of billions of rows), performance and report response time could be long. OAC is intended for smaller, curated datasets. The ORWD dataset is multiple terabytes, and some tables have more than 30 billion records. To improve performance, Oracle recommends preparing the data for use in OAC as follows:
 - Generate a subset of RWD with data for a patient cohort, yielding a smaller dataset representing 200,000 patients as opposed to the 100 million patients in the entire RWD dataset.
 - Group by attributes of interest instead of data at the most granular level to generate aggregates.
 - Once the subset of RWD is prepared and stored in a schema, use OAC to build visualizations and dashboards or workbooks based on that subset.

2

Analyze Health Data

Data Science Service (DSS) on Oracle Cloud Infrastructure (OCI) allows you to build study cohorts, manipulate data, apply statistical models, and develop and validate machine learning pipelines.

Data Science Service enables researchers to generate statistical insights and evidence to help further their research.

Data scientists, biostatisticians with Python and SQL skills, and qualitative clinical researchers can use DSS to construct cohorts from the Real-World Data dataset based on criteria defined in study design, perform advanced statistical analysis, handle complex data, and perform machine learning modeling on RWD data.

Data Science Service is a platform for data scientists to build, train, deploy, and manage machine learning models using Python and other open source tools. A JupyterLab environment is provided to experiment and develop models. You also can scale up model training using NVIDIA GPUs and distributed training.

This section contains the following tasks:

- [Analyze Data Using Data Science Service Notebooks](#)
Use Data Science Service to perform statistical analysis and modeling to answer your research questions. At a high level, you need to set up a workspace in Data Science Service, prepare the data, conduct analysis, interpret the results, then share the findings.
- [Sign in to Data Science Service from OCI](#)
Access Data Science Service from your organizations Oracle Cloud Infrastructure (OCI) console.
- [Create Data Science Service Projects](#)
Create projects in Data Science Service to organize your work. Projects contain a collection of Data Science resources such as notebooks. When you sign in to Data Science Service, access the **Projects** page.
- [Create Notebook Sessions Using JupyterLab](#)
Notebook sessions are interactive coding environments where you can build and train models. Oracle Life Sciences AI Data Platform uses Data Science Service from Oracle Cloud Infrastructure.
- [Configure Data Science Service Notebooks Using PySpark](#)
The Python3 conda environment is preinstalled in the notebook session. The conda environment is a Python3-based conda environment and has a minimal set of Python libraries installed.
- [Connect to Oracle Health Real-World Data from Notebooks](#)
There are a variety of methods to connect to Oracle Health Real-World Data from notebooks.
- [FAQs for Data Science Service](#)
Answers to frequently asked questions for Data Science Service in Oracle Life Sciences AI Data Platform.

Analyze Data Using Data Science Service Notebooks

Use Data Science Service to perform statistical analysis and modeling to answer your research questions. At a high level, you need to set up a workspace in Data Science Service, prepare the data, conduct analysis, interpret the results, then share the findings.

To analyze data using Data Science Service notebooks:

1. In Data Science Service, create a new project, notebook sessions, and notebooks.
2. Import the data and variables into your notebooks, then ensure that the data is accurate (for example, the data has no logical inconsistencies or outliers).
3. Analyze your data and generate tables and charts and adjust as necessary to assess patterns, effects, or predictors of outcomes.
4. Document and share your results, if applicable.

See [Data Science Service](#) for more information.

Sign in to Data Science Service from OCI

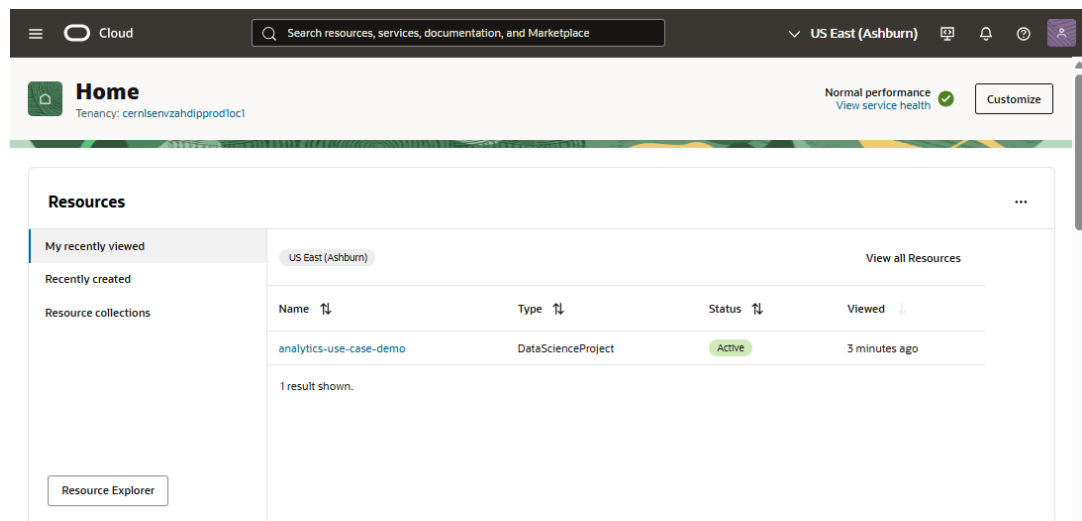
Access Data Science Service from your organizations Oracle Cloud Infrastructure (OCI) console.

Note

These steps are for administrators who have access to the Oracle Cloud Account. Contact your Oracle representative if you have any questions.

To sign in to Data Science Service for the first time:

1. From cloud.oracle.com, enter your cloud account name and select **Next**.
2. Select the Default domain, then select **Next**.
3. Enter your username and password, then select **Sign In**. Depending on your site's policies, you may need to use a second factor to authenticate.
4. Open the Navigation menu, select **Analytics & AI**, then select **Data Science** under Machine Learning. The Data Science Service home page is displayed.



Create Data Science Service Projects

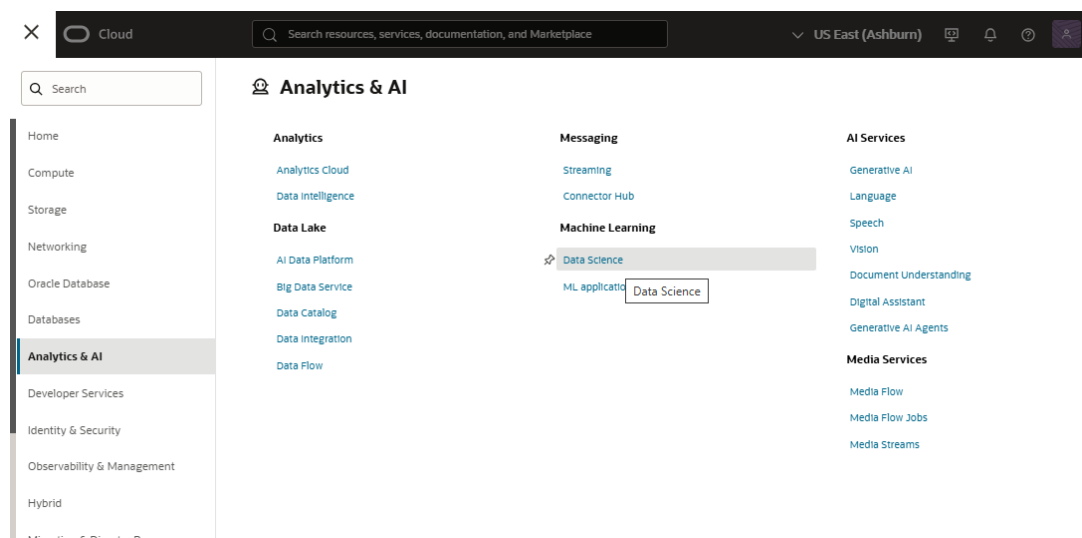
Create projects in Data Science Service to organize your work. Projects contain a collection of Data Science resources such as notebooks. When you sign in to Data Science Service, access the **Projects** page.

Available Project Actions

- Select a project.
- Use the Actions menu to view details, edit projects, move the project, or delete the project.
- From the List Scope filter, you can view projects in another compartment.
- Filter based on project status using the Project State list.
- If tags have been applied to projects, you can select **Add** or **Clear** next to Tag Filters.

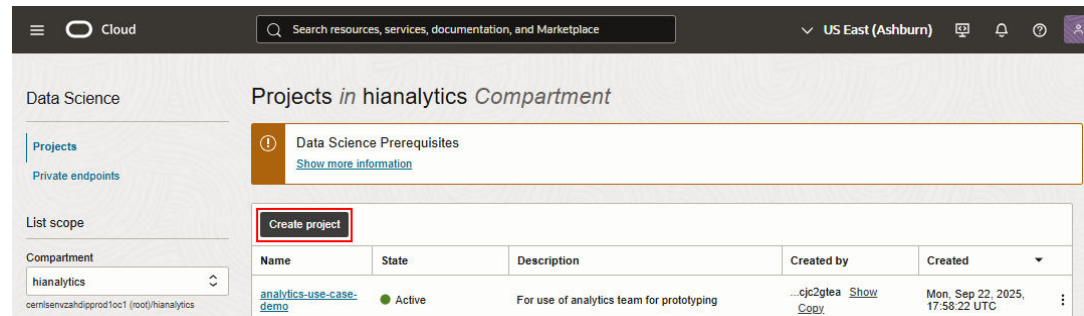
To create a project:

1. From the navigation menu, select **Analytics & AI**, then select **Data Science**.



The Projects page is displayed.

2. Select **Create Project**.



3. Enter a unique project name.
4. Enter a description for the project.
5. To add tags, select **Show Advanced Options** In the Tags section, add tags to the resourceType.
6. If you want to view details for the project after you create the project, select **View detail page on clicking create**.
7. Select **Create**. The Data Science Service project is created.

Create Notebook Sessions Using JupyterLab

Notebook sessions are interactive coding environments where you can build and train models. Oracle Life Sciences AI Data Platform uses Data Science Service from Oracle Cloud Infrastructure.

To create a notebook session:

1. From the Data Science projects section in the OCI Console, select **Create Notebook Session**. The Create Notebook Session page is displayed.
2. Enter a name and description.
3. Select the network setup based on your configuration.
4. Enter any custom environment variables and Git repository web addresses.
5. Enter any storage mounts.
6. Select **Create**. After a few minutes, the Notebook Sessions page opens.

Configure Data Science Service Notebooks Using PySpark

The Python3 conda environment is preinstalled in the notebook session. The conda environment is a Python3-based conda environment and has a minimal set of Python libraries installed.

To install a PySpark conda environment:

1. From the Data Science Service notebook session, select **File**, then **New Launcher**, then **Terminal**.
2. Enter your credentials to access the notebook. The notebook is displayed.

3. In the notebook, select **File** then **New Launcher**, then **Terminal** or select the Terminal icon if the launcher is already open. A terminal session window opens.
4. Enter the following command to install the PySpark dependency:

```
odsc conda install -s pyspark35_p311_cpu_x86_64_v1
```
5. To create a notebook that uses the PySpark kernel, select **File**, then select **New** and select the **pyspark35_p311_cpu_x86_v1** kernel.
6. From the menu on the left, select **spark_config_dir**, select the **spark-default.conf** file, add the configurations below to the bottom of the file, then save.

```
spark.hadoop.fs.oci.client.hostname=https://objectstorage.us-  
ashburn-1.oraclecloud.com  
spark.hadoop.oci.metastore.uris=https://datacatalog.us-  
ashburn-1.oci.oraclecloud.com/  
spark.hadoop.fs.oci.client.custom.authenticator=com.oracle.bmc.hdfs.auth.Re  
sourcePrincipalsCustomAuthenticator  
spark.hadoop.oracle.dcat.metastore.client.custom.authentication_provider=co  
m.oracle.bmc.hdfs.auth.ResourcePrincipalsCustomAuthenticator
```

Connect to Oracle Health Real-World Data from Notebooks

There are a variety of methods to connect to Oracle Health Real-World Data from notebooks.

Use one of the following methods to connect to Oracle Health Real-World Data from notebooks:

- [Authenticate with ADW Using a JDBC Connection String](#)
You can sign in to Autonomous Data Warehouse (ADW) using a JDBC connection string.
- [Authenticate with ADW Using Resource Principal Authentication \(Recommended\)](#)
Resource Principal Authentication lets your notebook use other OCI services securely without explicit credentials.
- [Authenticate with ADW Using Credentials in Notebook \(Not Recommended\)](#)
Although you can execute ADW queries from the notebook using credentials in your code, for security reasons this process is not a recommended authentication approach because it shares the same password.
- [Install an ADW Wallet](#)
Autonomous Database Warehouse (ADW) Wallets provide the connection information that you need to connect a data source to a notebook.
- [Install Python Packages](#)
Oracle recommends using conda environments to package python dependencies inside notebook sessions.

Authenticate with ADW Using a JDBC Connection String

You can sign in to Autonomous Data Warehouse (ADW) using a JDBC connection string.

To authenticate using a JDBC connection string:

1. To connect to ADW from DSS, you can use an ADW wallet. See [How to get an ADW Connection String](#) for more information.

2. Use the connection string, with your credentials, to connect to ADW.

Set up ADW Connection

```

]: #Load packages for connection
import oracledb
import os
import oci
import base64
import time
import warnings

warnings.filterwarnings('ignore')

#Load standard ds packages
import matplotlib.pyplot as plt
import pandas as pd
from getpass import getpass

]: #Replace with your own user credential
#Note: connection string has to be a private endpoint within DSS
jdbc_connection_str = "(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)(host=leazlcci.adb.us-ashburn-1.oraclecloud.com)))"
user_name = "<MY_USERNAME>"
pwd = getpass("Enter password: ")

]: # Setup the ADW connection
connection = oracledb.connect(
    user=user_name,
    password=pwd,
    dsn=jdbc_connection_str
)
cursor = connection.cursor()

]: list_schemas = f"""
SELECT table_name
FROM all_tables
WHERE owner = 'RWD_OMOP_2025_05'
ORDER BY table_name
"""

]: cursor.execute(list_schemas)
print(cursor.fetchall())

```

Authenticate with ADW Using Resource Principal Authentication (Recommended)

Resource Principal Authentication lets your notebook use other OCI services securely without explicit credentials.

Resource Principal Authentication is a secure authentication method offered by Oracle Cloud Infrastructure (OCI) that allows cloud-native resources (like OCI Data Science notebooks, functions, or OCI-based virtual machines and containers) to interact with other OCI services (for example, Vault, Object Storage, Database) without explicitly managing user credentials or API keys in your code.

When you run code in a Data Science notebook, OCI manages a special identity (resource principal) for the session. Identity and Access Management (IAM) policies assigned to the project or notebook session govern the identity, permitting or restricting which OCI resources it can access. See [Resource Principal](#) to access OCI Resources for more information.

```

ADW_Connect.ipynb
[ ]: #Load packages for connection
import oracledb
import os
import oci
import base64

# Step 1: Use resource principal signer (since you're running inside a Data Science notebook)
signer = oci.auth.signers.get_resource_principals_signer()

# Step 2: Use the correct SecretsClient from `vault_secrets` module
secrets_client = oci.secrets.SecretsClient(config={}, signer=signer)

# Step 3: Provide your secret OCID
## SWAP OUT

db_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_VAULT_SECRET"
wallet_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_WALLET_SECRET"
user_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_USER_SECRET"

# Step 4: Get the secret bundle
db_secret = secrets_client.get_secret_bundle(db_secret_ocid)
wallet_secret = secrets_client.get_secret_bundle(wallet_secret_ocid)
user_secret = secrets_client.get_secret_bundle(user_secret_ocid)

# Step 5: Decode the base64-encoded content
encoded_db_secret = db_secret.data.secret_bundle_content.content
decoded_db_secret = base64.b64decode(encoded_db_secret).decode("utf-8")

encoded_wallet_secret = wallet_secret.data.secret_bundle_content.content
decoded_wallet_secret = base64.b64decode(encoded_wallet_secret).decode("utf-8")

encoded_user_secret = user_secret.data.secret_bundle_content.content
decoded_user_secret = base64.b64decode(encoded_user_secret).decode("utf-8")

# Set TNS_ADMIN to the wallet directory
os.environ["TNS_ADMIN"] = "/home/datascience/Wallet_DEMO/"
connection = oracledb.connect(
    user=decoded_user_secret,
    password=decoded_db_secret,
    #!NEVER SET TO HIGH!
    dsn="MY_INSTANCE_MEDIUM",
    config_dir="/home/datascience/Wallet_DEMO/",
    wallet_location="/home/datascience/Wallet_DEMO/",
    wallet_password=decoded_wallet_secret
)
cursor = connection.cursor()

```

```

#Load packages for connection
import oracledb
import os
import oci
import base64

```

```

# Step 1: Use resource principal signer (since you're running inside a Data
Science notebook)
signer = oci.auth.signers.get_resource_principals_signer()

```

```

# Step 2: Use the correct SecretsClient from `vault_secrets` module
secrets_client = oci.secrets.SecretsClient(config={}, signer=signer)

```

```

# Step 3: Provide your secret OCID
## SWAP OUT

```

```

db_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_VAULT_SECRET"
wallet_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_WALLET_SECRET"
user_secret_ocid = "ocid1.vaultsecret.oc1.iad.MY_OCID_USER_SECRET"

# Step 4: Get the secret bundle
db_secret = secrets_client.get_secret_bundle(db_secret_ocid)
wallet_secret = secrets_client.get_secret_bundle(wallet_secret_ocid)
user_secret = secrets_client.get_secret_bundle(user_secret_ocid)

# Step 5: Decode the base64-encoded content
encoded_db_secret = db_secret.data.secret_bundle_content.content
decoded_db_secret = base64.b64decode(encoded_db_secret).decode("utf-8")

encoded_wallet_secret = wallet_secret.data.secret_bundle_content.content
decoded_wallet_secret =
base64.b64decode(encoded_wallet_secret).decode("utf-8")

encoded_user_secret = user_secret.data.secret_bundle_content.content
decoded_user_secret = base64.b64decode(encoded_user_secret).decode("utf-8")

# Set TNS_ADMIN to the wallet directory
os.environ["TNS_ADMIN"] = "/home/datascience/Wallet_DEMO/"
connection = oracledb.connect(
    user=decoded_user_secret,
    password=decoded_db_secret,
    #!NEVER SET TO HIGH!
    dsn="MY_INSTANCE_MEDIUM",
    config_dir="/home/datascience/Wallet_DEMO/",
    wallet_location="/home/datascience/Wallet_DEMO/",
    wallet_password=decoded_wallet_secret
)
cursor = connection.cursor()

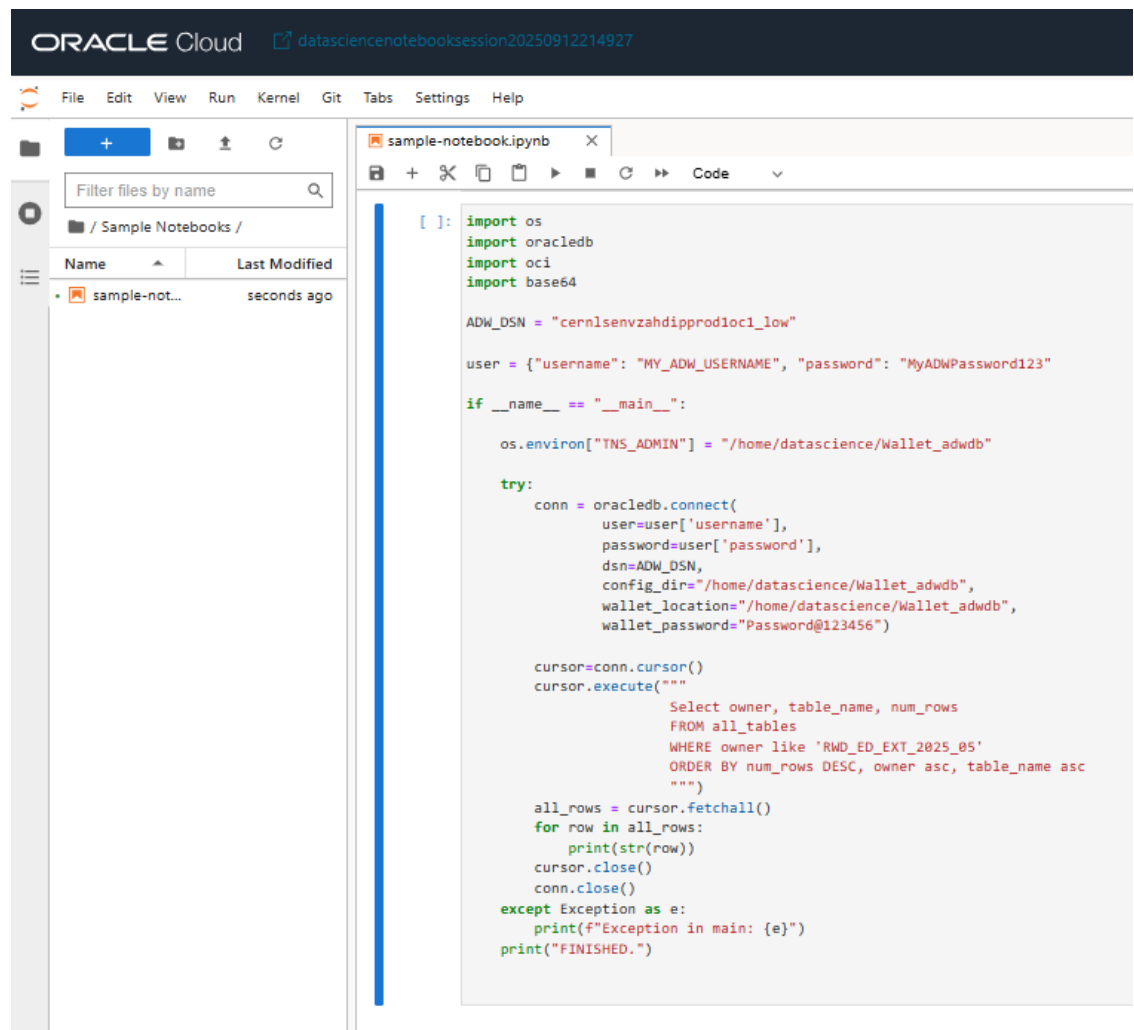
```

You can share the code, but other users cannot access your secrets just by having the secret OCIDs. Even if someone knows your `db_secret_ocid`, they must be running a resource (like a notebook session) with IAM permissions to both access OCI Vault to retrieve that specific secret. Never share secret OCIDs unless absolutely necessary, and always use OCI Vault and IAM to tightly control which identities and resources can access sensitive secrets.

Authenticate with ADW Using Credentials in Notebook (Not Recommended)

Although you can execute ADW queries from the notebook using credentials in your code, for security reasons this process is not a recommended authentication approach because it shares the same password.

Although you can execute ADW queries from the notebook using credentials in your code, for security reasons this process is not a recommended authentication approach because it shares the same password.



Install an ADW Wallet

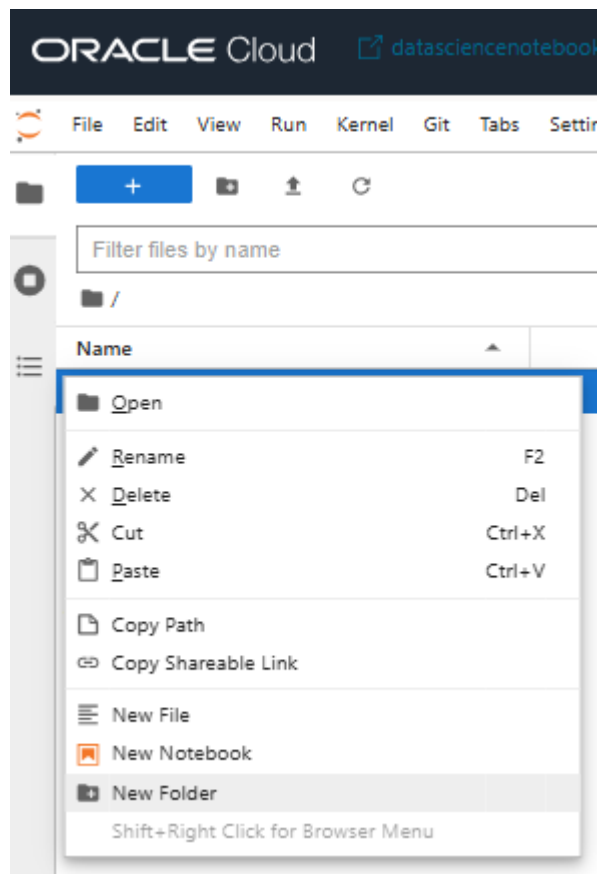
Autonomous Database Warehouse (ADW) Wallets provide the connection information that you need to connect a data source to a notebook.

Note

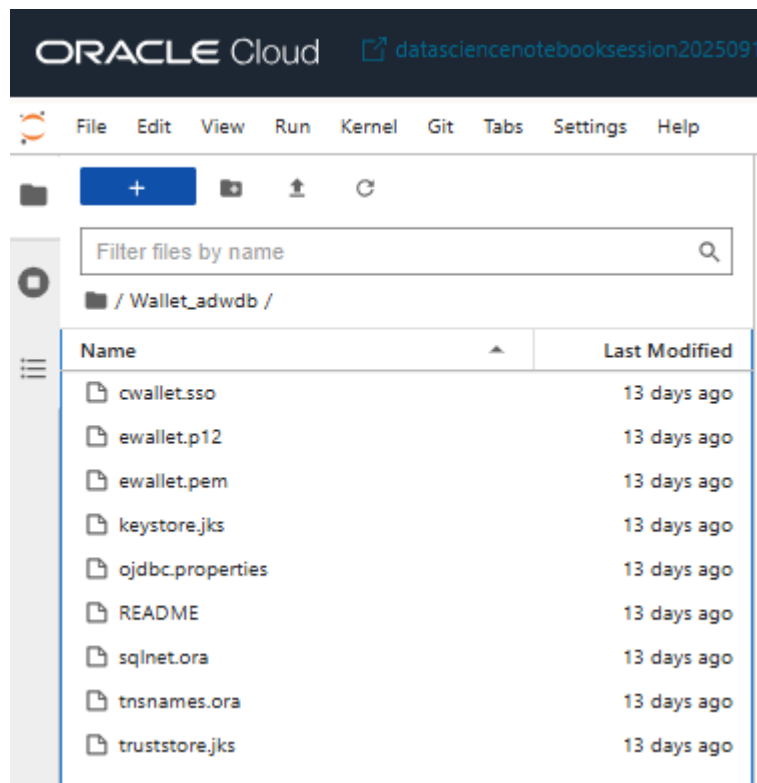
Reach out to your database administrator for more information about ADW Cloud Wallet.

To connect to ADW from Data Science service:

1. To connect to ADW from DSS, obtain the ADW wallet. See [Download an Autonomous Database Wallet](#) for more information.
2. Open your DSS Notebook, right-click the file, and select **New Folder** to create a folder for the wallet (**Wallet_adwdb**, for example).



3. Unzip the wallet .ZIP files downloaded from ADB and paste the files into the folder you created in Step 2.



Install Python Packages

Oracle recommends using conda environments to package python dependencies inside notebook sessions.

To view the list of packages installed in the conda environment, access a terminal window or a notebook running inside the conda environment kernel and enter `conda list`.

```

  Tabs  Settings  Help

  [X] Launcher  [X] datascience@:~

  (base) bash-4.4$ conda list
  # packages in environment at /opt/conda:
  #
  # Name                                Version      Build                Channel
  _libgcc_mutex                         0.1          conda_forge          conda-forge
  _openmp_mutex                         4.5          2_gnu                conda-forge
  aiofiles                             22.1.0       pyhd8ed1ab_0         conda-forge
  aiohappyeyeballs                     2.6.1        pypi_0               pypi
  aiohttp                              3.12.15      pypi_0               pypi
  aiosignal                            1.4.0        pypi_0               pypi
 aiosqlite                             0.21.0       pyhaa4b35c_0         conda-forge
  annotated-types                      0.7.0        pypi_0               pypi
  anyio                                 4.10.0       pyhe01879c_0         conda-forge
  anywidget                            0.9.13       pypi_0               pypi
  archspec                             0.2.3        pyhd8ed1ab_0         conda-forge
  argon2-cffi                          25.1.0       pyhd8ed1ab_0         conda-forge
  argon2-cffi-bindings                 25.1.0       py311h49ec1c0_0     conda-forge
  arrow                                 1.3.0        pyhd8ed1ab_1         conda-forge
  asteval                              1.0.6        pypi_0               pypi
  asttokens                            3.0.0        pyhd8ed1ab_1         conda-forge
  attrs                                 25.3.0       pyh71513ae_0         conda-forge
  babel                                 2.17.0       pyhd8ed1ab_0         conda-forge
  beautifulsoup4                       4.13.5       pyha770c72_0         conda-forge
  bleach                               6.2.0        pyh29332c3_4         conda-forge
  bleach-with-css                       6.2.0        h82add2a_4           conda-forge
  blessed                              1.21.0       pypi_0               pypi
  bokeh                                 3.1.1        pypi_0               pypi
  boltons                              23.1.1       pyhd8ed1ab_0         conda-forge
  branca                                0.8.1        pypi_0               pypi
  brotli-python                        1.1.0        py311h1ddb823_4     conda-forge
  bzip2                                1.0.8        hd590300_5           conda-forge
  c-ares                               1.34.5       hb9d3cd8_0           conda-forge
  ca-certificates                      2025.8.3     hbd8a1cb_0           conda-forge
  cached-property                      1.5.2        h28ad1ab_1           conda-forge

```

To directly install libraries, use pip or conda commands in a notebook. The libraries are installed in the conda environment that corresponds to the notebook kernel.

To install libraries from a terminal window tab:

1. Enter **conda activate <my_environment_name>**. Where <my_environment_name> is the name of your environment.
2. Enter **conda install**.

See [Conda Environments](#) for more information.

FAQs for Data Science Service

Answers to frequently asked questions for Data Science Service in Oracle Life Sciences AI Data Platform.

How do I extend my notebook session?

DSS Notebook sessions are subject to a 60-minute timeout by default. Select **Extend** to extend the session length.

3

Visualize Health Data

After you have connected your data to your notebooks, you can visualize the data.

Oracle Analytics Cloud allows you to share insights and make decisions. You can use OAC to create interactive visualizations and dashboards.

Researchers can synthesize their findings, identify insights, and share results with stakeholders. Principal investigators, research administrators, institutional stakeholders, and clinicians who want to view data without coding can use Oracle Analytics Cloud.

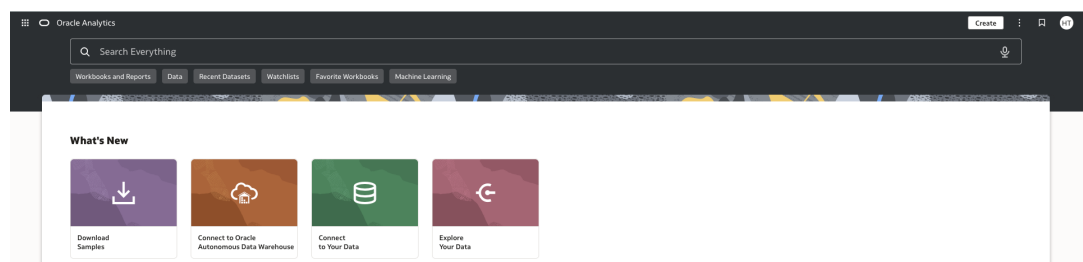
- [Sign In to Oracle Analytics Cloud](#)
Oracle Analytics Cloud (OAC) is a comprehensive, cloud-based platform that provides secure and scalable services for data exploration, data preparation, and collaborative analytics, enabling users of all technical skill levels to derive insights and make better business decisions. Use Oracle Analytics Cloud to create data visualizations to investigate research hypothesis.
- [View Existing Autonomous Database Connection](#)
Your Oracle Analytics Cloud instance may be configured by your administrator with a connection to shared reporting schema on ADW, where ADW users have write privileges.
- [Create a New Autonomous Database Connection](#)
Oracle Health Real-World Data is the database that contains Real-World Data from the Oracle Health Information Network.
- [View Existing Datasets](#)
Datasets are subsets of the data source's data that you use to meet your specific analytics and visualization needs. Use datasets to build data models for your visualization and analysis requirements.
- [Create New Health Datasets](#)
Datasets can include tables from multiple data source connections, and you can add joins and enrich data. For example, you may want to connect to the ADW database. Datasets are subsets of the file or data source's data that you curate and shape to meet your specific analytics and visualization needs. Datasets are self-service data models that you build specifically for your data visualization and analysis requirements. A dataset can be based on one table, spreadsheet, or file. Or a dataset can be a self-service data model that contains multiple tables with relationships defined between the tables. You can create datasets from data from files that you upload, or SaaS applications, Oracle Analytics reports, and many relational and big data sources that you build connections to.
- [Enrich a Dataset](#)
Before you deploy your visualization, you may want to enrich and transform your data. For example, you might rename data columns or add calculations to make the information more user-friendly.
- [View Workbooks](#)
Workbooks contain any in-progress visualizations and items.
- [Create a New Workbook](#)
Use workbooks to store and organize analytics content like graphs and charts.

Sign In to Oracle Analytics Cloud

Oracle Analytics Cloud (OAC) is a comprehensive, cloud-based platform that provides secure and scalable services for data exploration, data preparation, and collaborative analytics, enabling users of all technical skill levels to derive insights and make better business decisions. Use Oracle Analytics Cloud to create data visualizations to investigate research hypothesis.

To sign in to Oracle Analytics Cloud:

1. Locate the Oracle Analytics Cloud email provided by your OAC administrator and your single sign-on credentials.
2. Enter your credentials.
3. Select **Sign In**. The home page is displayed.



See [Oracle Analytics Cloud \(OAC\)](#) for more information.

View Existing Autonomous Database Connection

Your Oracle Analytics Cloud instance may be configured by your administrator with a connection to shared reporting schema on ADW, where ADW users have write privileges.

To view the connection:

1. From the navigator, select **Data**, then select **Connections**. The list of existing connections is displayed.
2. Select a connection to view the tables and schemas associated with the connection. You can drag and drop or double-click tables to add them to the dataset. You also can enter **Connection** in the search bar at the top of the page to view connections.

Create a New Autonomous Database Connection

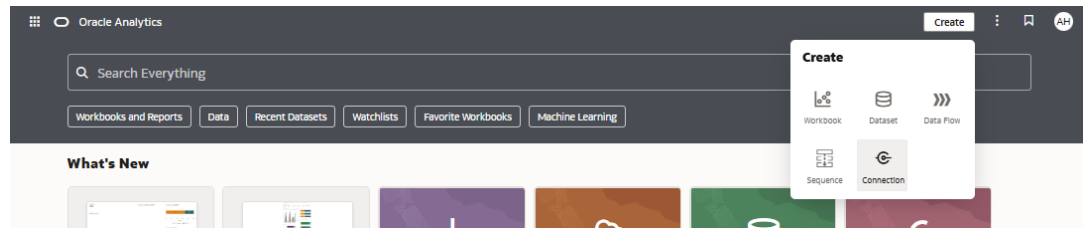
Oracle Health Real-World Data is the database that contains Real-World Data from the Oracle Health Information Network.

Note

Oracle does not recommend connecting OAC to the full RWD dataset.

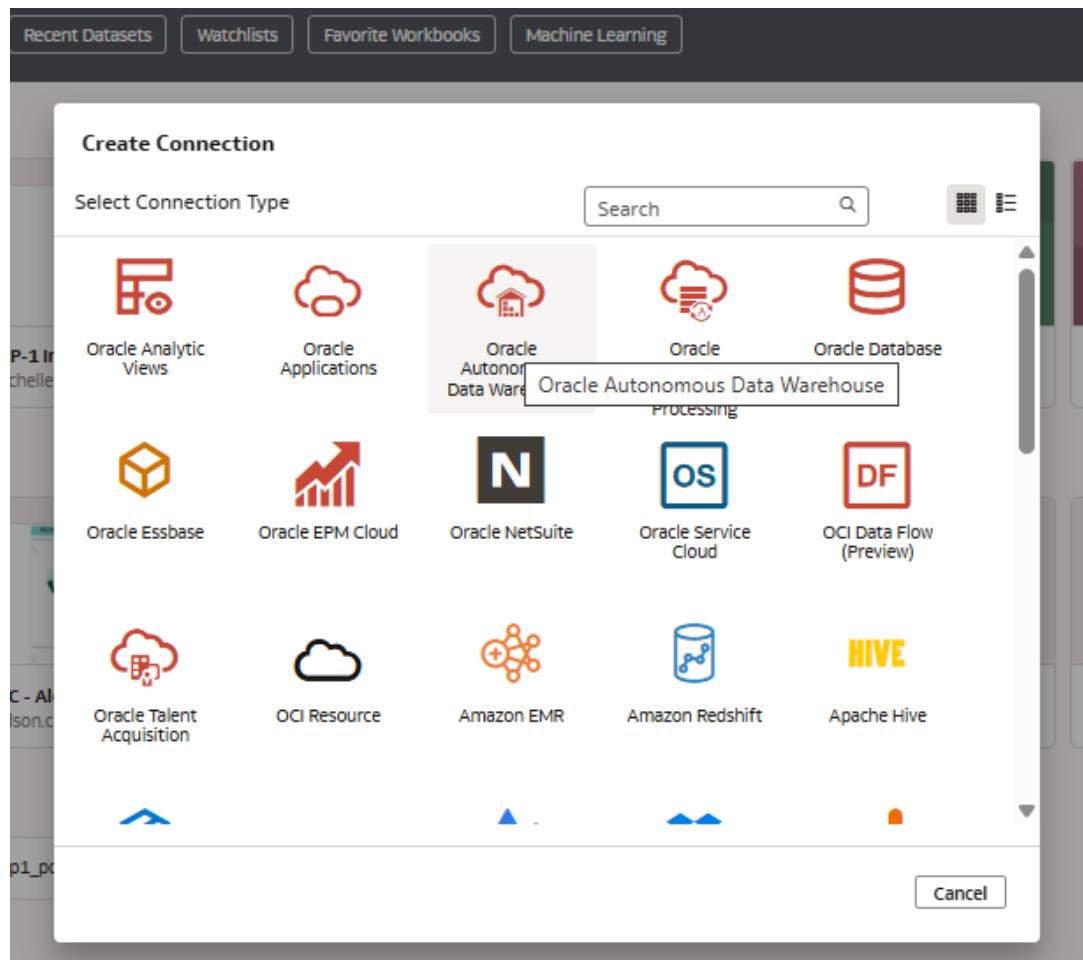
To connect to Oracle Health Real-World Data from Oracle Analytics Cloud:

1. From the home page of Oracle Analytics Cloud, select **Create**, then **Connection**.

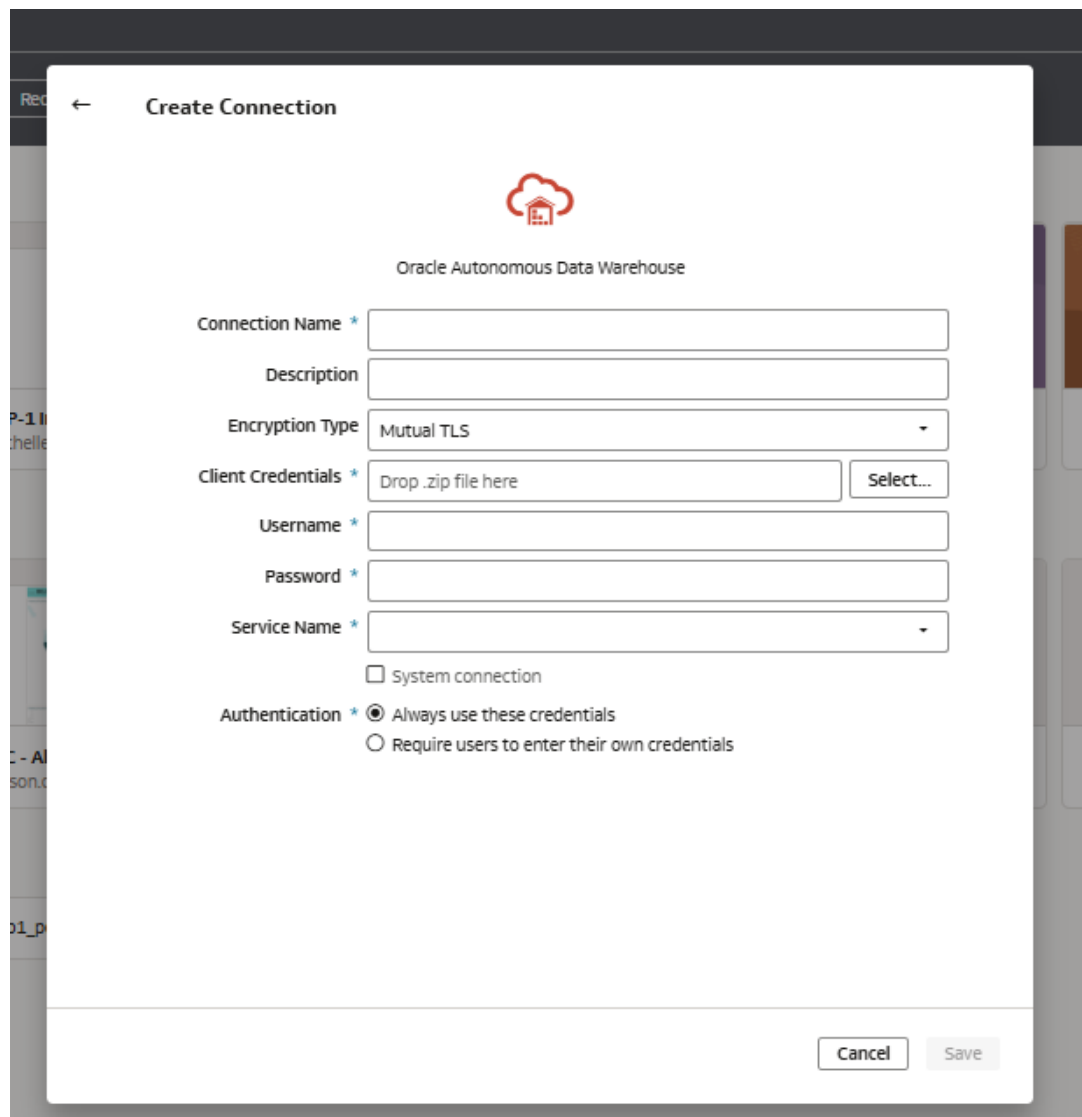


The Create Connection dialog box is displayed.

2. In the Select Connection Type section, select the icon for the connection type that you want. For example, **Oracle Autonomous Data Warehouse**.



3. You can connect to Autonomous Data Warehouse using security certificates from ADW to a wallet, (known as Mutual Transport Layer Security (mTLS)), or without a wallet (known as Transport Layer Security (TLS)). See [Connecting Oracle Analytics Cloud to Your Data](#) for more information.



The screenshot shows a 'Create Connection' dialog box for Oracle Autonomous Data Warehouse. The dialog has a title bar with a back arrow and the text 'Create Connection'. Below the title bar is the Oracle logo and the text 'Oracle Autonomous Data Warehouse'. The form contains the following fields and options:

- Connection Name ***: A text input field.
- Description**: A text input field.
- Encryption Type**: A dropdown menu with 'Mutual TLS' selected.
- Client Credentials ***: A text input field with the placeholder 'Drop .zip file here' and a 'Select...' button.
- Username ***: A text input field.
- Password ***: A text input field.
- Service Name ***: A dropdown menu.
- ☐ System connection
- Authentication ***: Two radio buttons: 'Always use these credentials' (selected) and 'Require users to enter their own credentials'.

At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

Note

If required, contact your ADW administrator for ADW wallet or JDBC string and ADW credentials.

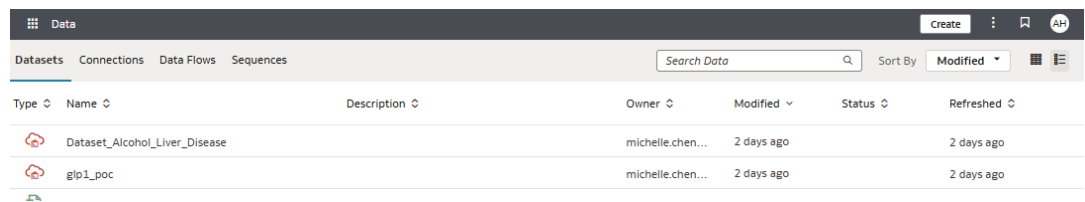
4. Once you have entered the connection information, select **Save**.

View Existing Datasets

Datasets are subsets of the data source's data that you use to meet your specific analytics and visualization needs. Use datasets to build data models for your visualization and analysis requirements.

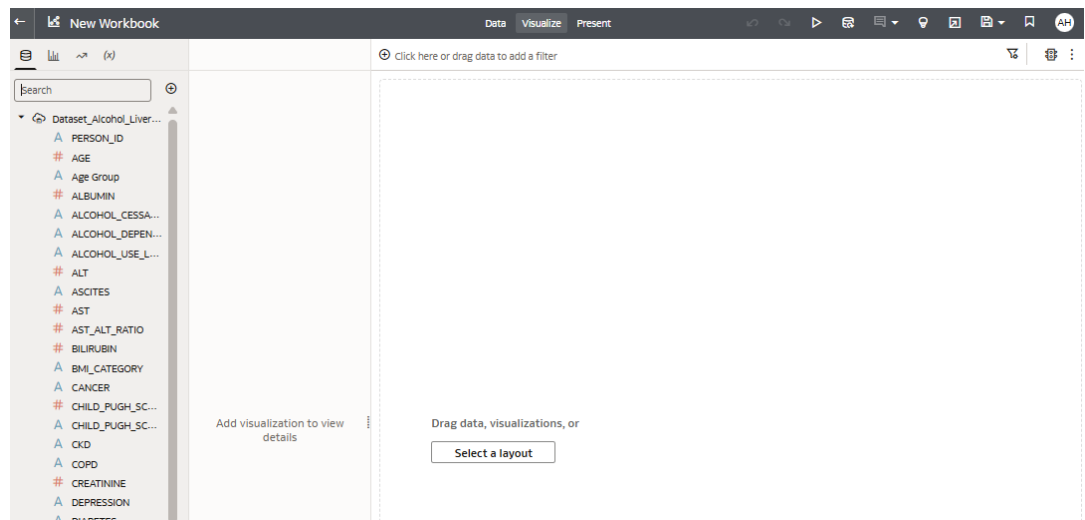
To view existing datasets:

1. In Oracle Analytics Cloud, from the navigator, select **Data**, then select **Datasets**. The list of existing datasets is displayed.



Type	Name	Description	Owner	Modified	Status	Refreshed
	Dataset_Alcohol_Liver_Disease		michelle.chen...	2 days ago		2 days ago
	glp1_poc		michelle.chen...	2 days ago		2 days ago

2. Select a dataset to open a new workbook, where you can explore and build visualizations using the data.



Create New Health Datasets

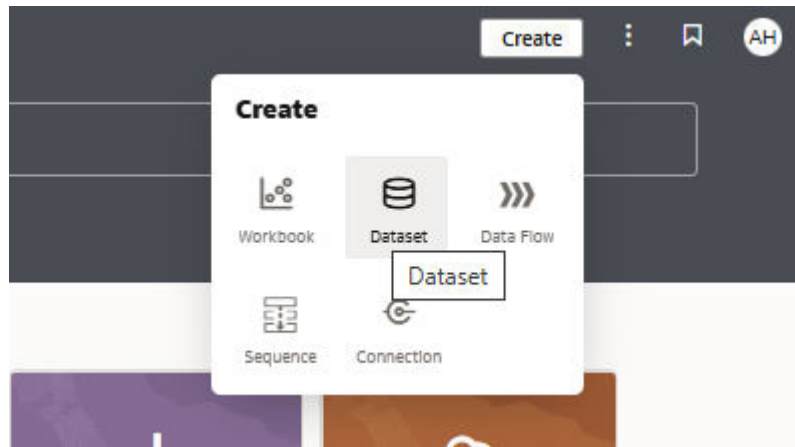
Datasets can include tables from multiple data source connections, and you can add joins and enrich data. For example, you may want to connect to the ADW database. Datasets are subsets of the file or data source's data that you curate and shape to meet your specific analytics and visualization needs. Datasets are self-service data models that you build specifically for your data visualization and analysis requirements. A dataset can be based on one table, spreadsheet, or file. Or a dataset can be a self-service data model that contains multiple tables with relationships defined between the tables. You can create datasets from data from files that you upload, or SaaS applications, Oracle Analytics reports, and many relational and big data sources that you build connections to.

Note

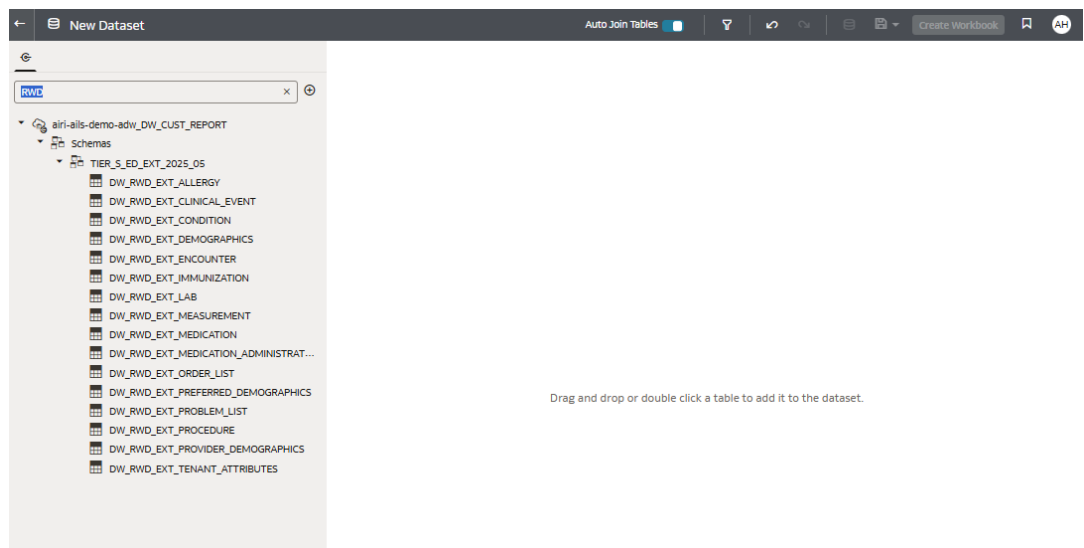
Oracle does not recommend connecting Oracle Analytics Cloud to the full dataset. Prepare your data for use by Oracle Analytics Cloud in Autonomous Data Warehouse to prevent performance issues. See [Share Subsets of RWD Data With OAC](#) for more information.

To create a dataset:

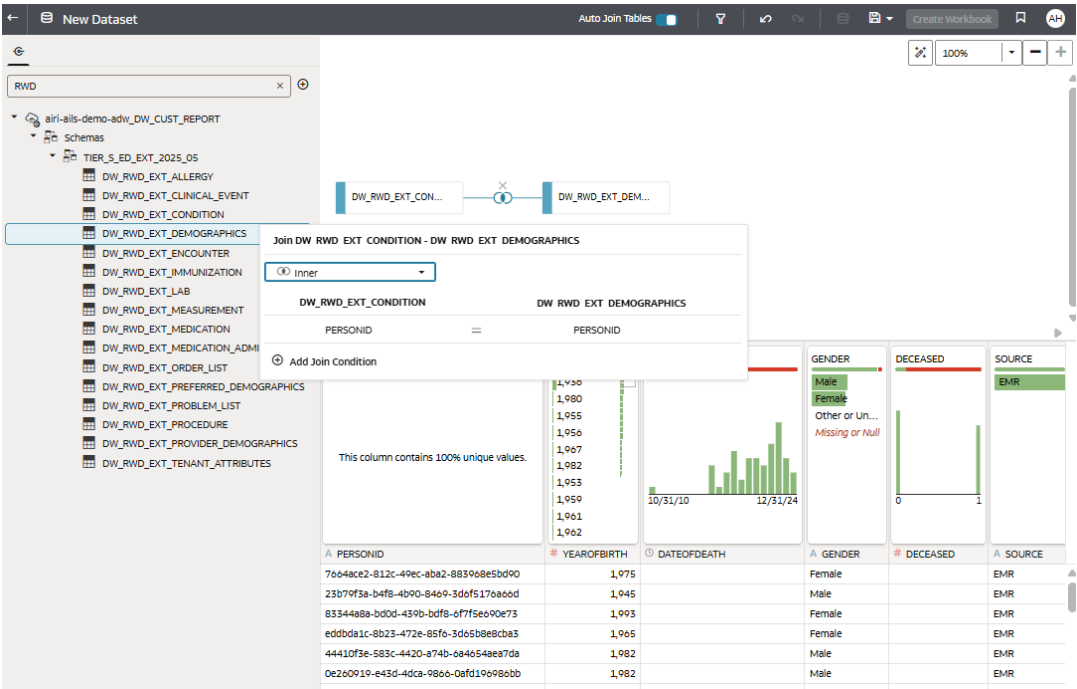
1. From the home page of Oracle Analytics Cloud, select **Create**, then **Dataset**. The Create Dataset dialog box is displayed.



2. Select the connection to ADW. The ADW schemas are displayed.
3. From the Connections pane, search for a schema.



4. Drag and drop or double-click a table to add it to the dataset. It may take a few seconds for the data to be displayed.
5. Drag and drop a second table. When you add tables to the Join diagram to create a multiple-table dataset, joins that are defined in the data source are automatically created between tables in the dataset. OAC also automatically creates joins based on column name matches it finds between tables. When you define your dataset, add the most detailed table first. This table is typically the primary table for your analysis.
6. To build dataset joins manually, turn off automatic joins. Selecting the join icon between two tables provides additional details and options to edit the join.

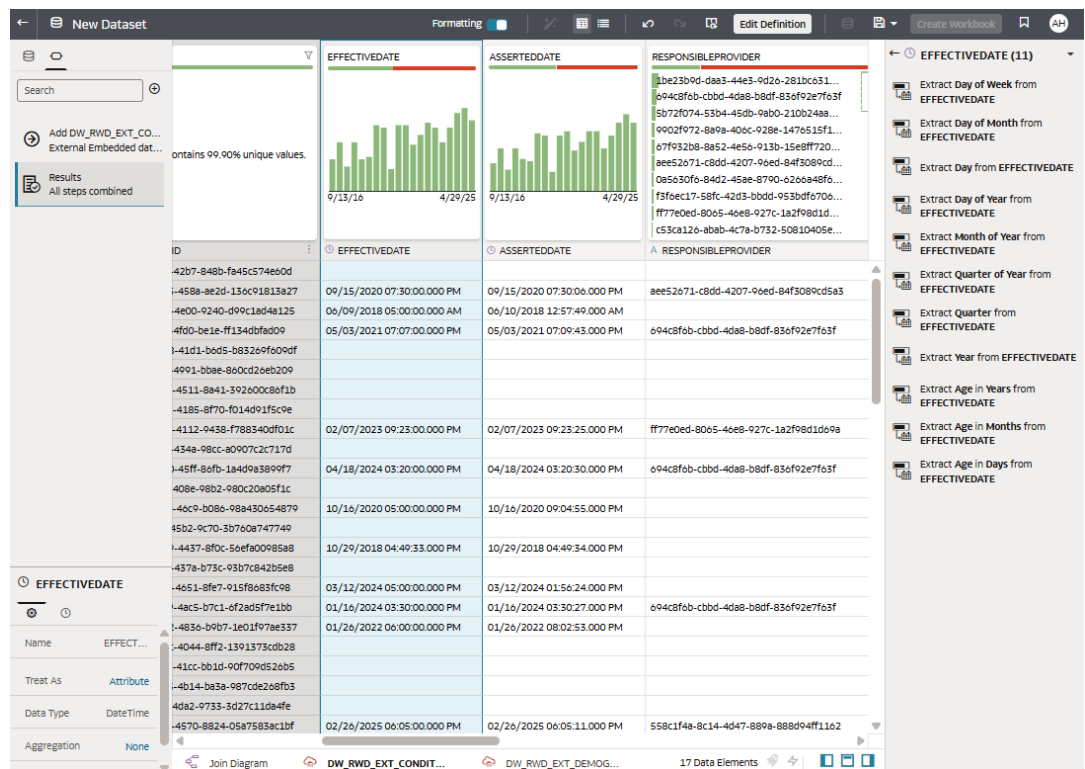


Enrich a Dataset

Before you deploy your visualization, you may want to enrich and transform your data. For example, you might rename data columns or add calculations to make the information more user-friendly.

To enrich a dataset:

- 1. Select a table from the bottom bar.



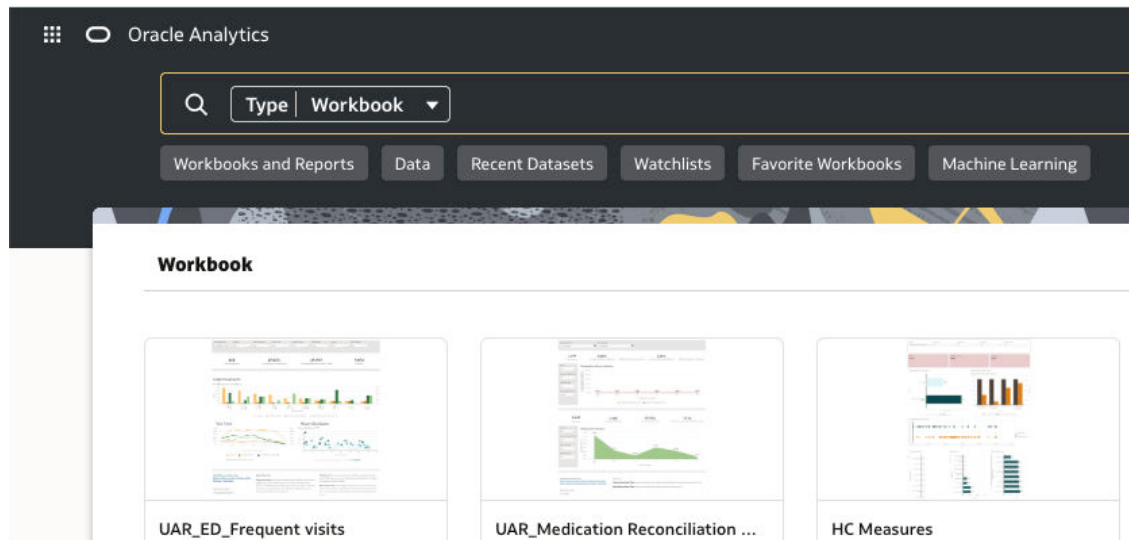
2. Select columns and use the enrich and transform tools to enhance your data.
3. Use the **Recommendations** panel for suggested enrichments.
4. Use the **Options** menu at the top of each column to apply common transformations such as Rename, Uppercase, and Trim.
5. From the **Options** menu at the top of a column, select **Edit** to enhance columns with functions and expressions from the Oracle Analytics functions library. For example, aggregates, strings, expressions, and math functions.
6. Select **Add Preparation Step** in the Data Panel to add a column based on a custom transformation. You can build your column using functions and expressions from the Oracle Analytics functions library. For example, aggregates, strings, expressions, and math functions.
7. Use the Properties pane at the bottom of the Data panel to review and change a column's type and aggregation. For example, you might select **Treat As** and change from an attribute to a measure, or change the default aggregation type from Sum to Average.
8. Use the Quality Insights tile above each column to explore and enrich your data.
9. Use the **Review auto Treat-as recommendations** option to change the default column type identified by the semantic profiler. For example, if the semantic profiler identifies a column with numeric IDs such as 1078220 as a measure, you can change the column to an attribute.

See [About Data Enrichment and Transformation in OAC](#) for more information.

View Workbooks

Workbooks contain any in-progress visualizations and items.

To view an existing workbook, enter **Workbooks** in the Search Everything bar, then select **Workbooks and Reports**.

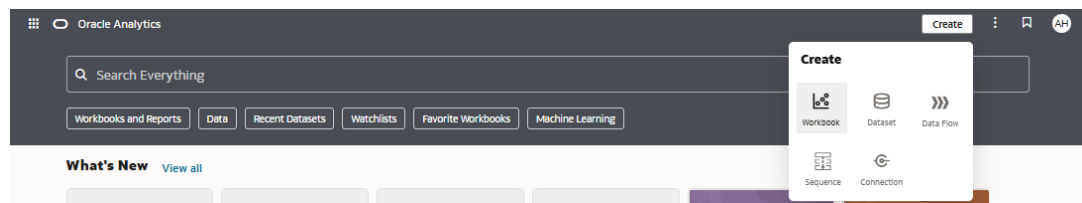


Create a New Workbook

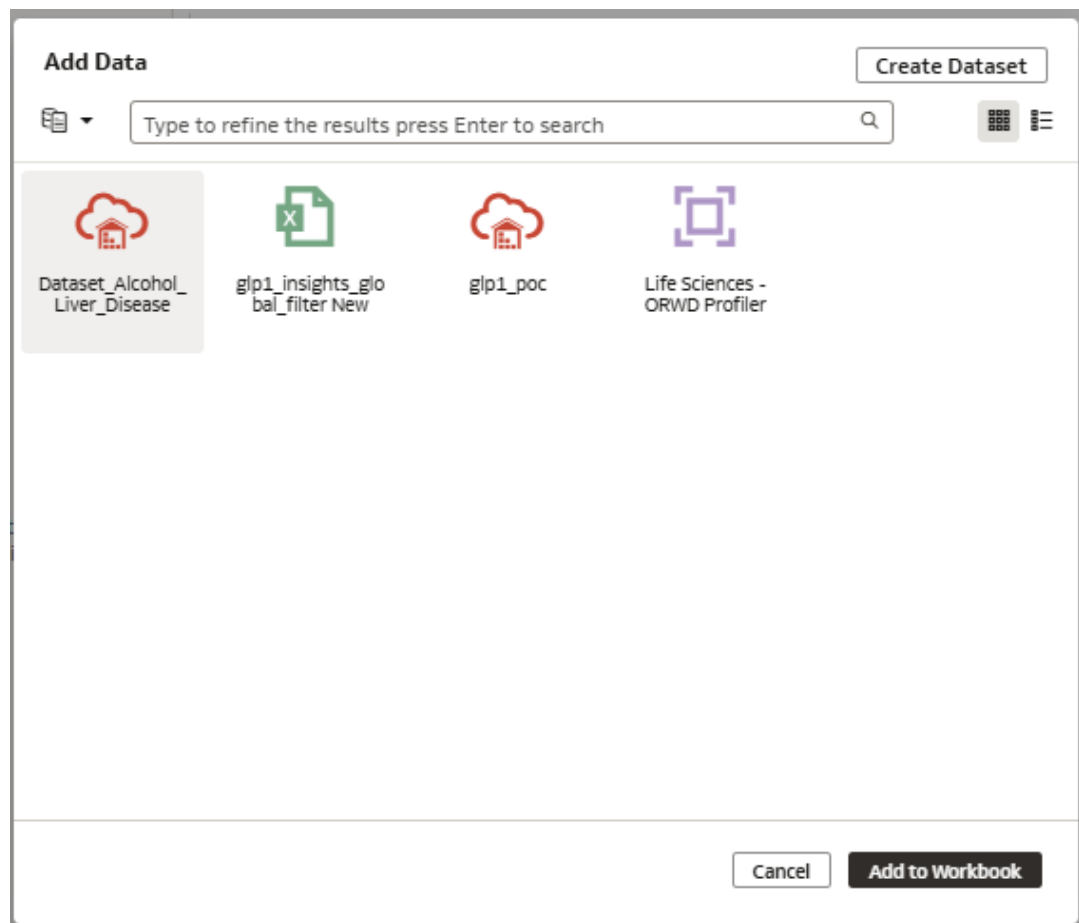
Use workbooks to store and organize analytics content like graphs and charts.

To create a new workbook:

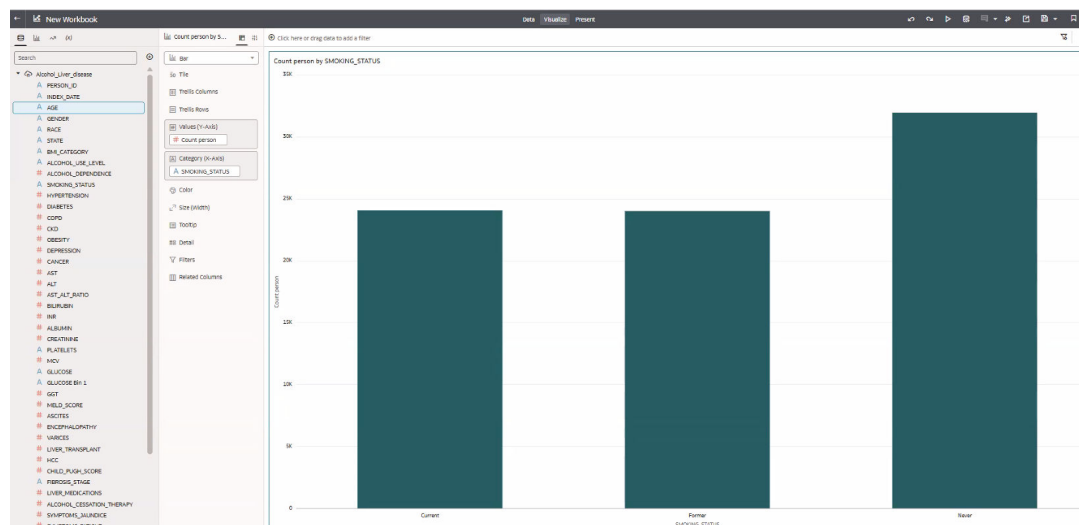
1. From the Oracle Analytics Cloud home page, select **Create**, then select **Workbook**. The Add Data dialog box is displayed.



2. Search for and select a dataset, then select **Add to Workbook**. This dataset is one that was previously created in Real-World Data or from other sources.



3. In the Data pane, locate and drag and drop data columns onto the Visualize canvas to build visualizations.



See [OAC Visualize and Analyze Data](#) for more information.

4

Oracle Health Real-World Data Assets

This section contains information about Oracle Health Real-World Data Assets.

- [Oracle Health Real-World Data Core Model](#)
Describes the main data types and structures in the Oracle Health Real-World Data Core Model and what information they contain.
- [Real-World Data HIPAA Conformance for De-Identification](#)
Oracle uses multiple methods to ensure that data provided for research purposes is deidentified in accordance with HIPAA standards using several methods.
- [Real-World Data Process Workflow and Real-World Data OMOP Common Data Model v5.4](#)
The Observational Medical Outcomes Partnership (OMOP) is an open-source industry standard used to transform data from disparate databases into a Common Data Model (CDM).
- [Real-World Data on Autonomous Data Warehouse](#)
Ensure that usage remains within the boundaries and purposes defined by your contractual agreement. Unauthorized export of regulated or sensitive data is prohibited.

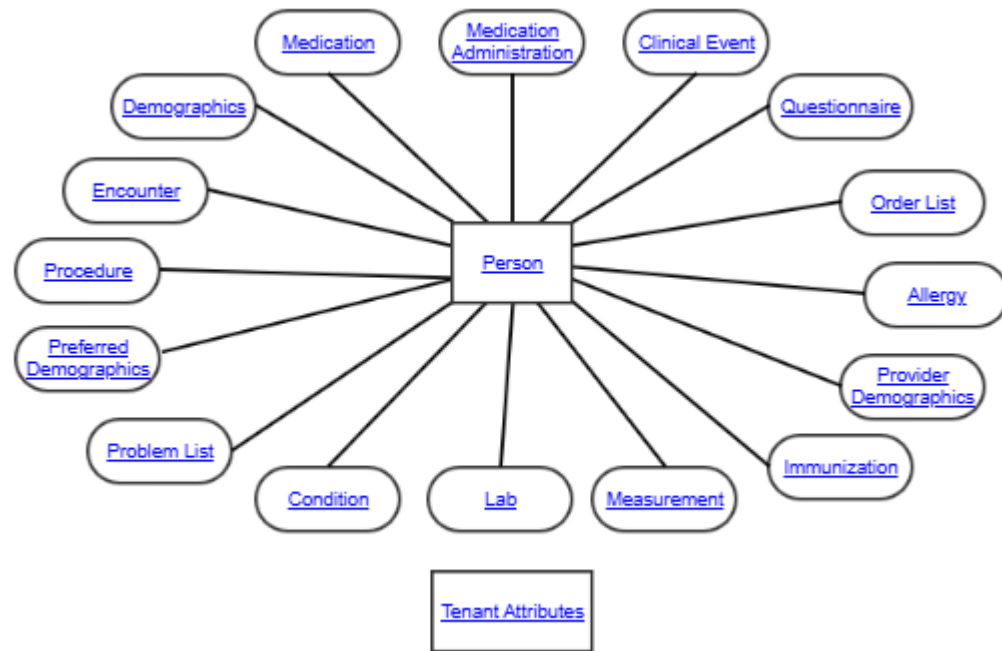
Oracle Health Real-World Data Core Model

Describes the main data types and structures in the Oracle Health Real-World Data Core Model and what information they contain.

Core Model Domains

Real-World Data is deidentified with the Safe Harbor method for Provider and Academic clients, and with the Expert Determination method for life sciences customer projects. See [Real-World Data Expert Determination Data Dictionary](#) for downloadable data dictionaries.

The following diagram displays the structure of data in the Real-World Data Core Model:



Domain	Description
Demographics	Demographics data describes the demographic information for a patient such as gender, race, ethnicity, and marital status.
Encounter	Encounter information represents a planned, current, or past interaction between the patient and one or more health care providers. For example, an encounter can represent an inpatient hospitalization, an ambulatory or outpatient visit, an emergency department visit, a home health visit at the patient's residence, or a virtual encounter through email or teleconference.
Allergy	The person's allergies, which indicate a susceptibility to an adverse reaction upon exposure to a specified allergen. An allergen can be a food substance, an environmental substance, a biologic substance, a medication substance, a pharmaceutical product, or a class of substances (such as penicillins). Negation of an allergen can be holistic (No Known Allergies), per category (No Known Drug Allergies), or per substance (No Known Latex Allergy). Allergies result in an undesirable physiologic reaction to an amount of a substance that would not produce a reaction in most individuals. The allergy list represents a propensity unique to this individual for a reaction upon future exposure to a specified substance. It excludes clinically identical episodes that may be caused by physical agents, for example, cold, sunlight, vibration, exercise activity, or infectious agents.
Clinical Event	Derived from the Result table, clinical event data contains clinical event results for a person, including tobacco use, visit type, observations of disease, conditions, and symptoms.
Condition	Condition information describes the health state of the person including diagnoses and problems. Diagnoses are typically captured in the context of a specific encounter or visit for the purpose of billing. Problems can be anything that has risen to the level of concern or warrants management or tracking through a problem list. Conditions are not limited to diseases and illnesses and can be acute or chronic.

Domain	Description
Immunization	Immunization information contains the vaccines that have been administered to the person and documentation of administrations that were refused, contraindicated, or not given.
Lab	Derived from the Result table, lab data contains laboratory results for a person, including specimen collection and fluid, respiratory, tissue, and other samples.
Measurement	Derived from the Result table, the measurement data contains measurement results for a person, including temperature, heart rate, BMI, and blood pressure.
Medication	Medication information describes all medications prescribed for the person or consumed by the person including inpatient and outpatient pharmacy orders, retail pharmacy prescriptions, over-the-counter medications, and dietary supplements.
Medication Administration	Medication administration events describe a person consuming or being administered a medication, for example swallowing a tablet or a long-running infusion.
Order List	Order list information is a list of the person's orders created on a given date by a specified provider and can include the order reconciliation actions that a provider determines upon discharge or the person's compliance status per order. When the order list is empty, the Order List Model supports documentation of an empty reason for situations where compliance was unable to be obtained due to the person being unconscious or reconciliation is not applicable because the person has no known medications.
Problem List	A list extracted from the Condition model where condition data with this problem type (with the coding system IDs of type.standard.id = 55607006 and type.standard.codingSystemId = 2.16.840.1.113883.6.96) are removed from the condition model and added to the Problem List table.
Procedure	Procedure information describes procedures performed on a person. This can be physical (an operation) or less invasive (counseling or hypnotherapy).
Provider Demographics	Provider demographics data describes the provider ID, CMS classification, CMS specialization, and tenant.
Preferred Demographics	Based on the Demographics table, the Preferred Demographics table contains demographic information in a single record and as a distinct value. This table is intended to resolve conflicts when you have multiple discrepant entries from an array, multiple rows per patient, and an invalid or uninterpretable (non-meaningful) entries. Oracle determines the most appropriate or preferred value for a patient.
Questionnaire	Questionnaire data describes a structured set of questions and their responses about the person. Questionnaires cover the need to communicate data originating from forms, for example, Social History or Outcome and Assessment Information Set (OASIS) assessments.
Tenant Attributes	An additional metadata output table that is provided to help describe the tenants contributing to Real-World Data.

Real-World Data HIPAA Conformance for De-Identification

Oracle uses multiple methods to ensure that data provided for research purposes is deidentified in accordance with HIPAA standards using several methods.

Oracle uses expert determination and safe harbor standards to deidentify Real-World Data as outlined by 45 CFR 164.514.

- **Expert Determination:** A qualified expert uses scientific and statistical methods to minimize identification risk and documents their analysis and findings.
- **Safe Harbor:** Specific direct identifiers such as names, contacts, and exact dates are fully removed or generalized with additional safeguards to prevent identification.

In addition, Oracle:

- Shifts dates by a consistent number of seven-day increments randomly (forward or backward).
- Reduces partial dates to year only.
- Categorizes individuals over 89 years of age as 90+.
- Uses system-assigned numbers instead of extracting direct identifiers like names, contacts, social security numbers, and so on.
- Does not extract location information more specific than the state.

Other sensitive identifiers are not present in the dataset. Data is deidentified before Oracle receives the data. It is encrypted at the provider site, transferred using secure shell (SSH), then stored in a controlled Oracle data center.

See [Understand Real-World Data HIPAA Conformance for Deidentification](#) on Oracle Health Wiki for more information.

Real-World Data Process Workflow and Real-World Data OMOP Common Data Model v5.4

The Observational Medical Outcomes Partnership (OMOP) is an open-source industry standard used to transform data from disparate databases into a Common Data Model (CDM).

OMOP is created by the Observational Health Data Sciences and Informatics (OHDSI) initiative, a collaborative group, and data network. The OHDSI initiative aims to improve health by empowering a community to collaboratively generate evidence that promotes better health decisions and better care.

OMOP is a person-centric relational data model where for each record, the person, and date are required at minimum. All content is mapped or transformed to OMOP standardized vocabularies that correspond to standard healthcare concepts. The original source data are stored to ensure minimal information loss.

The process of converting Oracle EHR Real-World Data (OERWD) to the OMOP CDM v5.4 includes mapping proprietary data to OMOP standard concepts. Custom mappings are applied when proprietary terms do not have corresponding OMOP standard concepts. Following the mapping process, an extraction, transformation, and loading (ETL) code is developed to convert the dataset.

See [OHDSI Specifications](#) for more information.

Real-World Data on Autonomous Data Warehouse

Ensure that usage remains within the boundaries and purposes defined by your contractual agreement. Unauthorized export of regulated or sensitive data is prohibited.

Contracts stipulate penalties for violations, require retention of audit access rights by Oracle, specify limitations on joining Real-World Data with other datasets, and mandate customer cooperation with security audits and policy enforcement.

Dataset	Format	Schema/Table
Oracle Real World Data	In OMOP format or Proprietary format (ED_EXT)	RWD_OMOP_2024_11, RWD_ED_EXT_2024_11, RWD_OMOP_2025_05, RWD_ED_EXT_2025_05
Multum	Multum format	MULTUM_DRUG_REF
Discern Ontology	Discern Ontology format	DISCERN_ONTOLOGY
Reference Dataset	ICD-9, ICD-10, SNOMED, LOINC	OID_REF, EXT_REF_CODE_DESCRIPTION

5

Appendix

This section provides supplementary information about the following topics.

- [Proprietary RWD Extension Data Model](#)
The Real-World Data Extension data model (RWD_ED_EXT_*Schemas) optimizes access to tables that use JSON data types by flattening content and incorporating data cleansing.

Proprietary RWD Extension Data Model

The Real-World Data Extension data model (RWD_ED_EXT_*Schemas) optimizes access to tables that use JSON data types by flattening content and incorporating data cleansing.

Key Features and Structure

Columns that use the JSON data type are not optimized for analytics. The RWD_ED_EXT data model extends the standard data model to improve access to data and performance:

- The EXT model extends the RAW RWD format with additional columns and data cleansing.
- EXT tables are named with the prefix `DW_RWD_EXT_<RWD Object Name>`. Synonyms are also created, so you could refer to the RWD Object Name.
- JSON column data expansion:
 - Original JSON columns are still in the EXT table.
 - JSON elements are expanded as separate string columns as follows:
`EXT_<JSONColumnName>_<JSONElementName>`. For example, for the `CONDITION` table `CONDITIONCODE JSON`, the following columns are included:
 - * `EXT_CONDITIONCODE_CODINGSYSTEMID`
 - * `EXT_CONDITIONCODE_ID`
 - * `EXT_CONDITIONCODE_PRIMARYDISPLAY`
 - * For selected use cases, the coding system name is stored as `EXT_CONDITIONCODE_CODINGSYSTEMNAME`. For codes marked as `**,**` the coding system name is available for use.
 - The following rows do not expand in the extension model:
 - * **Arrays (multiple rows):** Expanding these columns adds more than a million rows and changes the grain of the table.
 - * **Empty:** These rows do not contain values. They are not split into multiple text columns.
- Date Cleansing:
 - Date columns in JSON format use the string data type. The data is adjusted to the date format of `YYYY-MM-DD HH24:MI:SS`. For example:
 - * `2014` → `2014-01-01:00:00` (first day of the year)
 - * `2022-03` → `2022-03-01:00:00` (first day of the month)

- * 2022-W48 → 2022-11-28:00:00 (first day of week 48)
- * 2019-11-27 → 2019-11-27:00:00
- * 2023-03-07T12:05:00Z → 2023-03-07 12:05:00
- * 2023-11-09T18:00:00.000Z → 2023-11-09 18:00:00
- Enriched columns are frequently used attributes that are added as derived columns to the demographics and preferred_demographics tables:
 - AGE
 - * AGE_GROUP_5Y
 - * AGE_GROUP_5Y_SORT_KEY
 - * AGE_GROUP_10Y
 - RACE_ETHNICITY is added to the preferred_demographics table.

Example of the Extended Data Model

For full details of the extended data model, see
RWD_Expert_Determination_Extended_Data_Dictionary.xlsx.

Data Domain	Table Name	Key Columns
ALLERGY	DW_RWD_EXT_ALLERGY	ALLERGENCODE ALLERGYTYPE CATEGORY CRITICALITY STATUS
CLINICAL_EVENT	DW_RWD_EXT_CLINICAL_EVENT	CLINICALEVENTCODE INTERPRETATION SPECIMENTYPE STATUS TYPEDVALUE
CONDITION	DW_RWD_EXT_CONDITION	CLASSIFICATION CONDITIONCODE
DEMOGRAPHICS	DW_RWD_EXT_DEMOGRAPHICS	MARITALSTATUS

Data Domain	Table Name	Key Columns
ENCOUNTER	DW_RWD_EXT_ENCOUNTER	CLASSIFICATION FINANCIALCLASS HOSPITALSERVICE STATUS TYPE
IMMUNIZATION	DW_RWD_EXT_IMMUNIZATION	DRUGCODE IMMUNIZATIONCO MANUFACTURER STATUS
LAB	DW_RWD_EXT_LAB	INTERPRETATION LABCODE SPECIMENTYPE STATUS TYPEDVALUE
MEASUREMENT	DW_RWD_EXT_MEASUREMENT	INTERPRETATION MEASUREMENTCO SPECIMENTYPE STATUS TYPEDVALUE
MEDICATION	DW_RWD_EXT_MEDICATION	DRUGCODE STATUS
MEDICATION_ADMINISTRATION	DW_RWD_EXT_MEDICATION_ADMINISTRATION	EFFECTIVETIME STATUS

Data Domain	Table Name	Key Columns
ORDER_LIST	DW_RWD_EXT_ORDER_LIST	EMPTYREASON
PREFERRED_DEMOGRAPHICS	DW_RWD_EXT_PREFERRED_DEMOGRAPHICS	Not applicable
PROBLEM_LIST	DW_RWD_EXT_PROBLEM_LIST	CONFIRMATIONST PROBLEMLISTCOD STATUS
PROCEDURE	DW_RWD_EXT_PROCEDURE	PROCEDURECOD
PROVIDER_DEMOGRAPHICS	DW_RWD_EXT_PROVIDER_DEMOGRAPHICS	Not applicable
TENANT_ATTRIBUTES	DW_RWD_EXT_TENANT_ATTRIBUTES	Not applicable

Glossary

Index