Oracle® Application Express
SQL Workshop Guide

Release 18.1
E86883-03
May 2018

Oracle Application Express SQL Workshop Guide, Release 18.1

E86883-03

Copyright © 2003, 2018, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Harish Konakondla, Doug Gault, David Peake

Contributors: Terri Jennings, Christina Cho, Michael Hichwa, Joel Kallman, Hilary Farrell, Marc Sewtz, Jason Straub, Drue Swadener, John Synders

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 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 installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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 xi
Documentation Accessibility xi
Related Documents xi
Conventions xii

Changes in This Release

Changes in Oracle Application Express Release 18.1 xiii

1 Getting Started with SQL Workshop

1.1 SQL Workshop Home Page 1-1
1.2 Selecting a Schema 1-2

2 Managing Database Objects with Object Browser

2.1 Accessing Object Browser 2-2
2.2 Object Browser 2-3
2.3 Searching for and Browsing Database Objects 2-4
2.4 Searching For and Selecting Database Objects 2-4
2.5 Hiding the Object Selection Pane 2-5
2.6 Creating Database Objects 2-6
2.7 Managing Tables 2-6
2.7.1 Creating a Table 2-7
2.7.2 Browsing a Table 2-9
2.7.3 Reports Available for Tables 2-9
2.7.4 Editing a Table 2-11
2.7.5 Dropping a Table 2-12
2.8 Managing Views 2-12
2.8.1 Creating a View 2-13
2.8.2 Browsing a View 2-14
2.8.3 Reports for Views 2-14

2.8.4 Editing a View Manually 2-15
2.8.5 Using Find and Replace 2-15
2.8.6 Downloading a View 2-16
2.8.7 Compiling a View 2-16
2.8.8 Dropping a View 2-16
2.9 Managing Indexes 2-17
2.9.1 Creating an Index 2-17
2.9.2 Browsing an Index 2-18
2.9.3 Reports for Indexes 2-18
2.9.4 Dropping an Index 2-18
2.10 Managing Sequences 2-19
2.10.1 Creating a Sequence 2-19
2.10.2 Browsing a Sequence 2-20
2.10.3 Reports for Sequences 2-20
2.10.4 Dropping a Sequence 2-20
2.10.5 Altering a Sequence 2-21
2.11 Managing Types 2-21
2.11.1 Creating a Type 2-22
2.11.2 Browsing a Type 2-22
2.11.3 Reports for Types 2-22
2.11.4 Dropping a Type 2-23
2.12 Managing Packages 2-23
2.12.1 Creating a Package 2-24
2.12.2 Viewing a Package 2-25
2.12.3 Reports for Packages 2-25
2.12.4 Editing a Package Manually 2-26
2.12.5 Compiling a Package 2-26
2.12.6 Downloading a Package 2-27
2.12.7 Dropping a Package While Viewing Specification 2-27
2.12.8 Dropping a Package While Viewing the Body 2-27
2.13 Managing Procedures 2-28
2.13.1 Creating a Procedure 2-28
2.13.2 Browsing a Procedure 2-29
2.13.3 Reports for Procedures 2-29
2.13.4 Editing a Procedure 2-30
2.13.4.1 Editing a Procedure Manually 2-30
2.13.5 Compiling a Procedure 2-31
2.13.6 Downloading a Procedure 2-31
2.13.7 Dropping a Procedure 2-31
2.14 Managing Functions 2-31
2.14.1 Creating a Function 2-32

2.14.2 Browsing a Function 2-33
2.14.3 Reports for Functions 2-33
2.14.4 Editing a Function Manually 2-34
2.14.5 Compiling a Function 2-34
2.14.6 Downloading a Function 2-35
2.14.7 Dropping a Function 2-35
2.15 Managing Triggers 2-35
2.15.1 Creating Triggers 2-36
2.15.2 Browsing a Trigger 2-36
2.15.3 Reports for Triggers 2-37
2.15.4 Editing a Trigger Manually 2-37
2.15.5 Compiling a Trigger 2-38
2.15.6 Downloading a Trigger 2-38
2.15.7 Dropping a Trigger 2-39
2.15.8 Disabling and Enabling a Trigger 2-39
2.16 Managing Database Links 2-39
2.16.1 Creating a Database Link 2-40
2.16.2 Browsing a Database Link 2-40
2.16.3 Reports for Database Links 2-41
2.16.4 Dropping a Database Link 2-41
2.17 Managing Materialized Views 2-41
2.17.1 Creating a Materialized View 2-42
2.17.2 Browsing a Materialized View 2-43
2.17.3 Reports for Materialized Views 2-43
2.17.4 Dropping a Materialized View 2-44
2.18 Managing Synonyms 2-44
2.18.1 Creating Synonyms 2-44
2.18.2 Viewing a Synonym 2-45
2.18.3 Dropping a Synonym 2-45

3 Using SQL Commands

3.1 About SQL Commands 3-2
3.2 Accessing SQL Commands 3-2
3.3 About the SQL Commands Home Page 3-3
3.4 Using the Command Editor 3-5
3.4.1 Running a SQL Command 3-6
3.4.2 About Transactions in SQL Commands 3-6
3.4.3 Disabling Transactional SQL Commands 3-6
3.4.4 Enabling Transactional SQL Commands 3-7
3.4.5 About Unsupported SQL*Plus Commands 3-7

3.4.6 Terminating a Command 3-7
3.4.7 Using Bind Variables 3-7
3.4.8 Using the Find Tables Button 3-7
3.5 Saving an SQL Command 3-8
3.6 Copying a Command 3-8
3.7 Viewing Results 3-9
3.7.1 Accessing the Results Pane 3-9
3.7.2 About the Results Pane 3-9
3.8 Using Explain Plan 3-10
3.8.1 Viewing an Explain Plan 3-10
3.9 Using Saved Commands 3-10
3.9.1 Accessing Saved Commands 3-11
3.9.2 About the Saved SQL Pane 3-11
3.10 Using SQL Command History 3-11
3.10.1 Accessing a Command from Command History 3-12
3.10.2 About the History Pane 3-12

4 Using Oracle Application Express Utilities

4.1 Using Data Workshop 4-2
4.1.1 About the Data Load and Unload Wizards 4-2
4.1.2 About Importing, Exporting, Loading, and Unloading Data 4-3
4.1.3 Import/Export/Load/Unload Options 4-3
4.1.4 Accessing the Data Load/Unload Page 4-5
4.1.5 Loading Data 4-5
4.1.5.1 About Loading Data 4-5
4.1.5.2 Loading a Text File or Spreadsheet Data 4-6
4.1.5.3 Loading an XML Document 4-8
4.1.6 Unloading Data 4-8
4.1.6.1 Unloading a Text File 4-9
4.1.6.2 Unloading to an XML Document 4-10
4.1.7 Using the Repository 4-10
4.2 Using Query Builder 4-10
4.2.1 Query Builder Home Page 4-12
4.2.2 Accessing Query Builder 4-12
4.2.3 Understanding the Query Building Process 4-13
4.2.4 Using the Object Selection Pane 4-13
4.2.4.1 Searching and Filtering Objects 4-13
4.2.4.2 Hiding and Restoring the Object Selection Pane 4-14
4.2.4.3 Selecting Objects 4-14
4.2.5 Using the Design Pane 4-14

4.2.5.1 About Supported Column Types 4-15
4.2.5.2 Adding an Object to the Design Pane 4-15
4.2.5.3 Resizing the Design and Results Panes 4-16
4.2.5.4 Removing or Hiding Objects in the Design Pane 4-16
4.2.6 Specifying Query Conditions 4-17
4.2.7 Creating Relationships Between Objects 4-18
4.2.7.1 About Join Conditions 4-18
4.2.7.2 Joining Objects Manually 4-19
4.2.7.3 Joining Objects Automatically 4-19
4.2.8 Working with Saved Queries 4-20
4.2.8.1 Saving a Query 4-21
4.2.8.2 Editing a Saved Query 4-21
4.2.8.3 Deleting a Saved Query 4-21
4.2.9 Viewing Generated SQL 4-22
4.2.10 Viewing Query Results 4-22
4.3 Using Quick SQL 4-22
4.3.1 Getting Started with Quick SQL 4-23
4.3.1.1 About Quick SQL 4-23
4.3.1.2 About Quick SQL Shorthand Syntax 4-23
4.3.1.3 Accessing Quick SQL 4-24
4.3.2 Using Quick SQL Help 4-24
4.3.2.1 Accessing Quick SQL Help 4-25
4.3.2.2 About Quick SQL Help 4-25
4.3.3 Generating Quick SQL with SQL Shorthand 4-26
4.3.4 Saving Scripts 4-27
4.3.4.1 Saving a SQL Script 4-27
4.3.4.2 Downloading SQL Script 4-28
4.3.5 Creating an Application from a Generated SQL Script 4-29
4.3.6 Using Quick SQL Samples 4-31
4.3.6.1 About Quick SQL Samples 4-31
4.3.6.2 Accessing Quick SQL Samples 4-31
4.3.7 Configuring Settings 4-32
4.3.7.1 About Settings 4-32
4.3.7.2 Configuring Settings Declaratively 4-32
4.3.7.3 Configuring Settings Inline 4-34
4.3.8 About Database Terminology 4-34
4.3.9 Recommendations When Using Quick SQL 4-37
4.3.10 About Supported Data Types 4-37
4.4 Using Sample Datasets 4-38
4.4.1 About Sample Datasets 4-38
4.4.2 Accessing Sample Datasets 4-38

4.4.3 Installing a Sample Dataset 4-39
4.4.4 Removing a Sample Dataset 4-40
4.4.5 Refreshing a Sample Dataset 4-41
4.4.6 Changing the Schema for a Sample Dataset 4-41
4.5 Generating DDL 4-42
4.6 Managing User Interface Defaults 4-42
4.6.1 About User Interface Defaults 4-43
4.6.2 Creating User Interface Defaults for a Table 4-43
4.6.3 Modifying Table User Interface Defaults 4-44
4.6.4 Creating User Interface Attributes 4-45
4.6.5 Modifying User Interface Attributes 4-45
4.6.6 Exporting and Importing User Interface Defaults 4-46
4.7 Comparing Schemas 4-46
4.8 Managing Methods on Tables 4-47
4.9 Using the Recycle Bin 4-47
4.9.1 Managing Objects in the Recycle Bin 4-48
4.9.2 Emptying the Recycle Bin Without Viewing the Objects 4-48
4.10 Viewing Object Reports 4-48
4.10.1 Viewing Table Reports 4-49
4.10.2 Viewing PL/SQL Reports 4-49
4.10.2.1 Viewing Package Input and Output Parameters 4-50
4.10.2.2 Viewing the Number of Lines of Code for an Object 4-50
4.10.2.3 Searching PL/SQL Source Code 4-50
4.10.3 Viewing Exception Reports 4-51
4.10.4 Viewing Security Reports 4-51
4.10.5 Viewing All Object Reports 4-52
4.11 Viewing Database Details 4-53
4.12 Monitoring the Database 4-53
4.12.1 Accessing Session Page Reports 4-54
4.12.1.1 Accessing Reports on the Session Page 4-54
4.12.1.2 Creating Sessions Reports 4-55
4.12.1.3 Controlling the Number of Rows for Locks Reports 4-55
4.12.1.4 Creating Waits Reports 4-55
4.12.1.5 Creating I/O Reports 4-56
4.12.1.6 Creating SQL Reports 4-56
4.12.1.7 Creating Open Cursors Report 4-56
4.12.2 System Statistics 4-56
4.12.3 Viewing System Statistics 4-57
4.12.4 About Top SQL Page 4-57
4.12.5 Viewing Top SQL Page 4-58
4.12.6 About Long Operations Page 4-58

4.12.7 Viewing the Long Operations Page 4-58

5 Using SQL Scripts

5.1 About SQL Scripts 5-1
5.2 Accessing SQL Scripts 5-2
5.3 About the SQL Scripts Page 5-3
5.4 About the Tasks List 5-4
5.5 Creating a SQL Script 5-4
5.5.1 Creating a SQL Script in the Script Editor 5-4
5.5.2 Uploading a SQL Script 5-5
5.6 Using the Script Editor 5-5
5.6.1 Editing an Existing Script 5-5
5.6.2 Script Editor Controls 5-6
5.7 Deleting a SQL Script 5-7
5.7.1 Deleting Scripts from the SQL Scripts Page 5-7
5.7.2 Deleting a Script in the Script Editor 5-7
5.8 Copying a SQL Script 5-8
5.9 Executing a SQL Script 5-8
5.9.1 Executing a SQL Script in the Script Editor 5-8
5.9.2 Executing a SQL Script from the SQL Scripts Page 5-9
5.10 Viewing SQL Script Results 5-9
5.11 Exporting and Importing SQL Scripts 5-10
5.11.1 Copying Scripts to an Export Script 5-10
5.11.2 Importing Scripts 5-11
5.12 Viewing Script and Result Quotas 5-12

6 Enabling Data Exchange with RESTful Services

6.1 What Is REST? 6-2
6.2 About Oracle RESTful Services 6-3
6.3 RESTful Service Requirements 6-4
6.4 How to Access RESTful Services 6-6
6.4.1 Accessing RESTful Services 6-6
6.4.2 Registering Schema with ORDS 6-7
6.4.3 About ORDS RESTful Services Dashboard 6-8
6.5 How to Create the ORDS Based RESTful Services Module Example 6-10
6.5.1 About the Example RESTful Service Module (oracle.example.hr) 6-11
6.5.2 Returning the Result Set Based on a PL/SQL Block (empsecformat/:empname) 6-13
6.5.3 Returning the Result Set in JSON Format with Pagination Set (employees/) 6-14

6.5.4 Retrieving Data Based on a Parameter (employees/:id) 6-15
6.5.5 Returning the Result Set in CSV Format (empinfo/) 6-16
6.5.6 Returning the Result Set as a Feed (employeesfeed/) 6-17
6.6 Managing RESTful Service Modules 6-18
6.6.1 Installing a Sample RESTful Service Module with Schema ORDS not Enabled 6-19
6.6.2 Installing a Sample RESTful Service Module with Schema ORDS Enabled 6-20
6.6.3 Creating an ORDS RESTful Service Module 6-21
6.6.4 Editing an ORDS RESTful Service Module 6-21
6.6.5 Deleting an ORDS RESTful Service Module 6-22
6.6.6 Managing an ORDS RESTful Service Privileges 6-22
6.6.7 Assigning Privileges to an ORDS RESTful Service Modules 6-23
6.6.8 Creating an ORDS RESTful Service Privilege 6-23
6.6.9 Editing an RESTful Service Privilege 6-24
6.6.10 Deleting an ORDS RESTful Service Privilege 6-24
6.6.11 Exporting an ORDS RESTful Service Module 6-25
6.6.12 Importing an ORDS RESTful Service Module 6-25
6.6.13 Generating a Swagger Document 6-26
6.7 Managing Resource Templates 6-26
6.7.1 Adding a Resource Template 6-27
6.7.2 Editing a Resource Template 6-27
6.7.3 Deleting a Resource Template 6-28
6.8 Managing Resource Handlers 6-28
6.8.1 Adding a Resource Handler 6-28
6.8.2 Editing a Resource Handler 6-29
6.8.3 Deleting a Resource Handler 6-29

Index

Preface

Oracle Application Express SQL Workshop Guide describes how to use Oracle Application Express SQL Workshop and utilities to load and unload data from an Oracle database, generate DDL, view object reports, and restore dropped database objects.

- Audience (page xi)
- Documentation Accessibility (page xi)
- Related Documents (page xi)
- Conventions (page xii)

Audience

Oracle Application Express SQL Workshop Guide is intended for application developers who are building database-centric Web applications using Oracle Application Express. The guide describes how to use Oracle Application Express SQL Workshop and utilities to load and unload data from an Oracle database, generate DDL, view object reports, and restore dropped database objects.

To use this guide, you must have a general understanding of relational database concepts and the operating system environment under which Oracle Application Express is running.

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.

Related Documents

For more information, see these Oracle resources:

- *Oracle Application Express Release Notes*
- *Oracle Application Express Installation Guide*

- *Oracle Application Express App Builder User's Guide*
- *Oracle Application Express API Reference*
- *Oracle Application Express Administration Guide*
- *Oracle Application Express Application Migration Guide*
- *Oracle Application Express End User's Guide*
- *Oracle Database Concepts*
- *Oracle Database Development Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database SQL Language Reference*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database PL/SQL Language Reference*

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.
italic	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.

Changes in This Release

This preface contains:

- Changes in Oracle Application Express Release 18.1 (page xiii)

Changes in Oracle Application Express Release 18.1

The following are changes in *Oracle Application Express SQL Workshop Guide* for Oracle Application Express release 18.1.

- New Features (page xiii)
- Deprecated and Desupported Features (page xiv)
- Other Changes (page xiv)

New Features

The following feature is new in this release:

- Quick SQL

  Quick SQL is used to rapidly design and prototype data models using a markdown-like shorthand syntax. It is simply a quick way to develop a script for simple tables and views. Once the SQL is generated it can be tweaked and expanded upon. Quick SQL provides a quick way to generate the SQL required to create a relational data model from an indented text document.

  See "Using Quick SQL (page 4-22)"

- Sample Datasets

  Sample datasets enable you to easily build sample applications. Each dataset includes sufficient data to build pages with various different components. Install, refresh, or replace sample datasets within one of the schemas associated with your workspace. The database objects can also be easily removed. Each dataset includes various database objects and sample data. The sample data is available in one or more languages.

  See "Using Sample Datasets (page 4-38)"

- New REST workshop

  Introduction to the concepts of RESTful services and describes how to create, edit and delete RESTful Services using Application Express.

  RESTful Services enable the creation of a set of standards-based API's on top of database objects available for a workspace. Without connecting directly to the underlying data source these API's allow external systems to interact securely with the data by querying, inserting, updating or deleting.

  See "Enabling Data Exchange with RESTful Services (page 6-1)"

Deprecated and Desupported Features

See "Deprecated Features" and "Desupported Features" in *Oracle Application Express Release Notes* .

Other Changes

The following are additional changes in the release:

- All content has been updated to reflect new functionality.

1

Getting Started with SQL Workshop

SQL Workshop provides tools that enable you to view and manage database objects.

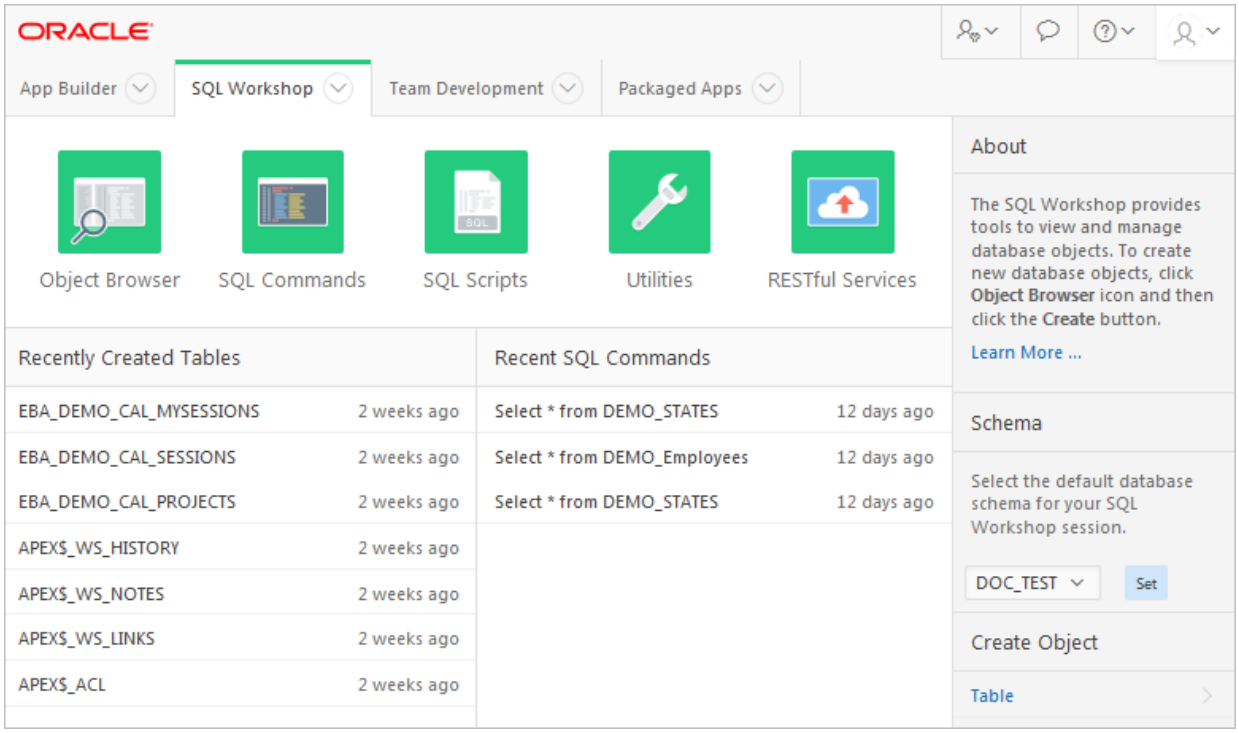
Object Browser enables you to use a tree control to view object properties and create new objects. The SQL Commands tool enables you to enter ad-hoc SQL. Query Builder enables you to create join queries using drag and drop. SQL Scripts enables you to store and run scripts. The Data Workshop enables you to load and unload text, DDL, and spreadsheet data.

- SQL Workshop Home Page (page 1-1)
  Learn about the SQL Workshop home page.
- Selecting a Schema (page 1-2)
  Select the default database schema for your SQL Workshop session.

1.1 SQL Workshop Home Page

Learn about the SQL Workshop home page.

You access SQL Workshop by clicking the **SQL Workshop** icon on the Workspace home page. The SQL Workshop home page appears.



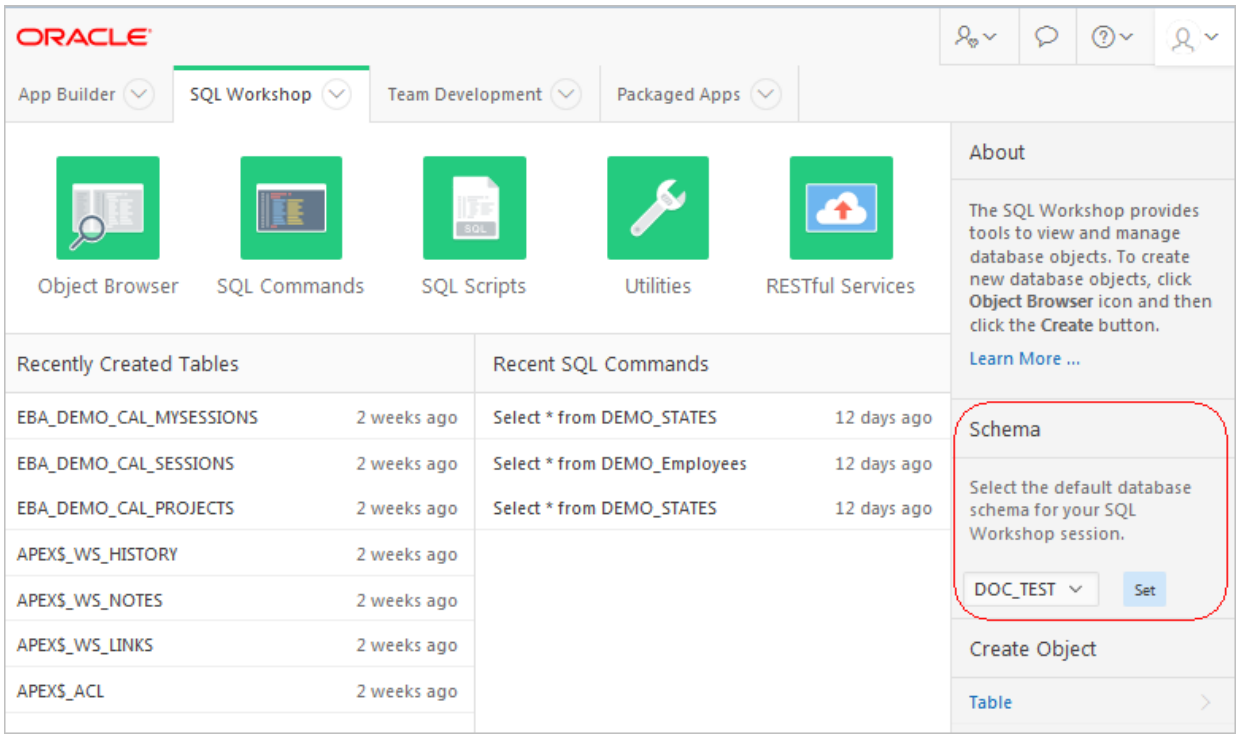
SQL Workshop home page features the following icons:

- Object Browser.
- SQL Commands.
- SQL Scripts.
- Utilities.
- RESTful Services.

1.2 Selecting a Schema

Select the default database schema for your SQL Workshop session.

A schema is a logical container for database objects. Use the Schema list on the right side of the SQL Workshop home page to select the default database schema for your SQL Workshop session. Only schemas assigned to your workspace display in the Schema list.



2

Managing Database Objects with Object Browser

Object Browser enables developers to browse, create, and edit objects in a database.

Oracle Application Express has many views defined to help you work with the underlying table structures. The views combine multiple base tables and use meaningful column names to minimize complexity and to be more intuitive to use.

- Accessing Object Browser (page 2-2)
  Learn about how to access Object Browser.
- Object Browser (page 2-3)
  The Object Browser page is divided into two panes - Object Selection pane and Detail pane.
- Searching for and Browsing Database Objects (page 2-4)
  Filter the view by selecting an object type or entering a case insensitive search term.
- Searching For and Selecting Database Objects (page 2-4)
  Search for and select database objects in Object Browser by selecting an object type from the Object list.
- Hiding the Object Selection Pane (page 2-5)
  You can hide the Object Selection pane by selecting the Collapse control which displays on the right side of the Object Selection pane.
- Creating Database Objects (page 2-6)
  You can create database objects using the Create Database Object Wizard.
- Managing Tables (page 2-6)
  You can create, browse, edit, or drop tables as well as view different table reports.
- Managing Views (page 2-12)
  A view is a logical representation of another table or combination of tables. You can create, browse, edit, download, compile, drop a view and view reports.
- Managing Indexes (page 2-17)
  An index is an optional structure associated with tables and clusters. You can create, browse, drop an index and view reports.
- Managing Sequences (page 2-19)
  A sequence generates a serial list of unique numbers for numeric columns of a database table. You can create, browse, drop, alter a sequence and view a report.
- Managing Types (page 2-21)
  A type is a user-specified object or collection definition. You can create, browse, drop a type and view a report.
- Managing Packages (page 2-23)
  A package is a database object that groups logically related PL/SQL types, items, functions and procedures. You can create, view, edit, compile, drop, download a package, and view a report.

- Managing Procedures (page 2-28)
  A procedure is a subprogram that performs a specific action. You can use Object Browser to view, create, edit, download, and drop procedures.
- Managing Functions (page 2-31)
  A function is a subprogram that can take parameters and return a single value. You can use Object Browser to create, browse, edit, compile, download, drop functions and view report.
- Managing Triggers (page 2-35)
  A database trigger is a stored subprogram associated with a database table, view, or event. You can create, browse, edit, compile, download, drop, disable or enable a trigger and view report.
- Managing Database Links (page 2-39)
  A database link is a schema object in one database that enables you to access objects in another database. You can create, browse, drop a database link and view report.
- Managing Materialized Views (page 2-41)
  A materialized view provides indirect access to table data by storing the results of a query in a separate schema object. You can create, browse, drop a materialized view and view report.
- Managing Synonyms (page 2-44)
  A synonym is an alias for a schema object. You can create, view and drop a synonym.

See Also:

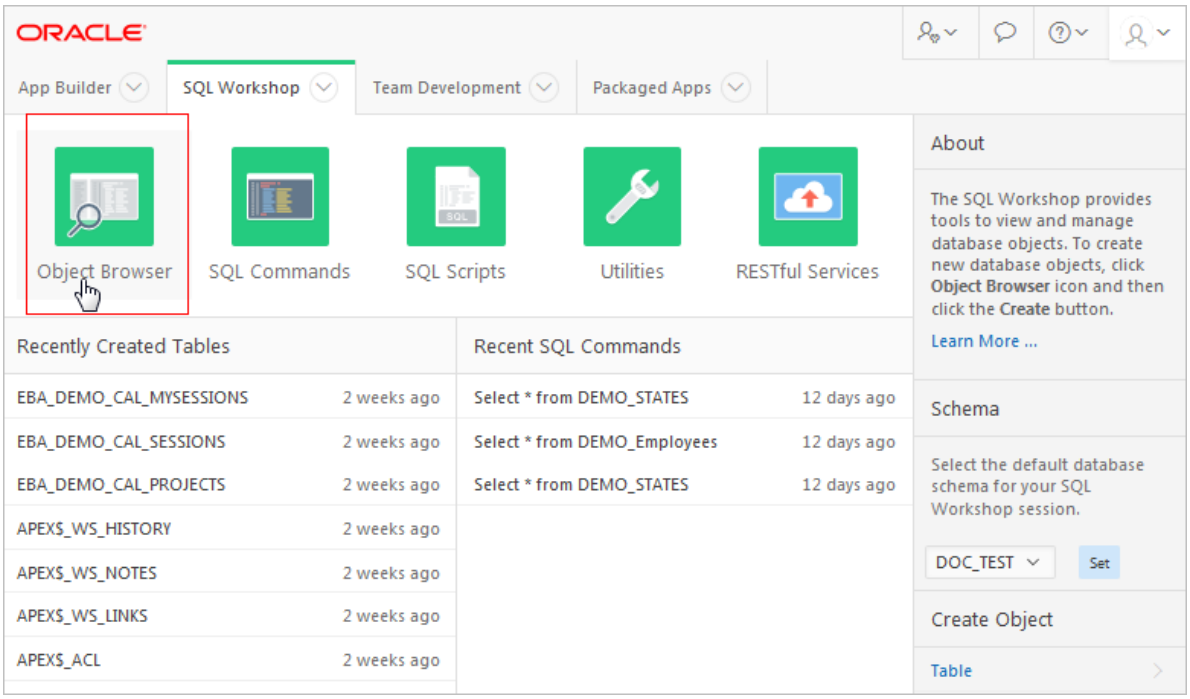
- *Oracle Database SQL Language Reference*
- *Oracle Application Express API Reference* for details of the views defined

2.1 Accessing Object Browser

Learn about how to access Object Browser.

To access Object Browser:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Object Browser**.



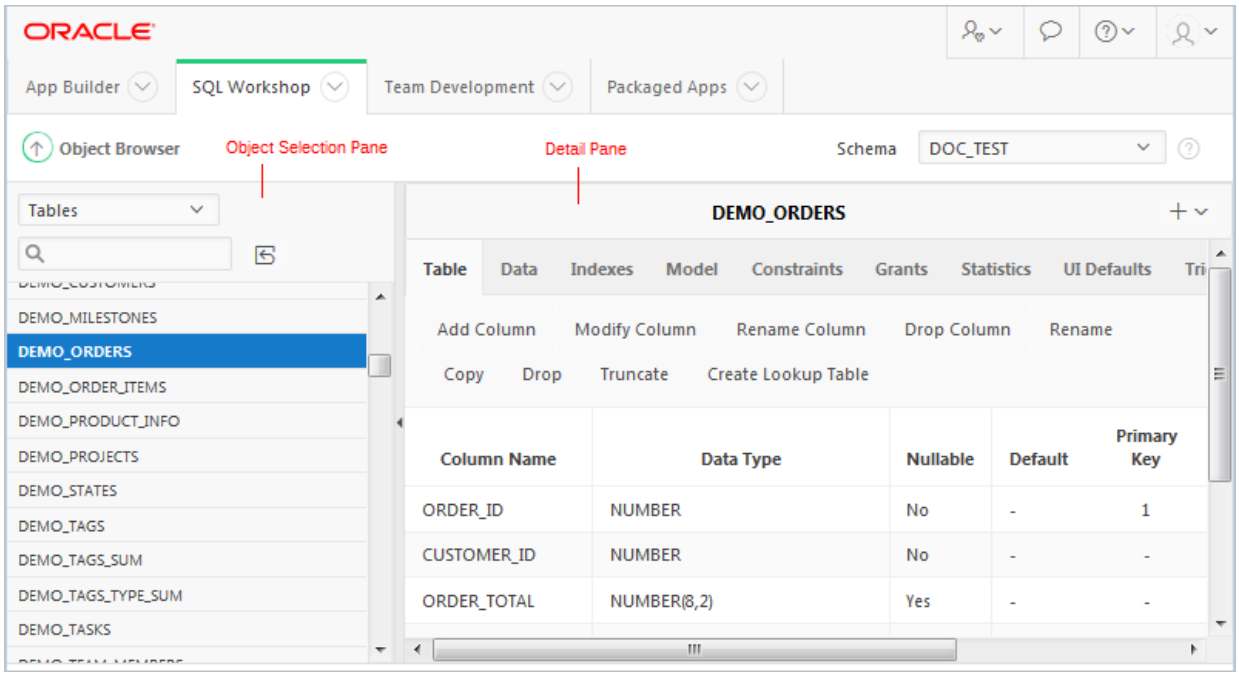
> **Note:**
>
> If the instance administrator has disabled RESTful Services for this Application Express instance, RESTful Services are not available for this instance and the RESTful Services icon does not display.

> **See Also:**
>
> "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*

2.2 Object Browser

The Object Browser page is divided into two panes - Object Selection pane and Detail pane.

- **Object Selection pane** displays on the left side of the Object Browser page and lists database objects of a selected type within the current schema. You can further narrow the results by filtering on the object name.
- **Detail pane** displays to the right of the page and displays detailed information about the selected object. To view object details, select an object in the Object Selection pane. Click the tabs at the top of the Detail pane to view additional details about the current object. To edit an object, click the appropriate button.



2.3 Searching for and Browsing Database Objects

Filter the view by selecting an object type or entering a case insensitive search term.

The Object Selection pane displays on the left side of the Object Browser page and lists database objects by type with the current schema. You can filter the view by selecting an object type or entering a case insensitive search term.

2.4 Searching For and Selecting Database Objects

Search for and select database objects in Object Browser by selecting an object type from the Object list.

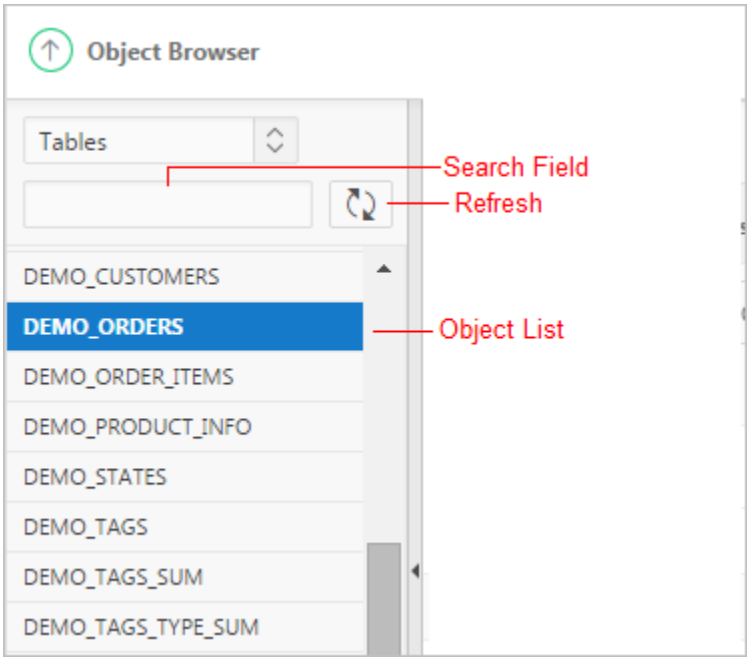
To search for a database object in the Object Selection pane:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. Select an object type from the Object list.

   The list of objects that appears depends upon the available objects in the current schema. Note that any object having a red bar adjacent to it is invalid.



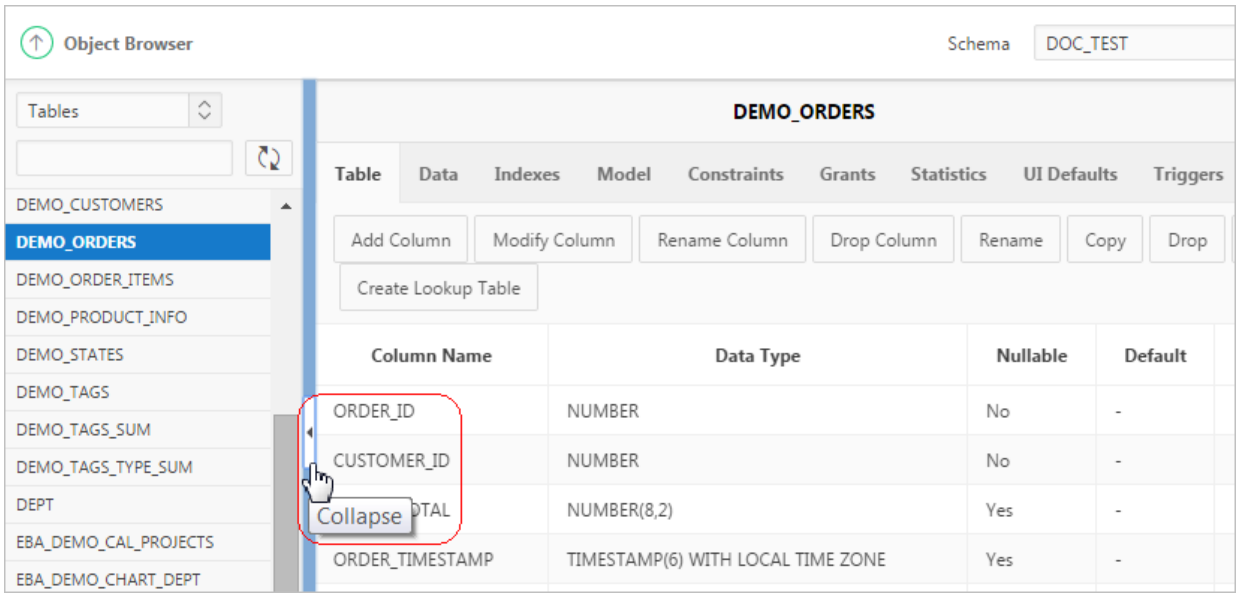
3. To search for an object name, enter a case insensitive search term in the Search field.
4. To view all objects, leave the search field blank.

Once you locate the database object you want to view, simply select it. The selected object displays in the Detail pane. If no object is selected, the Detail pane is blank.

2.5 Hiding the Object Selection Pane

You can hide the Object Selection pane by selecting the Collapse control which displays on the right side of the Object Selection pane.

If the Object Selection pane appears, selecting this control hides it. Similarly, if the Object Selection pane is hidden, selecting this control causes the pane to reappear.



2.6 Creating Database Objects

You can create database objects using the Create Database Object Wizard.

Once you select an object, a set of tabs and buttons appears at the top of the Detail pane. Use the tabs to view different aspects of the current items (for example, the indexes of the table). Use the buttons to modify the current object.

To create an object:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon which displays as a plus (+) sign in the upper right corner of the Detail pane.
3. From the list of object types, select the type of object you want to create.
4. Follow the on-screen instructions.

2.7 Managing Tables

You can create, browse, edit, or drop tables as well as view different table reports.

A table is a unit of data storage in an Oracle database, containing rows and columns. When you view a table in Object Browser, a table description appears that describes each column in the table.

- Creating a Table (page 2-7)
  Create a table using Object Browser.
- Browsing a Table (page 2-9)
  Learn how to view different reports about a table.
- Reports Available for Tables (page 2-9)
  Learn about alternative views available when viewing a table in Object Browser.
- Editing a Table (page 2-11)
  Click the appropriate buttons on a table in Object Browser to edit a table.
- Dropping a Table (page 2-12)
  Drop a table using Object Browser.

See Also:

- *Oracle Database Administrator's Guide* for information on managing tables
- *Oracle Database Concepts* for conceptual information on tables types
- *Oracle Database SQL Language Reference* for the syntax required to create and alter tables

2.7.1 Creating a Table

Create a table using Object Browser.

To create a table:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Table**.
4. Enter a table name.

   Table names must conform to Oracle naming conventions.
5. To have the final table name match the case entered in the Table Name field, click **Preserve Case**.
6. Enter column details:
   a. Column Name - Enter the column name.
   b. Type - Select the column type.
   c. Precision - The precision is the number of digits before the decimal point. Not all column types have this setting. Precision must be a positive integer. Only `NUMBER`, `FLOAT`, `INTERVAL YEAR TO MONTH`, and `INTERVAL DAY TO SECOND` have this setting. For example, if a column of type `NUMBER` has precision of 4, the column value can be 0 through 9999.
   d. Scale - The function of the scale setting depends on the column type. For `NUMBER` types, scale is the number of digits after the decimal point. For `VARCHAR2` and `CHAR` types, scale is the number of stored characters. For `TIMESTAMP` types, scale is the fractional seconds precision and must be a number between 0 and 9. Scale must be a positive number. For example, if a column type `TIMESTAMP` has a scale of 3, seconds are between .000 seconds and .999 seconds.
   e. Not Null - To specify a column must always have a value, select the **Not Null** check box.
   f. If running Oracle Database 12*c* - Select an identity option for NUMBER and FLOAT columns. Options include:
      - **Always** - The Oracle Database always uses the sequence generator to assign a value to the column. If you attempt to explicitly assign a value to the column using INSERT or UPDATE, then an error is returned.
      - **Default** - The Oracle Database uses the sequence generator to assign a value to the column by default, but you can also explicitly assign a specified value to the column.
      - **Default on Null** - The Oracle Database uses the sequence generator to assign a value to the column when a subsequent INSERT statement attempts to assign a value that evaluates to NULL.
   g. Move - Click the **Up** and **Down** arrows in the Move column to change the order of previously entered columns.
   h. Add Column - Click Add Column to add additional columns.
   i. Click **Next**.

Next, define the primary key for this table (optional). A primary key is a single field or combination of fields that uniquely identifies a record.

7. For Primary Key, select the appropriate option and click **Next**:
   a. Select an option:
      - **No Primary Key** - No primary key is created.
      - **Populate from a new sequence** - Creates a primary key and creates a trigger and a sequence. The sequence is used in the trigger to populate the selected primary key column. The primary key can only be a single column.
      - **Populated from an existing sequence** - Creates a primary key and creates a trigger. The selected sequence is used in the trigger to populate the selected primary key column. The primary key can only be a single column.
      - **Not populated** - Defines a primary key but does not have the value automatically populated with a sequence within a trigger. You can also select this option to define a composite primary key (that is, a primary key made up of multiple columns).
      - If running Oracle Database 12*c* - **Populate By Identity Column** - Creates a primary key and a sequence. The sequence is used to populate the selected primary key column.
   b. Click **Next**.

   Next, add foreign keys (optional). A foreign key establishes a relationship between a column (or columns) in one table and a primary or unique key in another table.

8. Add Foreign Key:
   a. Name - Enter a name of the foreign key constraint that you are defining.
   b. Select the appropriate option:
      - **Disallow Delete** - Blocks the delete of rows from the referenced table when there are dependent rows in this table.
      - **Cascade Delete** - Deletes the dependent rows from this table when the corresponding parent table row is deleted.
      - **Set to Null on Delete** - Sets the foreign key column values in this table to null when the corresponding parent table row is deleted.
   c. Select Key Column(s) - Select the columns that are part of the foreign key, then click the **Add** icon to move them to Key Column(s).
   d. References Table - Select the table which is referenced by this foreign key. Then, select the columns referenced by this foreign key. Once selected, click the **Add** icon to move the selected columns to Referenced Column(s).
   e. Click **Add**.
   f. Click **Next**.

   Next, add a constraint (optional). You can create multiple constraints, but you must add each constraint separately.

9. To add a constraint:
   a. Specify the type of constraint (Check or Unique).

A **check constraint** is a validation check on one or more columns within the table. No records can be inserted or updated in a table which violates an enabled check constraint. A **unique constraint** designates a column or a combination of columns as a unique key. To satisfy a unique constraint, no two rows in the table can have the same values for the specified columns.A **check constraint** is a validation check on one or more columns within the table. No records can be inserted or updated in a table which violates an enabled check constraint. A **unique constraint** designates a column or a combination of columns as a unique key. To satisfy a unique constraint, no two rows in the table can have the same values for the specified columns.

b. Enter the constraint in the field provided. For unique constraints, select the column(s) that are to be unique. For check constraints, enter the expression that should be checked such as `flag in ('Y','N')`.

c. Click **Add**.

10. Click **Next**.

    A confirmation page appears.

11. Click **Create**.

2.7.2 Browsing a Table

Learn how to view different reports about a table.

When you view a table in Object Browser, the table description appears. While viewing this description, you can add a column, modify a column, rename a column, drop a column, rename the table, copy the table, drop the table, truncate the table, or create a lookup table based upon a column in the current table. Additionally, you have access to other reports that offer related information including the table data, indexes, data model, constraints, grants, statistics, user interface defaults, triggers, dependencies, and SQL to create the selected table.

To view a table description:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, ensure **Tables** is selected.
3. From the Object Selection pane, select a table.

   The table description appears.

4. Click the tabs at the top of the page to view different reports about the table.

2.7.3 Reports Available for Tables

Learn about alternative views available when viewing a table in Object Browser.

Table 2-1 (page 2-10) describes all available reports for tables.

Table 2-1 Available Reports for Tables

View	Description
Table	Displays details of the first 57 columns including the column name, data type, nullable status, default value, and primary key. While viewing table details you can add, modify, delete, or rename a column. Additionally, you can drop, rename, copy, or truncate the table only if the referencing table has no records and create a lookup table. To export the data as a comma-delimited file (.csv) file, click the **Download** link. To print the data in a portable document format (.pdf) file, click the **Print** link. Please note, this option must have a Report Server configured at the instance level. See "Configuring Report Printing" in *Oracle Application Express Administration Guide*.
Data	Displays a report of the data in the current table. Actions you can perform include: • **Query** - Enables you to sort by column. To restrict specific rows, enter a condition in the Column Condition field. Use the percent sign (%) for wildcards. From Order by, select the columns you want to review and click **Query**. • **Count Rows** - Displays a report of the number of rows in the current table. • **Insert Row** - Enables you to insert a row into the table. • **Download** - Exports all data in the table to a spreadsheet. Click the **Download** link at the bottom of the page to export all data in the selected table. • **Edit** - Click the **Edit** icon to edit a row.
Indexes	Displays indexes associated with this table. Actions you can perform include **Create** and **Drop**.
Model	Displays a graphical representation of the selected table along with all related tables. Related tables are those that reference the current table in a foreign key and those tables referenced by foreign keys within the current table. You can position the cursor over an underlined table name to view the relationship between that table and the current table. Click an underlined table name to view the model of the related table.
Constraints	Displays a list of constraints for the current table. Actions you can perform include **Create**, **Drop**, **Enable**, and **Disable**.
Grants	Displays a list of grants on the current table, including the grantee, the privilege, and grant options. Actions you can perform in this view include **Grant** and **Revoke**.
Statistics	Displays collected statistics about the current table, including the number of rows and blocks, the average row length, sample size, when the data was last analyzed, and the compression status (enabled or disabled). Click **Analyze** to access the Analyze Table Wizard.

Table 2-1 (Cont.) Available Reports for Tables

View	Description
UI Defaults	Displays user interface defaults for forms and reports. User interface defaults enable developers to assign default user interface properties to a table, column, or view within a specified schema. Click **Edit** to edit defined user interface defaults. Click **Create Defaults** to initialize user interface defaults for tables that do not currently have user interface defaults defined.
Triggers	Displays a list of triggers associated with the current table. Actions you can perform include **Create**, **Drop**, **Enable**, and **Disable**. To view trigger details, click the trigger name.
Dependencies	Displays report showing objects referenced by this table, objects this table references, and synonyms for this table.
SQL	Displays the SQL necessary to re-create this table, including keys, indexes, triggers and table definition.

2.7.4 Editing a Table

Click the appropriate buttons on a table in Object Browser to edit a table.

While viewing a table description, you can edit it by clicking the buttons above the table description.

To edit a table:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, ensure **Tables** is selected.
3. From the Object Selection pane, select a table.

   The table description appears.
4. Click the appropriate button described in Table 2-2 (page 2-11).

Table 2-2 Edit Table Buttons

Button	Description
Add Column	Adds a column to the table. Enter a column name and select a type. Depending upon the column type, specify whether the column requires a value, the column length, precision, and scale.
Modify Column	Modifies the selected column. For Oracle Database 11*g* Release 2 (11.2.0.4) and later, a Data Redaction region may be shown if the current schema selected has been granted execute on `DBMS_REDACT`.
Rename Column	Renames the selected column.
Drop Column	Drops the selected column.

Table 2-2 (Cont.) Edit Table Buttons

Button	Description
Rename	Renames the selected table.
Copy	Copies the selected table.
Drop	Drops the selected table.
Truncate	Removes all rows from the selected table. Truncating a table can be more efficient than dropping and re-creating a table. Dropping and re-creating a table may invalidate dependent objects, requiring you to regrant object privileges or re-create indexes, integrity constraints, and triggers.
Create Lookup Table	Creates a lookup table based on the column you select. That column becomes a foreign key to the lookup table.

2.7.5 Dropping a Table

Drop a table using Object Browser.

To drop a table:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, ensure **Tables** is selected.
3. From the Object Selection pane, select a table.

   The table description appears.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.8 Managing Views

A view is a logical representation of another table or combination of tables. You can create, browse, edit, download, compile, drop a view and view reports.

A view does not contain or store data but derives its data from the views or tables on which it is based. These tables are called **base tables**. All operations performed on a view affect the base table of the view. In order to be updatable, a view cannot contain any of the following constructs: `SET` or `DISTINCT`, operators, aggregate or analytic functions, `GROUP BY`, `ORDER BY`, `CONNECT BY`, `START WITH` clause, subquery, or collection expression in a `SELECT` list.

For an example where using a view might be preferable to a table, consider the `HR.EMPLOYEES` table which has several columns and numerous rows. To allow users to see only five of these columns or only specific rows, a view is created as follows:

```
CREATE VIEW staff AS
SELECT employee_id, last_name, job_id, manager_id, department_id
FROM employees
```

- Creating a View (page 2-13)
  Create a view using Object Browser.
- Browsing a View (page 2-14)
  Select a view from the Object Selection pane and view different reports about the view.
- Reports for Views (page 2-14)
  Alternative views available when browsing a view in Object Browser.
- Editing a View Manually (page 2-15)
  When you edit a view you can edit the code manually, perform a search and replace, and compile the view.
- Using Find and Replace (page 2-15)
  Select a view from the Object Selection pane and click Find or Replace.
- Downloading a View (page 2-16)
  Select a view from the Object Selection pane and download a view by clicking Download Source.
- Compiling a View (page 2-16)
  If you edit and make changes to a view, you must compile to save your changes.
- Dropping a View (page 2-16)
  Select a view from the Object Selection pane, select the View or Code tab, and click Drop.

> **See Also:**
>
> *Oracle Database Administrator's Guide*

2.8.1 Creating a View

Create a view using Object Browser.

To create a view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **View**.
4. Define the view:
   - View Name - Enter a name for the View.
   - Query - Enter a query to define the view.

> **Tip:**
>
> To access Query Builder or SQL Commands, click the appropriate button at the bottom of the page. The selected tool displays in a pop-up window. Once you create the appropriate SQL, click **Return** to automatically close the popup window and return to the wizard with the SQL.

5. Click **Next**.

   A confirmation page appears, which displays the SQL used to create the view.

6. Click **Create View**.

2.8.2 Browsing a View

Select a view from the Object Selection pane and view different reports about the view.

When you access a view in Object Browser, the Detail pane displays a report listing the columns in that view.

To browse a view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.

   The view definition appears displaying the appropriate columns.

4. Click the tabs at the top of the page to view different reports about the view.

2.8.3 Reports for Views

Alternative views available when browsing a view in Object Browser.

Table 2-3 (page 2-14) describes all available reports for views.

Table 2-3 Available Reports for Views

View	Description
View	(Default) Displays the columns in the current view. Actions you can perform include: • Compile • Drop
Code	Displays the code editor.

Table 2-3 (Cont.) Available Reports for Views

View	Description
Data	Displays a report of the data in the columns in the view. Actions you can perform include: • **Query** - Enables you to sort by column. To restrict specific rows, enter a condition in the Column Condition field. Use the percent sign (%) for wildcards. From Order by, select the columns you want to review and click **Query**. • **Count Rows** - Enables you to count rows in the table. • **Insert Row** - Enables you to insert a row into the table.
Grants	Displays a list of grants associated with the columns in the view. Grant details include grantee, privilege, and grant options. Actions you can perform include **Grant** and **Revoke**.
UI Defaults	Displays user interface defaults for forms and reports. User interface defaults enable developers to assign default user interface properties to a table, column, or view within a specified schema. Click **Edit** to edit existing user interface defaults. Click **Create Defaults** to initialize user interface defaults for views that do not currently have user interface defaults defined.
Dependencies	Displays a report showing objects referenced by this view, objects this view references, and synonyms for this view.
SQL	Displays the SQL necessary to re-create this view.

2.8.4 Editing a View Manually

When you edit a view you can edit the code manually, perform a search and replace, and compile the view.

You can save the view as a file or drop it.

To edit a view manually:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.
4. Select the **Code** tab.

   The code editor appears.

   If you edit and make changes to a view, you must compile.

2.8.5 Using Find and Replace

Select a view from the Object Selection pane and click Find or Replace.

To use Find and Replace:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

Object Browser appears.

2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.
4. To perform a basic search, click the **Find** icon (which resembles a magnifying glass).
5. To replace existing code, click the **Replace** icon (which resembles a two-sided arrow).

2.8.6 Downloading a View

Select a view from the Object Selection pane and download a view by clicking Download Source.

To save the current view as a file:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.
4. Select the **Code** tab.

   The code editor appears.
5. Click **Download Source**.

2.8.7 Compiling a View

If you edit and make changes to a view, you must compile to save your changes.

There is no save function since this is just a view of the object within the database.

To re-create the current view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.
4. Select the **Code** tab.

   The code editor appears.
5. Click **Save & Compile**.

2.8.8 Dropping a View

Select a view from the Object Selection pane, select the View or Code tab, and click Drop.

To drop a view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Views**.
3. From the Object Selection pane, select a view.
4. Select the **View** tab or the **Code** tab.
5. Click **Drop**.
6. To confirm, click **Finish**.

2.9 Managing Indexes

An index is an optional structure associated with tables and clusters. You can create, browse, drop an index and view reports.

You can create indexes on one or more columns of a table to speed access to data on those columns.

When you view an index in Object Browser, the Detail pane displays a report containing the index name, index type, table owner, table name, table type, uniqueness, compression, prefix length, tablespace name, status, last analyzed and a listing of the indexed columns.

- Creating an Index (page 2-17)
  Create an index using Object Browser.
- Browsing an Index (page 2-18)
  Select an index from the Object Selection pane and view different reports about the an index.
- Reports for Indexes (page 2-18)
  Alternative views available when browsing an index in Object Browser.
- Dropping an Index (page 2-18)
  Select an index from the Object Selection pane and click Drop.

2.9.1 Creating an Index

Create an index using Object Browser.

To create an index:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Index**.
4. Select a table and select the type of index you want to create. Available index types include:
   - **Normal** - Indexes one or more scalar typed object attributes of a table
   - **Text** - Creates a text index (Oracle Text)
5. Click **Next**.
6. Create the index definition. Specify an index name, select one or more columns to be indexed, and click **Next**.

   A confirmation page appears, which displays the SQL used to create the index.

7. To confirm, click **Create Index**.

2.9.2 Browsing an Index

Select an index from the Object Selection pane and view different reports about the an index.

To browse an index:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Indexes**.
3. From the Object Selection pane, select an index.

   The index appears displaying the index name, index type, table owner, table name, table type, uniqueness, compression, prefix length, tablespace name, status, last analyzed and a listing of the indexed columns.
4. Click the tabs at the top of the page to view different reports about the index.

2.9.3 Reports for Indexes

Alternative views available when browsing an index in Object Browser.

Table 2-4 (page 2-18) describes all available reports for indexes.

Table 2-4 Available Reports for Indexes

View	Description
Object Details	Displays the index name, index type, table owner, table name, table type, uniqueness, compression, prefix length, tablespace name, status, last analyzed and also a listing of the indexed columns. Actions you can perform while viewing Object Details include: • Disable - Disables the current index • Drop - Drops the current index • Rebuild - Rebuilds the current index
Statistics	Displays collected statistics about the current view, including the number of rows, sample size, when the data was last analyzed, and the compression status (enabled or disabled). Click **Analyze** to refresh the displayed statistics.
SQL	Displays the SQL necessary to re-create this index.

2.9.4 Dropping an Index

Select an index from the Object Selection pane and click Drop.

To drop an index:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Indexes**.

3. From the Object Selection pane, select an index.
4. Under Object Details, click **Drop**.
5. To confirm, click **Finish**.

2.10 Managing Sequences

A sequence generates a serial list of unique numbers for numeric columns of a database table. You can create, browse, drop, alter a sequence and view a report.

Database sequences are generally used to populate table primary keys.

- Creating a Sequence (page 2-19)
  Create a sequence using Object Browser.
- Browsing a Sequence (page 2-20)
  Select a sequence from the Object Selection pane and view different reports about the sequence.
- Reports for Sequences (page 2-20)
  Alternative views available when browsing a sequence in Object Browser.
- Dropping a Sequence (page 2-20)
  Select a sequence from the Object Selection pane and click Drop.
- Altering a Sequence (page 2-21)
  Select a sequence from the Object Selection pane and click Alter.

2.10.1 Creating a Sequence

Create a sequence using Object Browser.

To create a sequence:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Sequence**.
4. For Define:
   a. **Sequence Name** - Enter the name of the sequence.
   b. **Preserve Case** - To have the final sequence name match the case entered in the Sequence Name field, click **Preserve Case**.
   c. **Start With** - Enter the number of the first sequence. The first reference to `sequence_name.nextval` returns this number.
   d. **Minimum Value** - Enter the minimum value this sequence can return.
   e. **Maximum Value** - Enter the maximum value this sequence can return.
   f. **Increment By** - Each call to `sequence_name.nextval` returns a value greater than the last, until the maximum value is reached. Enter the value used to increment to the next sequence number.
   g. **Cycle** - Select this option to restart the sequence number to the minimum value when the maximum value is reached. Note that this is not recommended if using the sequence for primary key creation.

h. **Number to Cache** - For faster access, specify how many sequence values are stored in memory.

i. **Order** - Specify ORDER to guarantee that sequence numbers are generated in order of request. This option is necessary if using Real Application Clusters (Oracle RAC).

5. Click **Next**.

   A confirmation page appears, which displays the SQL used to create the sequence.

6. Click **Create Sequence**.

2.10.2 Browsing a Sequence

Select a sequence from the Object Selection pane and view different reports about the sequence.

To browse a sequence:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Sequences**.
3. From the Object Selection pane, select a sequence.

   The Object Details view appears.

4. Click the tabs at the top of the page to view different reports about the sequence.

2.10.3 Reports for Sequences

Alternative views available when browsing a sequence in Object Browser.

Table 2-5 (page 2-20) describes all available reports for sequences.

Table 2-5 Available Reports for Sequences

View	Description
Object Details	Displays details about the current sequence. You can perform **Alter** and **Drop** in this view.
Grants	Displays a list of grants associated with the sequence. Grant details include grantee, privilege, and grant options. You can perform **Grant** and **Revoke** in this view.
Dependencies	Displays a list of objects that use (or depend) upon this sequence.
SQL	Displays the SQL necessary to re-create this sequence.

2.10.4 Dropping a Sequence

Select a sequence from the Object Selection pane and click Drop.

To drop a sequence:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

Object Browser appears.

2. From the Object list, select **Sequences**.
3. From the Object Selection pane, select a sequence.

   The Object Details view appears.
4. Click **Drop**.

2.10.5 Altering a Sequence

Select a sequence from the Object Selection pane and click Alter.

To alter a sequence

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Sequences**.
3. From the Object Selection pane, select a sequence.

   The Object Details view appears.
4. Click **Alter**.

2.11 Managing Types

A type is a user-specified object or collection definition. You can create, browse, drop a type and view a report.

Oracle Application Express currently only supports collection definitions. There are two categories of Oracle collections (SQL collections):

- Variable-length arrays (`VARRAY` types)
- Nested tables (`TABLE` types)

`VARRAY` types are used for one-dimensional arrays, while nested table types are used for single-column tables within an outer table.

- Creating a Type (page 2-22)
  Create a type using Object Browser.
- Browsing a Type (page 2-22)
  Select a type from the Object Selection pane and view different reports for the type.
- Reports for Types (page 2-22)
  Alternative views available when viewing a type in Object Browser.
- Dropping a Type (page 2-23)
  Select a type from the Object Selection pane and click Drop.

> **See Also:**
>
> - "*Oracle Database Concepts*"
> - "*Oracle Database PL/SQL Language Reference*" and for information about collection types

2.11.1 Creating a Type

Create a type using Object Browser.

To create a collection type:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. For Name:
   a. **Type Name** - Enter the name of the type to be created.
   b. **Preserve Case** - To have the type name match the case entered in the Type Name field, click **Preserve Case**.
   c. **Type to Create** - Specify the kind of type to create.
   d. Click **Next**.
4. Follow the on-screen instructions.

   A confirmation page appears, which displays the SQL used to create the type.
5. Click **Create Type**.

2.11.2 Browsing a Type

Select a type from the Object Selection pane and view different reports for the type.

To browse a collection type:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Types**.
3. From the Object Selection pane, select a type.

   The Object Details view appears.
4. Click the tabs at the top of the page to view different reports for the type.

2.11.3 Reports for Types

Alternative views available when viewing a type in Object Browser.

Table 2-6 (page 2-23) describes all available reports for types.

Table 2-6 Available Reports for Types

View	Description
Object Details	Displays details about the selected type. To drop a type, click **Drop**.
Synonyms	Displays a list of synonyms for the current type.
Grants	Displays a list of grants associated with the type. Grant details include grantee, privilege, and grant options. You can perform **Grant** and **Revoke** actions.
SQL	Displays the SQL necessary to re-create this type.

2.11.4 Dropping a Type

Select a type from the Object Selection pane and click Drop.

To drop a collection type:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Types**.
3. From the Object Selection pane, select a type.

   The Object Details view appears.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.12 Managing Packages

A package is a database object that groups logically related PL/SQL types, items, functions and procedures. You can create, view, edit, compile, drop, download a package, and view a report.

Packages usually have two parts, a specification and a body. The **specification** is the interface to your application. The **body** implements the specification.

- Creating a Package (page 2-24)
  Create a package using Object Browser. Select the type of package you want to create.
- Viewing a Package (page 2-25)
  Select a package from the Object Selection pane and view different reports about the package.
- Reports for Packages (page 2-25)
  Alternative views available when viewing a package in Object Browser.
- Editing a Package Manually (page 2-26)
  Select a package from the Object Selection pane and click Find or Replace.
- Compiling a Package (page 2-26)
  If you edit and make changes to a view, you must compile to save your changes.

- Downloading a Package (page 2-27)
  Select a package from the Object Selection pane and download a package by clicking Download Source to save the current package as a file.
- Dropping a Package While Viewing Specification (page 2-27)
  Select a package from the Object Selection pane and click Drop.
- Dropping a Package While Viewing the Body (page 2-27)
  Select a package from the Object Selection pane, click Body tab, and click Drop.

> **See Also:**
>
> - "Using PL/SQL Packages " in *Oracle Database PL/SQL Language Reference* for additional information on PL/SQL packages.
> - "Using PL/SQL Subprograms" in *Oracle Database PL/SQL Language Reference* for information on PL/SQL subprograms

2.12.1 Creating a Package

Create a package using Object Browser. Select the type of package you want to create.

To create a package:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Package**.
4. For Create Package, select the type of package you want to create:
   - **Specification**
   - **Body**
   - **Package with methods on database table(s)**

     Packages usually have two parts: **Specification** and **Body**.
5. Click **Next**.
6. If you select **Specification**:
   a. Package Name - Enter a package name. To preserve the case of the name you enter, click **Preserve Case**.
   b. Click **Next**.

      The wizard creates a dummy package specification and displays it for editing.
   c. Edit the specification and click **Create Package Specification**.
7. If you select **Body**:
   a. Package - Select the package you want to create the body for and click **Next**.

      The wizard creates a package body with stubbed out calls identified in the specification and displays it for editing.

   b. Edit the package body and click **Create Package Body**.

8. If you select **Package with methods on database tables**:
   a. Package Name - Enter a package name. To preserve the case of the name you enter, click **Preserve Case**.
   b. Select up to ten tables and click **Next**.

      The wizard creates a specification and body with insert, update, delete, and GET APIs for the selected tables. Note that you have the option to show or download the specification or body.
   c. To confirm, click **Create Package**.

2.12.2 Viewing a Package

Select a package from the Object Selection pane and view different reports about the package.

When you access a package in Object Browser the specification appears.

To view a specification:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Packages**.
3. From the Object Selection pane, select a package.

   The Specification appears. You can also edit the code in this view.
4. Click the tabs at the top of the page to view different reports about the package.

2.12.3 Reports for Packages

Alternative views available when viewing a package in Object Browser.

Table 2-7 (page 2-25) describes all available reports for packages.

Table 2-7 Available Reports for Packages

View	Description
Specification	Displays the package specification. This defines the interface to your application. Actions you can perform include: • Edit the code manually • **Save & Compile** • **Find** • **Replace** • **Auto Complete** • **Undo** • **Redo** • **Download Source** • **Drop**

Table 2-7 (Cont.) Available Reports for Packages

View	Description
Body	Displays the package body, if one exists, for the selected package. Actions you can perform include: • Edit the code manually • **Save & Compile** • **Find** • **Replace** • **Auto Complete** • **Undo** • **Redo** • **Download Source** • **Drop**
Dependencies	Displays objects that use (or depend on) on the current package and objects the package depends on.
Errors	Displays errors related to the current package.
Grants	Lists details of grants for the current package, including privilege, grantee, grantable, grantor, and object name.

2.12.4 Editing a Package Manually

Select a package from the Object Selection pane and click Find or Replace.

When you edit a package, you can edit the code manually, perform a search and replace, and compile the package.

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Packages**.
3. From the Object Selection pane, select a package.

   The Specification appears. You can edit the code in this view.
4. To perform a basic search, click the **Find** icon (which resembles a magnifying glass).
5. To replace existing code, click the **Replace** icon (which resembles a two-sided arrow).

2.12.5 Compiling a Package

If you edit and make changes to a view, you must compile to save your changes.

There is no save function because this is just a view of the object within the database.

To compile a package:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Packages**.

3. From the Object Selection pane, select a package.

   The Specification appears.

4. Click **Save & Compile** to compile the current package.

   Compiling re-creates the object in the database. If the compilation fails, an error message displays above the code.

2.12.6 Downloading a Package

Select a package from the Object Selection pane and download a package by clicking Download Source to save the current package as a file.

To download a package:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Packages**.
3. From the Object Selection pane, select a package.

   The Specification appears.

4. Click **Download Source** to save the current package as a file.

2.12.7 Dropping a Package While Viewing Specification

Select a package from the Object Selection pane and click Drop.

To drop a package while viewing the Specification:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Packages**.
3. From the Object Selection pane, select a package.

   The Specification appears.

4. Click **Drop**.
5. To confirm, click **Finish**.

2.12.8 Dropping a Package While Viewing the Body

Select a package from the Object Selection pane, click Body tab, and click Drop.

To drop a package while viewing the Body:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.

2. From the Object list, select **Packages**.
3. From the Object Selection pane, select a package.

   The Specification appears.

4. Click the **Body** tab.

5. Click **Drop**.
6. To confirm, click **Finish**.

2.13 Managing Procedures

A procedure is a subprogram that performs a specific action. You can use Object Browser to view, create, edit, download, and drop procedures.

- Creating a Procedure (page 2-28)
  Create procedure using Object Browser.
- Browsing a Procedure (page 2-29)
  Select a procedure from the Object Selection pane and view different reports about the procedure.
- Reports for Procedures (page 2-29)
  Alternative views available when viewing a procedure in Object Browser.
- Editing a Procedure (page 2-30)
  When you edit a procedure you can edit the code manually or perform a search and replace.
- Compiling a Procedure (page 2-31)
  If you edit and make changes to a view, you must compile to save your changes.
- Downloading a Procedure (page 2-31)
  Select a procedure from the Object Selection pane and click Download Source to save the current procedure as a file.
- Dropping a Procedure (page 2-31)
  Select a procedure from the Object Selection pane and click Drop.

> **See Also:**
>
> "*Oracle Database PL/SQL Language Reference*"

2.13.1 Creating a Procedure

Create procedure using Object Browser.

To create a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Procedure**.
4. For Name:
   a. **Procedure Name** - Enter the name of the procedure.

> **Note:**
>
> Object names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

b. Specify whether to Include Arguments or Preserve Case.

c. Click **Next**.

5. For Arguments, define the arguments by specifying the following information (optional):
   - **Argument Name**
   - **In/Out (the parameter mode)**
   - **Argument Type (datatype)**
   - **Default (value)**
   - **Move**

   To add additional arguments, click **Add Argument**.
6. Click **Next**.
7. For Define, enter PL/SQL block you want to use as the procedure body and click **Next**.

   A confirmation page appears.
8. Click **Create Procedure**.

2.13.2 Browsing a Procedure

Select a procedure from the Object Selection pane and view different reports about the procedure.

To browse a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Procedures**.
3. From the Object Selection pane, select a procedure.

   The Code view appears, displaying the source code for the procedure. You can edit the code in this view.
4. Click the tabs at the top of the page to view different reports about the procedure.

2.13.3 Reports for Procedures

Alternative views available when viewing a procedure in Object Browser.

Table 2-8 (page 2-30) describes all available reports for procedures.

Table 2-8 Available Views for Procedures

View	Description
Code	(Default) Displays the source code for the procedure. You can edit the code in this view. Actions you can perform in this view include: • Edit the code manually • **Save & Compile** • **Download Source** • **Drop** • **Undo** • **Redo** • **Find** • **Replace** • **Auto Complete**
Dependencies	Displays objects that use (or depend) on the current procedure and objects the procedure depends on.
Errors	Lists errors related to the current procedure.
Grants	Lists details of grants for the current procedure, including privilege, grantee, grantable, grantor, and object name.

2.13.4 Editing a Procedure

When you edit a procedure you can edit the code manually or perform a search and replace.

- Editing a Procedure Manually (page 2-30)
  Select a procedure from the Object Selection pane and click Find or Replace.

2.13.4.1 Editing a Procedure Manually

Select a procedure from the Object Selection pane and click Find or Replace.

To edit a procedure manually:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Procedures**.
3. From the Object Selection pane, select a procedure.

   The Code view appears. You can edit the code in this view.
4. To perform a basic search, click the **Find** icon (which resembles a magnifying glass).
5. To replace existing code, click the **Replace** icon (which resembles a two-sided arrow).

2.13.5 Compiling a Procedure

If you edit and make changes to a view, you must compile to save your changes.

There is no save function because this is just a view of the object within the database.

To compile a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Procedures**.
3. From the Object Selection pane, select a procedure.
4. Click **Save & Compile** to compile the current procedure.

   Compiling re-creates the object in the database. If the compilation fails, an error message displays above the code.

2.13.6 Downloading a Procedure

Select a procedure from the Object Selection pane and click Download Source to save the current procedure as a file.

To download a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Procedures**.
3. From the Object Selection pane, select a procedure.
4. Click **Download Source** to save the current procedure as a file.

2.13.7 Dropping a Procedure

Select a procedure from the Object Selection pane and click Drop.

To drop a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Procedures**.
3. From the Object Selection pane, select a procedure.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.14 Managing Functions

A function is a subprogram that can take parameters and return a single value. You can use Object Browser to create, browse, edit, compile, download, drop functions and view report.

- Creating a Function (page 2-32)
  Create a function using Object Browser.
- Browsing a Function (page 2-33)
  Select a function from the Object Selection pane and view different reports about the function.
- Reports for Functions (page 2-33)
  Alternative views available when browsing a function in Object Browser.
- Editing a Function Manually (page 2-34)
  Select a function from the Object Selection pane and click Edit.
- Compiling a Function (page 2-34)
  Select a function from the Object Selection pane click Save & Compile.
- Downloading a Function (page 2-35)
  Select a function from the Object Selection pane click Download Source to save the current function as a file.
- Dropping a Function (page 2-35)
  Select a function from the Object Selection pane and click Drop.

> **See Also:**
>
> - "*Oracle Database SQL Language Reference*" for information about PL/SQL functions
> - "*Oracle Database PL/SQL Language Reference*"

2.14.1 Creating a Function

Create a function using Object Browser.

To create a function:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Function**.
4. For Name:
   a. **Function Name** - Enter the name of the function.

      > **Note:**
      >
      > Object names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

   b. **Return Data Type** - Select a data type.
   c. Specify whether to Include Arguments or Preserve Case.

d. Click **Next**.

5. For Arguments, define the arguments by specifying the following information (optional):
   - **Argument Name**
   - **Argument Type (datatype)**
   - **Default (value)**
   - **Move**

   To add additional arguments, click **Add Argument**.
6. Click **Next**.
7. For Function Body, enter PL/SQL block you want to use and click **Next**.

   A confirmation page appears.
8. Click **Create Function**.

2.14.2 Browsing a Function

Select a function from the Object Selection pane and view different reports about the function.

To view a function in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Functions**.
3. From the Object Selection pane, select a function.

   The Code view appears. You can edit the code in this view.
4. Click the tabs at the top of the page to view different reports about the function.

2.14.3 Reports for Functions

Alternative views available when browsing a function in Object Browser.

Table 2-9 (page 2-34) describes all available reports for functions.

Table 2-9 Available Reports for Functions

View	Description
Code	(Default) Displays the source code for the function. You can edit the code in this view. Actions you can perform in this view include: • Edit the code manually • **Save & Compile** • **Find** • **Replace** • **Auto Complete** • **Undo** • **Redo** • **Download Source** • **Drop**
Dependencies	Displays objects that use (or depend) on the current function and objects the function depends on.
Errors	Displays errors related to the current function.
Grants	Lists details of grants for the current function, including privilege, grantee, grantable, grantor, and object name.

2.14.4 Editing a Function Manually

Select a function from the Object Selection pane and click Edit.

When you edit a function you can edit the code manually, perform a search and replace, and compile the function.

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Functions**.
3. From the Object Selection pane, select a function.

   The Code view appears. You can edit the code in this view.
4. Click **Edit** to activate manual edit mode.
5. To perform a basic search, click the **Find** icon (which resembles a magnifying glass).
6. To replace existing code, click the **Replace** icon (which resembles a two-sided arrow).

2.14.5 Compiling a Function

Select a function from the Object Selection pane click Save & Compile.

If you edit and make changes to a function, you must compile to save your changes. There is no save function because this is just a view of the object within the database.

To compile a function in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Functions**.
3. From the Object Selection pane, select a function.

   The Code view appears.
4. Click **Save & Compile** to compile the current function.

   Compiling re-creates the object in the database. If the compilation fails, an error message displays above the code.

2.14.6 Downloading a Function

Select a function from the Object Selection pane click Download Source to save the current function as a file.

To save a function to a file in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Functions**.
3. From the Object Selection pane, select a function.

   The Code view appears.
4. Click **Download Source** to save the current function as a file.

2.14.7 Dropping a Function

Select a function from the Object Selection pane and click Drop.

To drop a function in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Functions**.
3. From the Object Selection pane, select a function.

   The Code view appears.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.15 Managing Triggers

A database trigger is a stored subprogram associated with a database table, view, or event. You can create, browse, edit, compile, download, drop, disable or enable a trigger and view report.

The trigger can be called once, for example when an event occurs, or many times, for example for each row affected by an INSERT, UPDATE, or DELETE statement.

- Creating Triggers (page 2-36)
  Create a trigger using Object Browser.
- Browsing a Trigger (page 2-36)
  Select a trigger from the Object Selection pane and view different reports about the trigger.
- Reports for Triggers (page 2-37)
  Alternative views available when viewing a trigger in Object Browser.
- Editing a Trigger Manually (page 2-37)
  Select a trigger from the Object Selection pane and click Code.
- Compiling a Trigger (page 2-38)
  Select a trigger from the Object Selection pane and click Save & Compile.
- Downloading a Trigger (page 2-38)
  Select a trigger from the Object Selection pane and click Download Source to save the current trigger as a file.
- Dropping a Trigger (page 2-39)
  Select a trigger from the Object Selection pane and click Drop.
- Disabling and Enabling a Trigger (page 2-39)
  Select a trigger from the Object Selection pane and click Disable or Enable.

See Also:

- "*Oracle Database Concepts*"
- "*Oracle Database Advanced Application Developer's Guide*"

2.15.1 Creating Triggers

Create a trigger using Object Browser.

To create a trigger in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Trigger**.
4. Select a table name and click **Next**.
5. Select the appropriate trigger attributes, enter the trigger body, and click **Next**.

   A confirmation page appears, which displays the SQL used to create the trigger.
6. Click **Create Trigger**.

2.15.2 Browsing a Trigger

Select a trigger from the Object Selection pane and view different reports about the trigger.

To browse a trigger in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Object Details view appears.
4. Click the tabs at the top of the page to view different reports about the trigger.

2.15.3 Reports for Triggers

Alternative views available when viewing a trigger in Object Browser.

Table 2-10 (page 2-37) describes all available reports for triggers.

Table 2-10 Available Reports for Triggers

View	Description
Object Details	(Default) Lists of the details about the current trigger. Actions you can perform include: • **Compile** • **Download** • **Drop** • **Disable**
Code	Displays the source code for the trigger. You can edit the code in this view. Actions you can perform in this view include: • Edit the code manually • **Save & Compile** • **Find** • **Replace** • **Auto Complete** • **Undo** • **Redo** • **Download Source** • **Drop**
Errors	Displays errors related to the current trigger.
SQL	Displays the SQL necessary to re-create the trigger.

2.15.4 Editing a Trigger Manually

Select a trigger from the Object Selection pane and click Code.

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Object Details view appears.
4. Click Code tab.

5. You can edit the code in this view. Other edit controls include:
   - **Find**
   - **Replace**
   - **Auto Complete**

When you edit a trigger you can edit the code manually, perform a search and replace, and compile the trigger.

2.15.5 Compiling a Trigger

Select a trigger from the Object Selection pane and click Save & Compile.

If you edit and make changes to a function, you must compile to save your changes. There is no save function because this is just a view of the object within the database. Compiling re-creates the object in the database.

To compile a trigger in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Details view appears.
4. To compile the current trigger:
   - Under Object Details, click **Compile**.
   - Under Code, click **Save & Compile**.

   If the compilation fails, an error message displays above the code.

2.15.6 Downloading a Trigger

Select a trigger from the Object Selection pane and click Download Source to save the current trigger as a file.

To save the current trigger as a file:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Details view appears.
4. Click **Download Source** to save the current trigger as a file.
5. To download the current trigger:
   - Under Object Details, click **Download**.
   - Under Code, click **Download Source**.

   If the compilation fails, an error message displays above the code.

2.15.7 Dropping a Trigger

Select a trigger from the Object Selection pane and click Drop.

To drop a trigger in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Details view appears.
4. Click **Drop**.

2.15.8 Disabling and Enabling a Trigger

Select a trigger from the Object Selection pane and click Disable or Enable.

To disable and enable a trigger in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Triggers**.
3. From the Object Selection pane, select a trigger.

   The Details view appears.
4. Click **Disable**.

> **Note:**
>
> The Enable button is activated. If you want to enable the trigger, click **Enable**.

2.16 Managing Database Links

A database link is a schema object in one database that enables you to access objects in another database. You can create, browse, drop a database link and view report.

Once you create a database link, you can access the remote objects by appending `@dblink` to the table or view name, where `dblink` is the name of the database link.

- Creating a Database Link (page 2-40)
  Create a database link using Object Browser.
- Browsing a Database Link (page 2-40)
  Select a database link from the Object Selection pane and view different reports about the database link.
- Reports for Database Links (page 2-41)
  Alternative views available when viewing a database links in Object Browser.

- Dropping a Database Link (page 2-41)
  Select a database link from the Object Selection pane and click Drop.

2.16.1 Creating a Database Link

Create a database link using Object Browser.

To create a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Database Link**.
4. For Define :
   a. **Database Link Name** - Enter the name of the database link. Database link names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.
   b. **Connect To Schema**- Enter the name of the database schema on the remote system to access.
   c. **Password** - Enter the password for the remote database schema.
   d. **Remote Hostname or IP** - Enter the hostname or IP address of the database server you wish to access.
   e. **Remote Host Port** - Enter the network port number on which the Oracle Net Listener is accepting requests. This is typically 1521.
   f. Identified by - Select Service Name or SID.

      If you select **SID** (Oracle System Identifier), enter the SID name if the remote database is identified by a SID. If you select **Service Name**, enter the service name if the remote database is identified by a service name.
5. Click **Next**.
6. Click **Create Database Link**.

2.16.2 Browsing a Database Link

Select a database link from the Object Selection pane and view different reports about the database link.

To browse a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Database Links**.
3. From the Object Selection pane, select a database link.

   The Object Details view appears.
4. Click the tabs at the top of the page to view different reports about the database link.

2.16.3 Reports for Database Links

Alternative views available when viewing a database links in Object Browser.

Table 2-11 (page 2-41) describes all available reports for database links.

Table 2-11 Available Reports for Database Links

View	Description
Object Details	(Default) Displays details about the database link. Actions you can perform include: • **Drop** - Deletes the database link • **Test** - Tests the database link
Dependencies	Displays a list of objects that use (or depend) upon this database link.
SQL	Displays the SQL necessary to re-create this database link.

2.16.4 Dropping a Database Link

Select a database link from the Object Selection pane and click Drop.

To drop a database link in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Database Links**.
3. From the Object Selection pane, select a database link.

   The Object Details View appears.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.17 Managing Materialized Views

A materialized view provides indirect access to table data by storing the results of a query in a separate schema object. You can create, browse, drop a materialized view and view report.

Unlike an ordinary view, which does not take up any storage space or contain any data, a materialized view contains the rows resulting from a query against one or more base tables or views. A materialized view can be stored in the same database as its base tables or in a different database.

Materialized views are often used to improve performance, especially when storing data locally that is sourced from tables or views using a database link. Similarly, materialized views are often used instead of views when there are extensive calculations or summarizations, especially on particularly large data sets. Materialized views are refreshed at specified intervals which means the data is not always up-to-date but queries against them are significantly faster for the examples given above.

- Creating a Materialized View (page 2-42)
  Create a materialized view using Object Browser.
- Browsing a Materialized View (page 2-43)
  Select a materialized view from the Object Selection pane and view different reports about the materialized view.
- Reports for Materialized Views (page 2-43)
  Alternative views available when viewing a materialized view in Object Browser.
- Dropping a Materialized View (page 2-44)
  Select a materialized view from the Object Selection pane and click Drop.

> **See Also:**
>
> "*Oracle Database Concepts*" for information about materialized views

2.17.1 Creating a Materialized View

Create a materialized view using Object Browser.

To create a materialized view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. Click the **Create** icon.
3. From the list of object types, select **Materialized View**.
4. For Define:
   a. **Materialized View Name** - Enter the name of the view. View names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

      To preserve the case of the name you enter, click **Preserve Case**.
   b. **Query** - Specify a query to define the view.

      > **Tip:**
      >
      > To access Query Builder or SQL Commands, click the appropriate button at the bottom of the page. The selected tool displays in a pop-up window. Once you create the appropriate SQL, click **Return** to automatically close the popup window and return to the wizard with the SQL.
   c. Click **Next**.

      A confirmation page appears, which displays the SQL used to create the materialized view.
5. Click **Create Materialized View**.

2.17.2 Browsing a Materialized View

Select a materialized view from the Object Selection pane and view different reports about the materialized view.

To view a materialized view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Materialized Views**.
3. From the Object Selection pane, select a view.

   The Materialized View appears.
4. Click the tabs at the top of the page to view different reports about the materialized view.

2.17.3 Reports for Materialized Views

Alternative views available when viewing a materialized view in Object Browser.

Table 2-12 (page 2-43) describes all available reports for materialized views.

Table 2-12 Available Reports for Materialized Views

View	Description
Materialized View	(Default) Displays details about the columns in the materialized view, including: • **Column Name** • **Data type** • **Nullable flag** • **Default value** • **Primary key** Click **Drop** to delete the current materialized view.
Data	Displays a report of the data in the columns. Actions you can perform include: • **Query** - Enables you to sort by column. To restrict specific rows, enter a condition in the Column Condition field. Use the percent sign (%) for wildcards. From Order by, select the columns you want to review and click Query. • **Count Rows** - Displays a report of the data in the current table. • **Download** - Click this link to export the data as a comma-delimited file (.csv) file.
Details	Displays object details stored in `DBA_SNAPSHOTS` such as updatable and status.
Grants	Displays a list of grants on the current view, including grantee, privilege, and grant options. You can perform **Grant** and **Revoke** in this view.
Dependencies	Displays a list of objects that use (or depend) upon this materialized view.

Table 2-12 (Cont.) Available Reports for Materialized Views

View	Description
SQL	Displays the SQL necessary to re-create this materialized view.

2.17.4 Dropping a Materialized View

Select a materialized view from the Object Selection pane and click Drop.

To drop a materialized view in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Materialized Views**.
3. From the Object Selection pane, select a view.

   The Materialized View appears.
4. Click **Drop**.
5. To confirm, click **Finish**.

2.18 Managing Synonyms

A synonym is an alias for a schema object. You can create, view and drop a synonym.

Synonyms can provide a level of security by masking the name and owner of an object and by providing location transparency for remote objects of a distributed database. Also, they are convenient to use and reduce the complexity of SQL statements for database users.

- Creating Synonyms (page 2-44)
  Create synonyms using Object Browser.
- Viewing a Synonym (page 2-45)
  Select a synonym from the Object Selection pane.
- Dropping a Synonym (page 2-45)
  Select a synonym from the Object Selection pane and click Drop.

> **See Also:**
>
> "*Oracle Database Administrator's Guide*" for information about synonyms

2.18.1 Creating Synonyms

Create synonyms using Object Browser.

To create a synonym:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

Object Browser appears.

2. Click the **Create** icon.
3. From the list of object types, select **Synonym**.
4. For Define:
   a. **Synonym Name** - Enter a name.
   b. **Public or Private** - Specify whether the synonym should be public or private.
   c. **Schema** - Select the database schema (or user name) which owns the object upon which you want to create your synonym.
   d. **Object** - Enter the name of the object upon which you want to create a synonym.
   e. **Database Link** - Enter the name of the database link to use if the synonym is to be created on a remote object.
   f. Click **Next**.

   A confirmation page appears, which displays the SQL used to create the synonym.
5. Click **Create Synonym**.

2.18.2 Viewing a Synonym

Select a synonym from the Object Selection pane.

To view a synonym:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Synonyms**.
3. From the Object Selection pane, select a synonym.

   The Object Details view appears displaying the following:
   - **Synonym Owner**
   - **Synonym Name**
   - **Object Owner**
   - **Object Name**
   - **Object Type**
   - **Status**

2.18.3 Dropping a Synonym

Select a synonym from the Object Selection pane and click Drop.

To drop a synonym in Object Browser:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser appears.
2. From the Object list, select **Synonyms**.

3. From the Object Selection pane, select a synonym and click **Drop**.
4. To confirm, click **Finish**.

3
Using SQL Commands

Using SQL commands you can create, edit, view, run, and delete database objects.

- About SQL Commands (page 3-2)
  You can use SQL Commands to create, edit, view, run, and delete database objects.
- Accessing SQL Commands (page 3-2)
  Learn how to access SQL Commands.
- About the SQL Commands Home Page (page 3-3)
  Learn about the SQL commands home page.
- Using the Command Editor (page 3-5)
  You use the command editor in SQL Commands to execute SQL commands within Oracle Application Express.
- Saving an SQL Command (page 3-8)
  You can save commands you enter in SQL Commands.
- Copying a Command (page 3-8)
  Click Saved SQL from the SQL Commands page, click the name of the command to load into command editor, enter a name, and click Save.
- Viewing Results (page 3-9)
  When you execute a SQL command, the results display in the display pane at the bottom of the SQL Commands home page.
- Using Explain Plan (page 3-10)
  Explain plan parses a query and records the plan that Oracle devises to execute it.
- Using Saved Commands (page 3-10)
  You can access the commands you save and commands saved by other users in the same workspace.
- Using SQL Command History (page 3-11)
  You use SQL Command History to access commands you have executed in SQL Commands.

See Also:

- "*Oracle Database SQL Language Reference*" for detailed information about SQL statements and other parts of SQL, such as operators, functions, and format models
- "*Oracle Database Concepts*" for conceptual information about SQL
- "*SQL*Plus User's Guide and Reference*" for information about SQL*Plus, Oracle's version of SQL
- "*Oracle Database Sample Schemas*" for information about the `HR` sample schema that is used for examples in this chapter

3.1 About SQL Commands

You can use SQL Commands to create, edit, view, run, and delete database objects.

A SQL command can contain SQL statements or PL/SQL blocks.

When using SQL Commands, remember the following:

- SQL commands created in the Query Builder can be accessed from SQL Commands.
- Saved SQL commands must have unique names within a given workspace.
- There is no interaction between SQL Commands and SQL Scripts.
- You can cut and paste an SQL command from SQL Commands to run in the SQL Script Editor.

Tip:

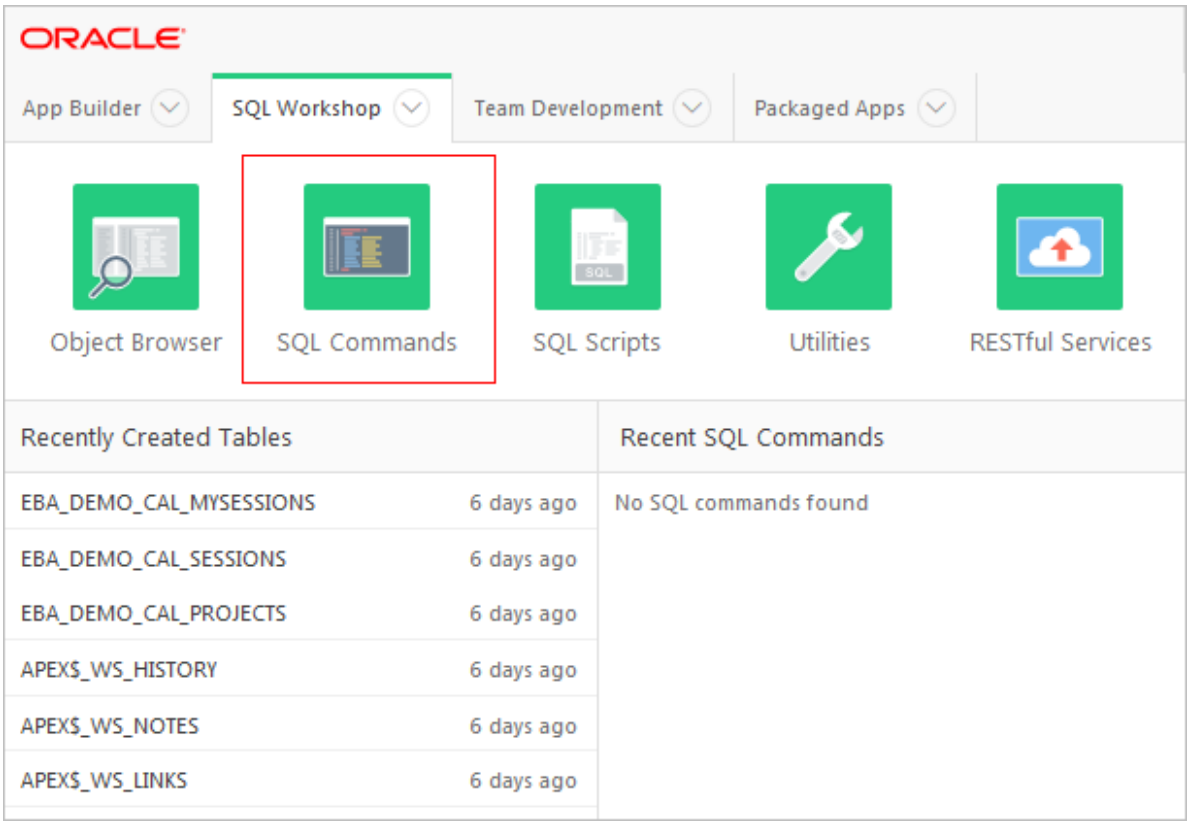
SQL commands created in the Query Builder can be accessed from SQL Commands.

3.2 Accessing SQL Commands

Learn how to access SQL Commands.

To access SQL Commands:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **SQL Commands**.



The SQL Commands home page appears.

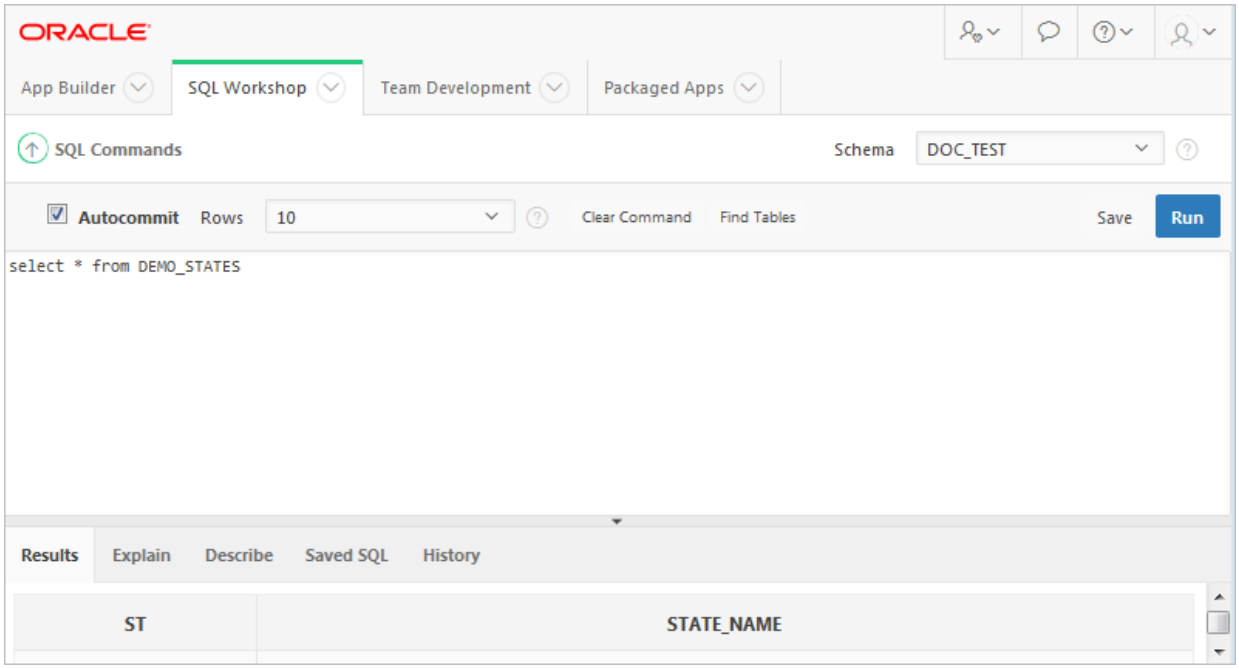
> **Note:**
>
> If the instance administrator has disabled RESTful Services for this Application Express instance, RESTful Services are not available for this instance and the RESTful Services icon does not display.

> **See Also:**
>
> "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*

3.3 About the SQL Commands Home Page

Learn about the SQL commands home page.

The SQL Commands home page is divided into two sections: a command editor and a display pane. You use the command editor to execute SQL commands and the display pane to view output, saved command lists, and history lists.



The top of the SQL Commands home page features a command editor and the following controls:

- **Autocommit**. If available, click the **Autocommit** check box to enable autocommit and disable transactional commands.
- **Rows**. Select the number of rows of output to display simultaneously up to a maximum of 100,000. All rows of DBMS Output are displayed regardless of the Display list setting.
- **Clear Command**. Use the Clear Command button to clear the text in the command editor.
- **Find Tables**. Use the Find Tables button to view tables within the currently selected schema.
- **Save**. Click the **Save** button to save the contents of the command editor, or the currently highlighted content to a file. You are prompted to enter a name and an optional description. The command appears in the Saved SQL list.
- **Run**. Click the **Run** button (**Ctrl+Enter**) to run the command in the command editor, or the currently highlighted command in the command editor.

The display pane is at the bottom of the SQL Commands home page and features the following five tabs:

- **Results**. Click the **Results** tab to see the results from the last successfully executed SQL command. Click **DBMS Output** at the bottom of the displayed results to display lines of DBMS output. This control only appears when there is DBMS output to display. Click **Download** to export results to a comma-delimited file on your local file system.
- **Explain**. Click the **Explain** tab to examine the execution plan used by the optimizer for statements that make changes to the database. Objects in the output are linked to the Object Browser. Click the linked object to view its properties in the Object Browser.

- **Describe**. Enter Describe *object_name* and click **Run** to display column definitions for a table or view, or specifications for a function or procedure in the **Describe** tab. Select links in the Describe results to write that information into the command editor. For example, click a table name to add *owner.table*, click a column name to add the *column name*, click a procedure or function name to add the object call with parameters, or click a package name to add the package call.
- **Saved SQL**. Click the **Saved SQL** tab to display a list of all SQL commands saved in the current workspace. Click the command title to load it into the command editor.
- **History**. Click the **History** tab to list your recently executed commands. Your last 200 executed commands are saved.

3.4 Using the Command Editor

You use the command editor in SQL Commands to execute SQL commands within Oracle Application Express.

- Running a SQL Command (page 3-6)
  Enter the SQL command you want to run in the command editor and click Run.
- About Transactions in SQL Commands (page 3-6)
  Transactional mode is a stateful transaction mode where you can, for example, perform an update, select data for review, and `COMMIT` or `ROLLBACK` changes. It is implemented using `DBMS_JOBS`.
- Disabling Transactional SQL Commands (page 3-6)
  Check the Autocommit check box to disable transactional SQL commands.
- Enabling Transactional SQL Commands (page 3-7)
  Uncheck the Autocommit check box to enable the transactional SQL commands.
- About Unsupported SQL*Plus Commands (page 3-7)
  SQL Commands does not support SQL*Plus commands.
- Terminating a Command (page 3-7)
  You can terminate a command in SQL Commands using a semicolon (;), a slash (/), or with nothing.
- Using Bind Variables (page 3-7)
  Enter values for bind variables during command execution.
- Using the Find Tables Button (page 3-7)
  Use the Find Tables button to view tables within the currently selected schema.

See Also:

"About the SQL Commands Home Page (page 3-3)"

3.4.1 Running a SQL Command

Enter the SQL command you want to run in the command editor and click Run.

To execute a SQL Command:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. Enter the SQL command you want to run in the command editor.
3. Click **Run** (**Ctrl+Enter**) to execute the command.

   > **Tip:**
   >
   > To execute a specific statement, select the statement you want to run and click **Run**.

   The results appear in the Results pane.
4. To export the resulting report as a comma-delimited file (.csv), click the **Download** link.

3.4.2 About Transactions in SQL Commands

Transactional mode is a stateful transaction mode where you can, for example, perform an update, select data for review, and `COMMIT` or `ROLLBACK` changes. It is implemented using `DBMS_JOBS`.

Consider the following behavior in transactional mode:

- Actions are not committed to the database until you enter an explicit `COMMIT` command.
- Exiting SQL Commands terminates and rolls back the current transaction.
- A session timeout terminates and rolls back the current transaction.

  Note that the Environment Setting, `SQL Commands Maximum Inactivity in minutes`, sets the time before an inactive session times out. The default timeout is 60 minutes.
- The **CSV Export** option is not available.

3.4.3 Disabling Transactional SQL Commands

Check the Autocommit check box to disable transactional SQL commands.

To disable transactional SQL commands in SQL Commands, check the Autocommit check box. Attempting to use any transactional SQL commands such as `COMMIT` or `ROLLBACK` when transactional mode is disabled returns an error message.

3.4.4 Enabling Transactional SQL Commands

Uncheck the Autocommit check box to enable the transactional SQL commands.

To enable transactional SQL commands, clear the Autocommit check box. Oracle Application Express verifies that the necessary system resources are available before entering the transactional mode. If resources are unavailable, an error message is displayed.

3.4.5 About Unsupported SQL*Plus Commands

SQL Commands does not support SQL*Plus commands.

If you attempt to enter an SQL Command Line command such as `SET ECHO` or `DEFINE` in SQL Commands, an error message displays.

3.4.6 Terminating a Command

You can terminate a command in SQL Commands using a semicolon (;), a slash (/), or with nothing.

Consider the following valid alternatives:

```
SELECT * from emp;

SELECT * from emp
/

SELECT * from emp
```

The first example demonstrates the use of a semicolon (;), the second example demonstrates using a slash (/), and the final example demonstrates a command with no termination.

3.4.7 Using Bind Variables

Enter values for bind variables during command execution.

Bind variables are supported. You are prompted to enter values for bind variables during command execution. Bind variables are prefixed with a colon.

For example

```
SELECT * FROM emp WHERE deptno = :dept
```

3.4.8 Using the Find Tables Button

Use the Find Tables button to view tables within the currently selected schema.

To view tables within the current schema:

1. Navigate to SQL Commands.
2. From the Schema list, select a schema (optional).
3. Click **Find Tables**.

The Table Finder appears. A search bar displaying the selected schema displays at the top of the page and contains the following controls:

- **Search**. Search for a table name. Enter case insensitive keywords in the Search field and click **Go**. To view all tables, leave the Search field blank and click **Go**.
- **Views**. Select the **Views** check box and click **Go** to include views in the resulting report.
- **Rows**. Determine how many rows display in the resulting report. To change the number of rows that display, make a selection from the list and click **Go**.

The Table Finder report appears displaying the table name, the number of rows, last analyzed date, and the object type.

4. Select a table name.

   The Table Finder report appears.

   This report displays the column names, data type, length, precision, scale and the SQL necessary to re-create the table that appears at the bottom of the page.

5. Click **Append** to append SQL from the page to the existing command in the command editor.
6. Click **Replace** to replace existing command in the command editor with the SQL from the page.

3.5 Saving an SQL Command

You can save commands you enter in SQL Commands.

To save an SQL command:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.

2. Enter the command in the command editor.
3. Click **Save** to save the command.

   You are prompted to enter a name and description for the command.

4. Click **Save**, or click **Cancel** to return to the command editor without saving.

   The saved command is listed in the display area.

3.6 Copying a Command

Click Saved SQL from the SQL Commands page, click the name of the command to load into command editor, enter a name, and click Save.

To copy a SQL command:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.

2. Click **Saved SQL**.

   The Saved SQL list of commands appears in the display pane.

3. Click the name of the command to load it into the command editor
4. Click **Save** to save the command.
5. Enter a name for the command in the Name field and click **Save**.

   The command is copied to the new name.

3.7 Viewing Results

When you execute a SQL command, the results display in the display pane at the bottom of the SQL Commands home page.

The results of the last executed command are available until you execute another SQL command, or leave SQL Commands.

- Accessing the Results Pane (page 3-9)
  From the SQL Commands page, click the Results tab.
- About the Results Pane (page 3-9)
  The Results pane displays SQL command results as HTML formatted table. The number of rows returned appears after the output, and the time taken. DBMS output appears as plain text after the HTML formatted results.

3.7.1 Accessing the Results Pane

From the SQL Commands page, click the Results tab.

To display SQL command results:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. Click the **Results** tab.

   The HTML formatted results appear in the display pane.
3. Click **DBMS Output** to display plain text DBMS output results.

   The **DBMS Output** control only appears if there are DBMS output results.

3.7.2 About the Results Pane

The Results pane displays SQL command results as HTML formatted table. The number of rows returned appears after the output, and the time taken. DBMS output appears as plain text after the HTML formatted results.

On the Results pane you can:

- **Display DBMS output.** Automatically displays HTML output or DBMS output depending on type of SQL command entered, SQL or PL/SQL block with DBMS output.
- **Download results.** Click **Download** to export results to a comma-delimited file on your local file system. You are prompted to enter a name and directory for the file. Only available for HTML output.

3.8 Using Explain Plan

Explain plan parses a query and records the plan that Oracle devises to execute it.

By examining this plan, you can find out if Oracle is picking the right indexes and joining your tables in the most efficient manner. It is not necessary to execute the command to view the explain plan.

- Viewing an Explain Plan (page 3-10)
  From the SQL Commands page, enter or load the command whose plan you want to view and click the Explain tab.

3.8.1 Viewing an Explain Plan

From the SQL Commands page, enter or load the command whose plan you want to view and click the Explain tab.

To view the Explain Plan:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. Enter or load the command whose plan you want to view.
3. Click the **Explain** tab.

   The explain plan used by the optimizer appears in the display pane.

   The Explain Plan pane shows the plan used by the Oracle Optimizer to run your SQL command. It typically displays the Query Plan, Index Columns and Table Columns used.

   On the Explain Plan pane you can:

   - **View object definitions**. Click the object name in Query Plan to display the object definition in the Object Browser.
   - **View index definitions**. Click the index name in Table Columns to display the index definition in the Object Browser.

3.9 Using Saved Commands

You can access the commands you save and commands saved by other users in the same workspace.

You can also access SQL commands you and other users of the same workspace saved from the Query Builder.

- Accessing Saved Commands (page 3-11)
  From the SQL Commands page, click the Saved SQL tab, click the title of the command to load it into the command editor, and click Run.
- About the Saved SQL Pane (page 3-11)
  The Saved SQL pane displays a list of all commands saved under the current workspace.

3.9.1 Accessing Saved Commands

From the SQL Commands page, click the Saved SQL tab, click the title of the command to load it into the command editor, and click Run.

To access saved SQL commands:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. Click the **Saved SQL** tab.

   The Saved SQL list of commands appears in the display pane.
3. Click the title of the command to load it into the command editor.

   The command appears in the editor.
4. Click **Run** to execute the command.

3.9.2 About the Saved SQL Pane

The Saved SQL pane displays a list of all commands saved under the current workspace.

The list displays commands saved from SQL Commands and SQL commands saved from Query Builder. Saved SQL commands must have unique names in the current workspace. The same name cannot be used in the Query Builder and SQL Commands.

Each command entry shows the owner name, the command name, a description if it exists, the first characters of the SQL command, who last updated the command and when.

On the Saved SQL pane you can:

- **Show commands by owner** - Make a selection from the Owner list to specify the user whose commands you want to display. To view all scripts select -All Users-.
- **Search for a command** - Enter a command name or partial name, or enter a code snippet in the Find field and click **Go**. To view all scripts, leave the Find field blank and click **Go**. You control how many rows display by making a selection from the Rows list.
- **Set the Number of Output Rows** - Make a selection from the Display list to specify the number of Saved SQL commands to display simultaneously.
- **Delete a command** - Click the check box associated with a command and click **Delete Checked**.

3.10 Using SQL Command History

You use SQL Command History to access commands you have executed in SQL Commands.

Commands you have executed are stored in the command history regardless of whether you explicitly save them.

- Accessing a Command from Command History (page 3-12)
  From the SQL Commands page, click the History tab.
- About the History Pane (page 3-12)
  The History pane displays a list of commands you have executed.

3.10.1 Accessing a Command from Command History

From the SQL Commands page, click the History tab.

To access history commands:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.

   The SQL Commands page appears.
2. Click the **History** tab.

   The list of commands in History appears in the display pane.
3. Click the partial command displayed in the SQL column.

   The command appears in the editor.

3.10.2 About the History Pane

The History pane displays a list of commands you have executed.

Each history entry shows the time the command was last executed, the first characters of the command, and the schema in which it was executed.

On the History pane you can:

- **Load a command** - Click the partial command displayed in the SQL column to load the command into the command editor. When the command loads, it also sets the schema in which it was last executed.

4

Using Oracle Application Express Utilities

Oracle Application Express provides utilities to build SQL queries, load and unload data from an Oracle database, generate DDL, view object reports, manage User Interface Defaults, restore dropped database objects, compare schemas, monitor the database, and view database details.

- Using Data Workshop (page 4-2)
  Data Workshop enables you to load and unload data to the database.
- Using Query Builder (page 4-10)
  Using Query Builder, you can search and filter database objects, select objects and columns, create relationships between objects, view formatted query results, and save queries with little or no SQL knowledge.
- Using Quick SQL (page 4-22)
  Rapidly design and prototype data models using a markdown-like shorthand syntax that expands to standards-based Oracle SQL.
- Using Sample Datasets (page 4-38)
  Use Sample datasets to build sample applications.
- Generating DDL (page 4-42)
  With Oracle Application Express, you can generate data definition language (DDL) statements from the Oracle data dictionary.
- Managing User Interface Defaults (page 4-42)
  Managing user interface defaults enables you to assign default interface properties, create default wizard, modify the table user interface defaults, add new attributes, and export the user interface defaults.
- Comparing Schemas (page 4-46)
  You can run a report that compares database objects in two schemas, displaying differences between them. You can compare all objects in the schemas or limit your report to specific objects.
- Managing Methods on Tables (page 4-47)
  The Managing Methods on Tables utility enables you to create an application programming interface, or package, based on a specified table.
- Using the Recycle Bin (page 4-47)
  You can use the Recycle Bin to view and restore dropped database objects.
- Viewing Object Reports (page 4-48)
  SQL Workshop Utilities includes a variety of object reports to help you better manage the objects in your database.
- Viewing Database Details (page 4-53)
  If you have an account that has been granted a database administrator role, you can view details about your database on the About Database page.

- Monitoring the Database (page 4-53)
  The reports available on the Database Monitor page provide a database-wide view of the database sessions, system statistics, SQL statements, and longer operations.

4.1 Using Data Workshop

Data Workshop enables you to load and unload data to the database.

- About the Data Load and Unload Wizards (page 4-2)
  Use the Data Load and Data Unload wizards to load and unload delimited text data to and from the database.
- About Importing, Exporting, Loading, and Unloading Data (page 4-3)
  You have several options when copying data between Oracle databases or between an Oracle database and external files. Data copying is accomplished by *exporting* and *importing* data, and by *unloading* and *loading* data.
- Import/Export/Load/Unload Options (page 4-3)
  The Oracle Database and Oracle Application Express provide several powerful options for importing, exporting, loading, and unloading data.
- Accessing the Data Load/Unload Page (page 4-5)
  Access the Data Load and Unload page by clicking Utilities and then Data Workshop.
- Loading Data (page 4-5)
  Use the Load Data Wizard to copy and paste or upload data.
- Unloading Data (page 4-8)
  Use the Unload page to export the contents of a table to a text file or XML document.
- Using the Repository (page 4-10)
  Loaded text data files and spreadsheets are stored in the Repository. You can select the appropriate repository.

4.1.1 About the Data Load and Unload Wizards

Use the Data Load and Data Unload wizards to load and unload delimited text data to and from the database.

The step-by-step wizards have the following features:

- Load or unload XML files or delimited-field text files (such as comma-delimited (`.csv`) or tab-delimited files).
- Load by copying and pasting from a spreadsheet.
- Omit (skip) columns when loading or unloading.
- Load into an existing table or create a new table from the loaded data.
- When loading into a new table, the primary key can be taken from the data or generated from a new or existing Oracle sequence.
- When loading into a new table, column names can be taken from the loaded data.
- Each time that you load from a file, file details are saved in a Text Data Load Repository. You can access these files from within the repository at any time.

Limitations include the following:

- The wizards load and unload table data only. They do not load or unload other kinds of schema objects.
- You can load and unload to and from your own schema only. This is also true for users with administrator privileges.
- You can load or unload only a single table at a time.
- There are no data type limitations for unloading to text or XML files, or for loading from XML files. However, when loading from spreadsheets (through copy and paste) or from text files, only the following data types are supported: `NUMBER`, `DATE`, `VARCHAR2`, `CLOB`, `BINARY_FLOAT`, and `BINARY_DOUBLE`.

Supported unload formats include:

- Text such as comma-delimited or tab-delimited data
- XML documents

4.1.2 About Importing, Exporting, Loading, and Unloading Data

You have several options when copying data between Oracle databases or between an Oracle database and external files. Data copying is accomplished by *exporting* and *importing* data, and by *unloading* and *loading* data.

The following table defines these terms.

Term	Definition
Exporting	Copying database data to external files for import into another Oracle database only. The files are in a proprietary binary format.
Importing	Copying data into the database from external files that were created by exporting from another Oracle database.
Unloading	Copying database data to external text files for consumption by another Oracle database or another application (such as a spreadsheet application). The text files are in an industry-standard format such as tab-delimited or comma-delimited (`CSV`).
Loading	Copying data into the database from external text files that are in either a standard delimited format or in any of the formats that are supported by the Oracle SQL*Loader utility.

You can export data from any Oracle Database edition (Express Edition, Standard Edition, and Enterprise Edition) into any other edition.

4.1.3 Import/Export/Load/Unload Options

The Oracle Database and Oracle Application Express provide several powerful options for importing, exporting, loading, and unloading data.

Table 4-1 (page 4-4) provides a summary of these options.

Table 4-1 Summary of Oracle Application Express Import/Export Options

Feature or Utility	Description
Data Load/Unload wizards in Oracle Application Express	• Easy to use graphical interface • Loads/unloads from and to external text files (delimited fields) or XML files • Loads/unloads tables only, one table at a time • Access only to schema of logged-in user • No data filtering on upload
SQL*Loader utility	• Command-line interface, invoked with `sqlldr` command • Bulk-loads data into the database from external files • Supports numerous input formats, including delimited, fixed record, variable record, and stream • Loads multiple tables simultaneously • Powerful data filtering capabilities
Data Pump Export and Data Pump Import utilities	• Command-line interface, invoked with `expdp` and `impdp` commands • Exports and imports from one Oracle database to another (proprietary binary format) • Imports/exports all schema object types • Imports/exports entire database, entire schema, multiple schemas, multiple tablespaces, or multiple tables • Powerful data filtering capabilities • High speed
Export and Import utilities	• Command-line interface, invoked with `exp` and `imp` commands • Exports and imports from one Oracle database to another (proprietary binary format) • Supports XMLType data • Does not support the `FLOAT` and `DOUBLE` data types • Capabilities similar to Data Pump; Data Pump is preferred

Table 4-2 (page 4-4) provides several load, unload, import, and export scenarios and suggests the appropriate option to use for each.

Table 4-2 Import/Export Scenarios and Recommended Options

Import/Export Scenario	Recommended Option
You have fewer than 10 tables to load, the data is in spreadsheets or tab-delimited or comma-delimited text files, and there are no complex data types (such as objects or multivalued fields).	Data Load/Unload wizards in Oracle Application Express
You have to load data that is not delimited. The records are fixed length, and field definitions depend on column positions.	SQL*Loader

Table 4-2 (Cont.) Import/Export Scenarios and Recommended Options

Import/Export Scenario	Recommended Option
You have tab-delimited text data to load, and there are more than 10 tables.	SQL*Loader
You have text data to load, and you want to load only records that meet certain selection criteria (for example, only records for employees in department number 3001).	SQL*Loader
You want to import or export an entire schema from or to another Oracle database.	Data Pump Export and Data Pump Import
You want to import or export data from or to another Oracle database. The data contains XMLType data and contains no `FLOAT` or `DOUBLE` data types.	Import (`imp`) and Export (`exp`)

4.1.4 Accessing the Data Load/Unload Page

Access the Data Load and Unload page by clicking Utilities and then Data Workshop.

To access the Data Load/Unload page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Data Workshop**.

   The Data workshop page appears.
4. Click the appropriate icon to load data, unload data, or view the repository.

4.1.5 Loading Data

Use the Load Data Wizard to copy and paste or upload data.

- About Loading Data (page 4-5)
  You can load data into the Oracle Application Express database in several ways.
- Loading a Text File or Spreadsheet Data (page 4-6)
  You can upload a Text file or copy and paste tab-delimited data directly into the Load Data Wizard.
- Loading an XML Document (page 4-8)
  Load an XML document using Load XML Data Wizard.

> **See Also:**
>
> "Import/Export/Load/Unload Options (page 4-3)"

4.1.5.1 About Loading Data

You can load data into the Oracle Application Express database in several ways.

- Copy and paste data from a spreadsheet.
- Upload a spreadsheet file in a delimited format (such as comma-delimited (.csv) or tab-delimited).

> **Note:**
>
> Upon selecting the file to be uploaded, a data sample of the first 5 lines of the file is displayed in the preview window. This is active by default and works for every chosen file. You can see the data and adjust the separator and enclosed by characters before actually uploading the data. You can do the column mapping in the next dialog box.

- Upload a text file containing comma-delimited or tab-delimited data.
- Upload an XML document.

4.1.5.2 Loading a Text File or Spreadsheet Data

You can upload a Text file or copy and paste tab-delimited data directly into the Load Data Wizard.

To load data from a text file:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Data Workshop**.
4. Under Data Load, click **Text Data**.

   The Load Data Wizard appears.
5. Under Load To, select either **Existing table** or **New table**.
6. Under Load From, select either **Upload file** or **Copy and paste**.
   - **Upload file** - Select to specify an existing text file to upload.
   - **Copy and paste** - Select to create a text file to upload by performing a copy and paste.
7. Click **Next**.
8. If you selected Upload file, under Load Data, make the following changes:
   - **Text File** - Browse for the name of the Text File to load.
   - **Separator** - Enter the column separator character. See field-level help for more details.
   - **Optionally Enclosed By** - If your data contains a delimiter character, enter it here. See field-level help for more details.
   - **First row contains column names** - Check this box if the first row of your data contains column names.
   - **File Character Set** - Select the character set in which your text file is encoded. See field-level help for more details.

9. If you selected Copy and paste, under Data make these changes:
   - **Data** - Enter the text file to upload.
   - **First row contains column names** - Check this box if the first row of your data contains column names.
10. Under Globalization, make the following changes:
   - **Currency Symbol** - If your data contains an international currency symbol, enter it here.
   - **Group Separator** - If your data contains a character that separates integer groups, for example to show thousands and millions, enter it here.
   - **Decimal Character** - If your data contains a character that separates the integer and decimal parts of a number, enter it here.

   > **Tip:**
   >
   > See field-level help for more details about a specific attribute.

11. Click **Next**.

    The table Properties page appears. Before performing the data upload, this page allows you to optionally customize the data.
12. For Table Properties, make the following changes and click **Next**:
   - **Schema** - Select the schema containing the table to load.
   - Table Name - Identify the table name you want to create. By default, all table names are converted to upper case. Select **Preserve Case** to override this default behavior.
   - **Column Names** - Enter the names of the columns.
   - **Data Type** - Select the data type for this column.
   - **Format** - Enter the format type.
   - **Description** - Enter the description.
   - **Column Length** - Enter the length of the data in this column.
   - **Upload** - Select **Yes** to upload data for this column. Select **No** to not upload data for this column and skip to the next column.
13. For Primary Key, make the following changes:
   - **Primary Key From** - Define the primary key column by choosing either an existing column or creating a new column. Options include:
     - **Use an existing column** - If you choose an existing column you must select the column from the columns being loaded.
     - **Create new column** - If you choose to create a new column, you must specify the name of the new column.
   - **Constraint Name** - The constraint name defaults to the table name appended with _PK. You can update the name if desired.
   - **Primary Key Population** - You must decide how your primary key is populated. You either generate it from a new sequence, from a new sequence or not at all. Options include:

- **Generated from a new sequence** - If you choose a new sequence, you can use the sequence name given or update it as necessary. The sequence is created for you.
- **Generated from an existing sequence** - If you choose an existing sequence, you must select the sequence from those currently defined in your schema.

  If you generate the primary key from a sequence, either new or existing, the wizard creates a trigger on the table to retrieve the next sequence value and populate the primary key when a record is inserted.
- **Not generated**

14. Click **Load Data**.

4.1.5.3 Loading an XML Document

Load an XML document using Load XML Data Wizard.

Oracle Application Express supports XML documents in Oracle's canonical XML format.

In Oracle's canonical XML format, each element represents a column value, each element is named after the column, all elements that are part of the same row are children of a <ROW> element, and all <ROW> elements are children of a <ROWSET> element.

To load an XML document:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Data Workshop**.
4. Click **XML Data**.

   The Load XML Data Wizard appears.
5. Select the appropriate schema for your data and click **Next**.
6. Select the table into which to load the data.
7. Follow the on-screen instructions.

4.1.6 Unloading Data

Use the Unload page to export the contents of a table to a text file or XML document.

- Unloading a Text File (page 4-9)
  Use the Unload to Text Wizard to export the contents of a table to a text file.
- Unloading to an XML Document (page 4-10)
  Use the Unload to XML Wizard to export the contents of a table to an XML document adhering to the Canonical XML specification.

> **See Also:**
>
> "Import/Export/Load/Unload Options (page 4-3)"

4.1.6.1 Unloading a Text File

Use the Unload to Text Wizard to export the contents of a table to a text file.

For example, you could export an entire table to a comma-delimited file (.csv).

To unload a table to a text file:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities.**
3. Click **Data Workshop**.
4. Under Data Unload, click **To Text**.

   The Unload to Text Wizard appears.
5. Select the appropriate schema.
6. Select the table to unload.
7. Highlight the columns to include, optionally enter a Where Clause to limit the records unloaded and click **Next**.
8. For Options, enter the following:
   - **Separator** - Enter the type of separator used to separate column values.
   - **Optionally Enclosed By** - If your data contains a delimiter character, enter it here.
   - **Include Column Names** - Check this box to include the name of each column in the text file.
   - **File Format** - Select the appropriate file format.
   - **File Character Set** - Select the character set in which your text file is encoded.

   > **Tip:**
   >
   > See field-level help for more details.
9. Click **Unload Data**.

   The File Download window appears.
10. Click **Save** to download the file.

4.1.6.2 Unloading to an XML Document

Use the Unload to XML Wizard to export the contents of a table to an XML document adhering to the Canonical XML specification.

To unload a table to an XML document:

1. On the Workspace home page, click the **Utilities** icon.
2. Click **Data Workshop**.
3. Under data Unload, click **to XML**.

   The Unload to XML Wizard appears.
4. Select the appropriate schema.
5. Select the table to unload.
6. Highlight the columns to include.
7. Optionally enter a Where Clause to limit the records unloaded, and click **Unload Data.**

   A browser displays the XML data.

4.1.7 Using the Repository

Loaded text data files and spreadsheets are stored in the Repository. You can select the appropriate repository.

To access the different repositories:

1. On the Workspace home page, click the **Utilities** icon.
2. Click **Data Workshop**.
3. Under Repository, select one of the following:
   - **Import Repository** - Displays the status of loaded Text data. Details include file name, imported by, imported on, schema, table, bytes, and the number of rows that were loaded successfully and the number were not.
   - **Spreadsheet Imports** - Displays the status of loaded spreadsheet data. Details include file name, imported by, imported on, schema, table, and the number of rows that were loaded successfully and the number were not.
4. To filter the display, make a selection from the Show list and click **Go**.
5. To view information about a specific file, click the **View** icon.
6. To delete an imported file, select it and click **Delete Checked**.

4.2 Using Query Builder

Using Query Builder, you can search and filter database objects, select objects and columns, create relationships between objects, view formatted query results, and save queries with little or no SQL knowledge.

Query Builder's graphical user interface enables database developers to build SQL queries without the need for manual SQL coding.

> **Tip:**
>
> SQL commands created in the Query Builder can be accessed from SQL Commands.

- Query Builder Home Page (page 4-12)
  Learn about Query Builder home page.
- Accessing Query Builder (page 4-12)
  Learn how to access Query Builder.
- Understanding the Query Building Process (page 4-13)
  Learn how to build a query in Query Builder.
- Using the Object Selection Pane (page 4-13)
  Use the Object Selection pane to search for and view tables, views, and materialized views within the current schema as well as select objects.
- Using the Design Pane (page 4-14)
  Learn about how to use the Design pane.
- Specifying Query Conditions (page 4-17)
  As you select columns within an object, you can specify conditions on the Conditions tab.
- Creating Relationships Between Objects (page 4-18)
  You can create relationships between objects by creating a join.
- Working with Saved Queries (page 4-20)
  You can create queries and save the query for later use.
- Viewing Generated SQL (page 4-22)
  The SQL view presents a read-only, formatted representation of the SQL generated by Query Builder. You can copy the SQL code that appears in the SQL View for use in other tools such as SQL Command Processor or App Builder.
- Viewing Query Results (page 4-22)
  Execute a query, once you select objects and determine what columns to include in your query.

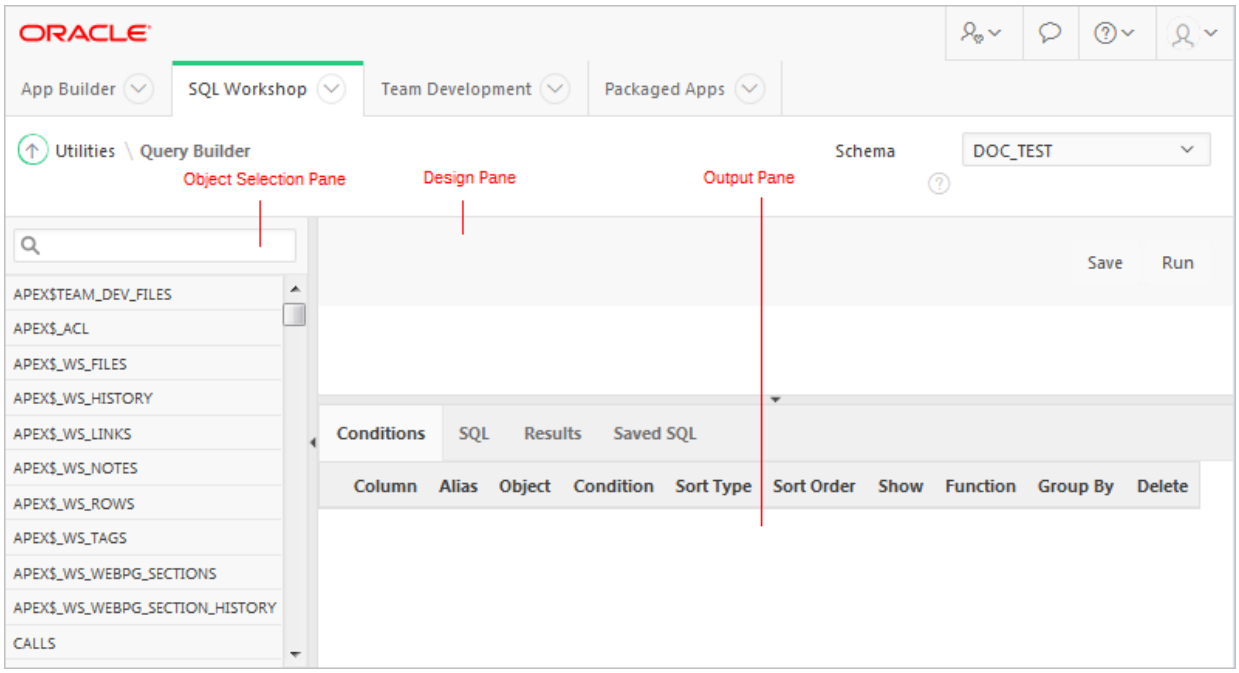
> **See Also:**
>
> - " Using SQL Commands (page 3-1)"
> - "Using the Query Finder" in *Oracle Application Express App Builder User's Guide*

4.2.1 Query Builder Home Page

Learn about Query Builder home page.

The Query Builder page is divided into three sections:

- **Object Selection pane** displays on the left side of the page and contains a list objects from which you can build queries. Only objects in the current schema display.
- **Design pane** displays to the right of the Object Selection pane and above the Conditions, SQL, Results, and Saved SQL tabs. When you select an object from the Object Selection pane, it appears in the Design pane.
- **Output pane** displays below the Design pane. Once you select objects and columns, you can create conditions, view the generated SQL, or view query results.



4.2.2 Accessing Query Builder

Learn how to access Query Builder.

To access Query Builder:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Query Builder**.

The Query Builder home page appears.

> **Tip:**
>
> You can also access Query Builder from a code editor in many part of Oracle Application Express by clicking the Query Builder button when editing a SQL attribute.

4.2.3 Understanding the Query Building Process

Learn how to build a query in Query Builder.

To build a query in Query Builder, you perform the following steps:

1. Select objects from the Object Selection pane.
2. Add objects to the Design pane and select columns.
3. **Optional**: Establish relationships between objects.
4. **Optional**: Create query conditions.
5. Execute the query and view results.

4.2.4 Using the Object Selection Pane

Use the Object Selection pane to search for and view tables, views, and materialized views within the current schema as well as select objects.

- Searching and Filtering Objects (page 4-13)
  Use the Object Selection pane to search for and view tables, views, and materialized views within the current schema.
- Hiding and Restoring the Object Selection Pane (page 4-14)
  You can hide the Object Selection pane by selecting the Collapse control.
- Selecting Objects (page 4-14)
  Use the Object Selection pane to select objects.

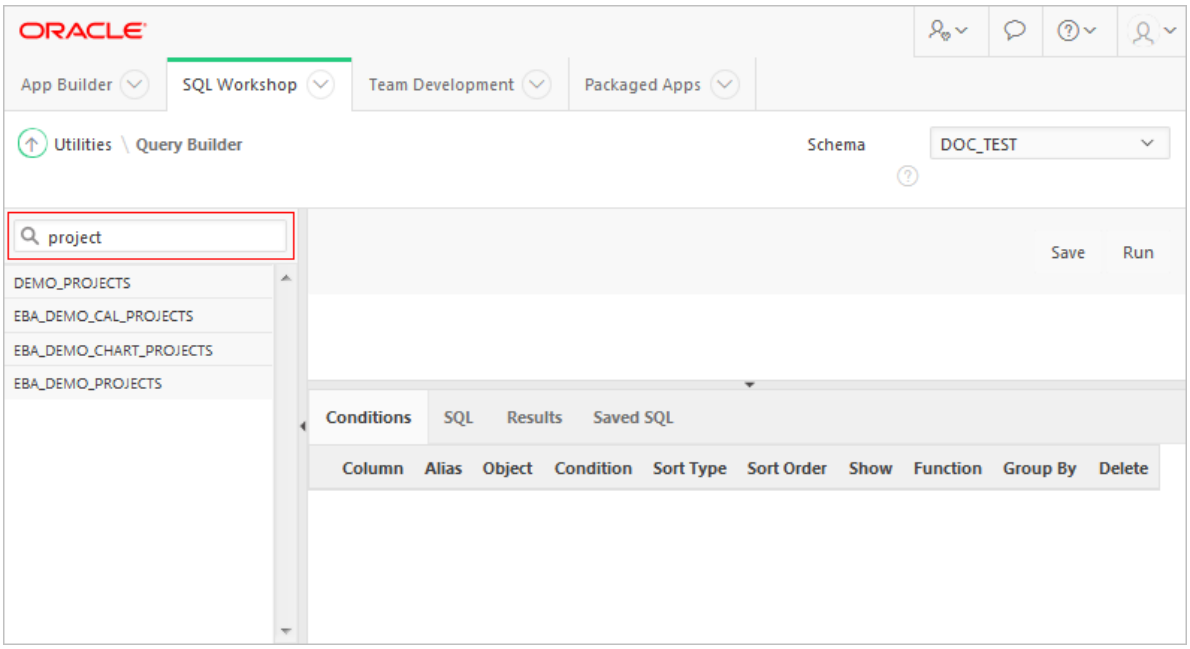
4.2.4.1 Searching and Filtering Objects

Use the Object Selection pane to search for and view tables, views, and materialized views within the current schema.

To search or filter objects:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.
2. In the search field at the top of the Object Selection pane, enter a case insensitive query.



3. To view all tables or views within the currently selected schema, leave the search field blank.

4.2.4.2 Hiding and Restoring the Object Selection Pane

You can hide the Object Selection pane by selecting the Collapse control.

The Collapse control displays on the right side of the Object Selection pane. If the Object Selection pane displays, selecting this control hides it. Similarly, if the Object Selection pane is hidden, selecting this control causes the pane restores it. Select the Left Splitter (click above or below the Collapse control to manually resize the Object Selection pane.

4.2.4.3 Selecting Objects

Use the Object Selection pane to select objects.

The Design pane displays to the right of the Object Selection pane. When you select an object from the Object Selection pane, it appears in the Design pane. You use the Object Selection pane to select objects (that is, tables, views, and materialized views) and the Design pane to identify how those selected objects are used in a query.

4.2.5 Using the Design Pane

Learn about how to use the Design pane.

Information on how to use how to use the Design pane includes the following topics:

- About Supported Column Types (page 4-15)
  Columns of all types available in Oracle Database 10*g* Release (10.2) or later display as objects in the Design pane.
- Adding an Object to the Design Pane (page 4-15)
  You add an object to the Design pane by selecting it from the Object Selection pane.

- Resizing the Design and Results Panes (page 4-16)
  As you select objects, you can resize Design and Results panes by selecting the Bottom Splitter in the center of the page.
- Removing or Hiding Objects in the Design Pane (page 4-16)
  You remove or hide objects in the Design pane by selecting controls at the top of the object.

4.2.5.1 About Supported Column Types

Columns of all types available in Oracle Database 10*g* Release (10.2) or later display as objects in the Design pane.

Note the following column restrictions:

- You may only select a maximum of 60 columns for each query.
- The following column types are not selectable and cannot be included in a generated query:
  - `BLOB`
  - `NCLOB`
  - `RAW`
  - `LONG`
  - `LONG RAW`
  - `XMLType`
  - Any other nonscalar column types

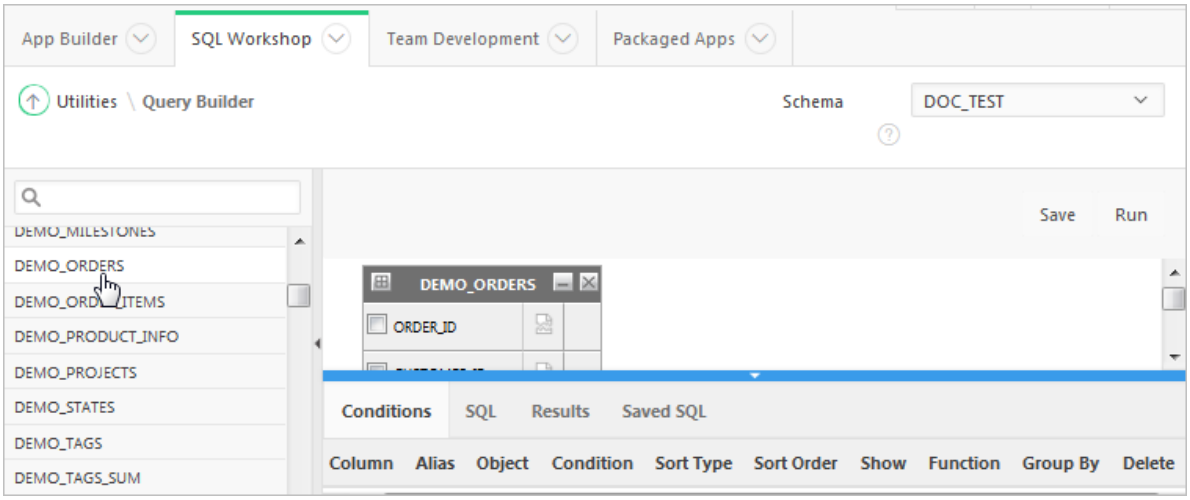
4.2.5.2 Adding an Object to the Design Pane

You add an object to the Design pane by selecting it from the Object Selection pane.

To add an object to the Design pane:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.
2. Select an object from the Object Selection pane.



The selected object appears in the Design Pane. The icon next to the column name represents the type of the column.

3. Select the columns to be included in your query by clicking the check box to the left of the column name.

   When you select a column you are indicating it is included in the query. As you select a column, it appears on the Conditions tab. Note that the Show check box on the Conditions tab controls whether a column is included in query results. By default, this check box is selected.

   To select the first twenty columns, click the **Table Actions** icon in the upper left corner of the object. The Actions window appears. Select **Check All**.

4. To execute the query and view results, click **Run** (**Ctrl+Enter**).

   The Results pane displays the query results.

4.2.5.3 Resizing the Design and Results Panes

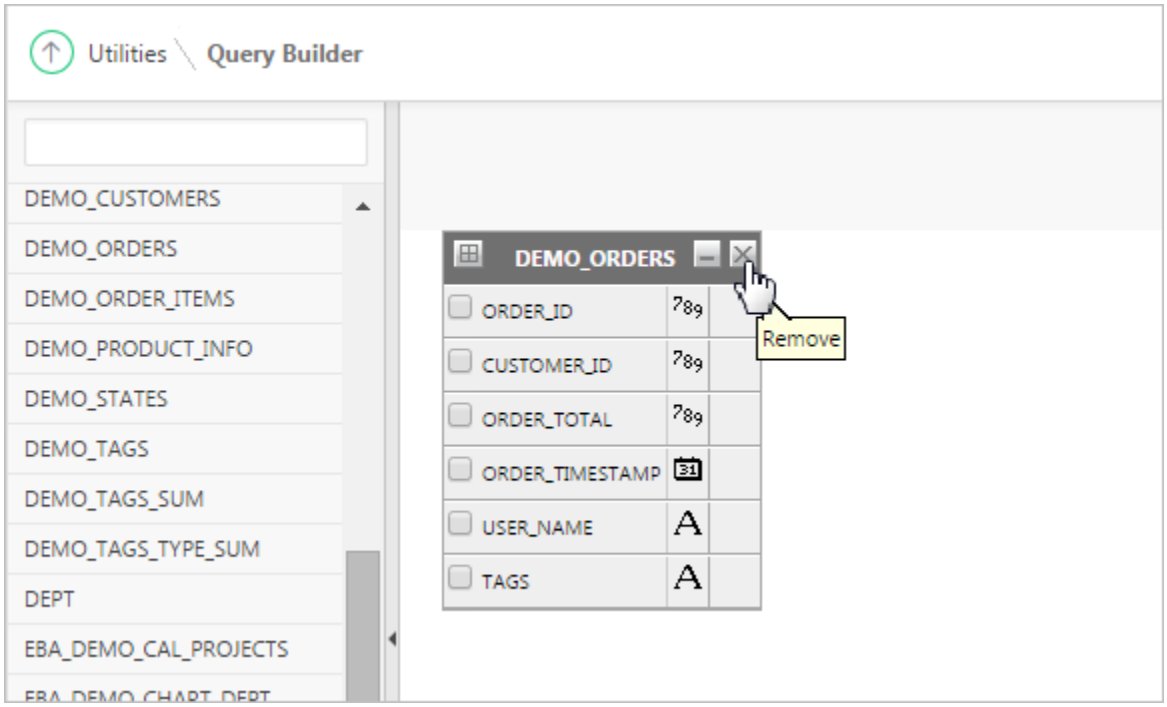
As you select objects, you can resize Design and Results panes by selecting the Bottom Splitter in the center of the page.

The Bottom Splitter resembles a gray horizontal rule in the center of the page. Move the Bottom Splitter up and down to shrink and expand the Design pane.

4.2.5.4 Removing or Hiding Objects in the Design Pane

You remove or hide objects in the Design pane by selecting controls at the top of the object.

To remove an object, select the **Remove** icon in the upper right corner. To temporarily hide the columns within an object, click the **Show/Hide Columns** icon.



4.2.6 Specifying Query Conditions

As you select columns within an object, you can specify conditions on the Conditions tab.

Conditions enable you to filter and identify the data you want to work with. You can use these attributes to modify the column alias, apply column conditions, sort columns, or apply functions.

To specify query conditions:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.

2. Select an object from the Object Selection pane.

   The selected object appears in the Design Pane.

3. Select the columns to be included in your query by clicking the box to the left of the column name.

   When you select a column, you are indicating you want to include it in your query. As you select each column, it appears as a separate row in the Conditions view. Table 4-3 (page 4-17) describes the attributes available on the Conditions tab.

Table 4-3 Conditions Tab

Condition Attribute	Description
Up and Down Arrows	Controls the order that the columns to be displayed in the resulting query. Click the arrow buttons to move columns up and down.
Column	Displays the column name.
Alias	Specify an optional column alias. An **alias** is an alternative column name. Aliases make a column name more descriptive, shorten the column name, or prevent possible ambiguous references.
Object	Specifies table or view name.

Condition	Specify a condition for the column.
	The condition you enter modifies the query's `WHERE` clause. When specifying a column condition, you must include the appropriate operator and operand. Consider the following examples:
	`>=10`
	`='VA'`
	`IN (SELECT dept_no FROM dept)`
	`BETWEEN SYSDATE AND SYSDATE + 15`

Sort Type	Select a sort type. Options include: • Ascending (`Asc`) • Descending (`Desc`)
Sort Order	Enter a number (1, 2, 3, and so on) to specify the order in which selected columns should display.

Table 4-3 (Cont.) Conditions Tab

Condition Attribute	Description
Show	Select this check box to include the column in your query results. It is not necessary to select Show to add a column to the query for filtering only. For example, suppose you want to create following query: `SELECT ename FROM emp WHERE deptno = 10` To create this query in Query Builder: a. From the Object list, select **emp**. b. In the Design Pane, select `ename` and `deptno`. c. For the deptno column, in Condition enter `=10` and deselect the Show check box.
Function	Select an argument function. Available functions are dependent on the column type and include: • **NUMBER columns** - `COUNT`, `COUNT DISTINCT`, `AVG`, `MAX`, `MIN`, `SUM` • **VARCHAR2, CHAR columns** - `COUNT`, `COUNT DISTINCT`, `INITCAP`, `LENGTH`, `LOWER`, `LTRIM`, `RTRIM`, `TRIM`, `UPPER` • **DATE, TIMESTAMP columns** - `COUNT`, `COUNT DISTINCT`, `TO_CHAR YEAR`, `TO_CHAR QUARTER`, `TO_CHAR MONTH`, `TO_CHAR DAY`, `TO_CHAR DAY OF YEAR`, `TO_CHAR WEEK`
Group By	Specify columns to be used for grouping when an aggregate function is used. Only applicable for columns included in output.
Delete	Deselect the column, excluding it from the query.

As you select columns and define conditions, Query Builder writes the SQL for you.

4. To view the underlying SQL, click the **SQL** tab.

4.2.7 Creating Relationships Between Objects

You can create relationships between objects by creating a join.

A **join** identifies a relationship between two or more tables, views, or materialized views.

- About Join Conditions (page 4-18)
  A join condition determines how the rows from one object combine with the rows from another object.
- Joining Objects Manually (page 4-19)
  You can create a join manually by selecting the Join column in the Design pane.
- Joining Objects Automatically (page 4-19)
  Learn how to join objects automatically.

4.2.7.1 About Join Conditions

A join condition determines how the rows from one object combine with the rows from another object.

When you write a join query, you specify a condition that conveys a relationship between two objects. This condition is called a **join condition**.

Query Builder supports inner, outer, left, and right joins. An **inner join** (also called a **simple join**) returns the rows that satisfy the join condition. An outer join extends the result of a simple join. An **outer join** returns all rows that satisfy the join condition and returns some or all of those rows from one table for which no rows from the other satisfy the join condition.

4.2.7.2 Joining Objects Manually

You can create a join manually by selecting the Join column in the Design pane.

To join two objects manually:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.

2. From the Object Selection pane, select the objects you want to join.

   The objects display in the Design pane.

3. Identify the columns you want to join.

   You create a join by selecting the Join column adjacent to the column name. The Join column displays to the right of the datatype, beneath the Remove icon. When your cursor is in the appropriate position, the following tooltip displays:

   ```
   Click here to select column for join
   ```

4. Select the appropriate Join column for the first object.

   When selected, the Join column displays as a dark gray. To deselect a Join column, simply select it again or press **ESC**.

5. Select the appropriate Join column for the second object.

   > **Tip:**
   >
   > You can also join two objects by dragging and dropping. Select a column in the first table and then drag and drop it onto a column in another table.

   When joined, a line connects the two columns.

6. Select the columns to be included in your query. You can view the SQL statement resulting from the join by positioning the cursor over the line.

7. Click **Run** to execute the query.

   The Results pane displays the query results.

4.2.7.3 Joining Objects Automatically

Learn how to join objects automatically.

When you join objects automatically, the Query Builder suggests logical, existing parent and child relationships between existing columns.

To join objects automatically:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.

2. From the Object Selection pane, select an object.

   The object displays in the Design pane.

3. Click the small icon in the upper left corner of the object. Depending upon the selected object, the icon label displays as **Table Actions** or **View Actions**.

   The Actions window appears. Use the Actions window to select all columns within the current object or objects related to the current object.

4. In the Actions window, select the appropriate options:

   - **Check All** - Select this option to select the first twenty columns in the current object.
   - **Add Parent** - Displays tables that are referenced as a foreign key to the current object.
   - **Add Child** - Displays tables that reference the current object in a foreign key.

   If using Add Parent or Add child, the selected object appears and a line connects the foreign key columns.

5. Select additional columns to be included in your query.

   You can view the SQL statement resulting from the join by positioning the cursor over the green line.

6. Click **Run** to execute the query.

   The Results pane displays the query results.

4.2.8 Working with Saved Queries

You can create queries and save the query for later use.

As you create queries, you can save them by clicking the Save button in the Design pane. Once you save a query, you can access it later in the Saved SQL view.

> **Tip:**
>
> The Saved SQL view only displays queries saved in Query Builder. In SQL Commands you can view queries saved in both SQL Commands and Query Builder.

- Saving a Query (page 4-21)
  Learn how to save a query.
- Editing a Saved Query (page 4-21)
  Once you save a query, you can access it in the Saved SQL view and edit.
- Deleting a Saved Query (page 4-21)
  Learn how to delete a saved SQL query.

4.2.8.1 Saving a Query

Learn how to save a query.

To save a query:

1. Build a query:
   a. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

      Query Builder appears.
   b. Select objects from the Object Selection pane.
   c. Add objects to the Design pane and select columns.
   d. Execute the query.
2. Click **Save**.
3. Enter a name and description and click **Save**.

   The saved query displays in the Saved SQL view.

Note that Query Builder does not support duplicate query names. If you open an existing query, keep the existing name, and save it again, Query Builder over-writes the existing query. If you change the name of an existing query and save it again, Query Builder saves the query again under the new name.

4.2.8.2 Editing a Saved Query

Once you save a query, you can access it in the Saved SQL view and edit.

To edit a Saved SQL query:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.
2. Select the **Saved SQL** tab.
3. To filter the display, you can:
   - Make a selection from the Owner list and click **Go**.
   - Enter a search query in the Name field and click **Go**.
4. To edit a query, select the appropriate name.

   The saved query appears. The selected objects display in the Design pane and the Conditions view appears.

4.2.8.3 Deleting a Saved Query

Learn how to delete a saved SQL query.

To delete a Saved SQL query:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.
2. Select the **Saved SQL** tab.

3. Select the queries to be deleted and click **Delete Checked**.

4.2.9 Viewing Generated SQL

The SQL view presents a read-only, formatted representation of the SQL generated by Query Builder. You can copy the SQL code that appears in the SQL View for use in other tools such as SQL Command Processor or App Builder.

To access the SQL view:

1. On the Workspace home page, click **SQL Workshop** and then **Query Builder**.

   Query Builder appears.
2. Select an object from the Object Selection pane.

   The selected object appears in the Design Pane.
3. Select the columns to be included in your query.
4. Click the **SQL** tab.

   The SQL code generated by Query Builder appears.

4.2.10 Viewing Query Results

Execute a query, once you select objects and determine what columns to include in your query.

Execute a query by:

- Clicking the **Run** button (or pressing **Ctrl+Enter**)
- Selecting the **Results** tab

The Results view appears, displaying formatted query results. To export the report as a comma-delimited file (.csv) file, click the **Download** link at the bottom of the page.

4.3 Using Quick SQL

Rapidly design and prototype data models using a markdown-like shorthand syntax that expands to standards-based Oracle SQL.

- Getting Started with Quick SQL (page 4-23)
- Using Quick SQL Help (page 4-24)
- Generating Quick SQL with SQL Shorthand (page 4-26)
- Saving Scripts (page 4-27)
- Creating an Application from a Generated SQL Script (page 4-29)
- Using Quick SQL Samples (page 4-31)
- Configuring Settings (page 4-32)
- About Database Terminology (page 4-34)
- Recommendations When Using Quick SQL (page 4-37)
- About Supported Data Types (page 4-37)

4.3.1 Getting Started with Quick SQL

- About Quick SQL (page 4-23)
  Quick SQL is designed to reduce the time and effort required to create SQL tables, triggers, and index structures.
- About Quick SQL Shorthand Syntax (page 4-23)
  Learn about the rules to use Quick SQL.
- Accessing Quick SQL (page 4-24)
  Access the Quick SQL page by clicking Utilities and then Quick SQL.

4.3.1.1 About Quick SQL

Quick SQL is designed to reduce the time and effort required to create SQL tables, triggers, and index structures.

Quick SQL is not designed to be a replacement for data modeling. It is simply a quick way to develop a script for simple tables and views. Once the SQL is generated it can be tweaked and expanded upon. Quick SQL provides a quick way to generate the SQL required to create a relational data model from an indented text document.

Use Cases:

- Quickly create robust data models
- Easily generate random data
- Learn SQL create table, select, insert, index, trigger, PL/SQL package, and view syntax using provided examples

4.3.1.2 About Quick SQL Shorthand Syntax

Learn about the rules to use Quick SQL.

Follow these rules to use Quick SQL syntax:

- Parent table names - Enter parent table names without any indention.
- Table column names - Enter table columns with a uniform indention of two or more spaces.
- Parent/child relationships - Create parent child relationships by indenting child tables under parent tables. Indent the child tables to the same level as the columns in the parent table.

> **Note:**
>
> As a best practice, do not specify primary keys, the system will generate them for you.

Changing the Columns Data Type

If you want a column of a different data Type, add a space `NUM`, `INT`, `DATE` to the end of a column name. Based on the English text contained in the column name, and in the

absence of any data Type specified, the system makes educated deductions for some column names.

For example, if the column name contains the word `DATE` then it will be a date data type, or if the column name contains the word `NUMBER` or `NUM` then it will be a number data type. If you want to specify a specific `VARCHAR2` length, enter `VCn` where `n` is the length of the `VARCHAR2`. Click **Help** and then **Data Types** to view the available syntax. Object names are automatically formatted, replacing spaces with underscores, so you can use spaces in column and table names.

About Settings

Click the **Settings** button to view many generation options. You can automatically add Who Columns, for example, who created or last updated a given row and when. You can add a row sequence column that automatically increments by one on each row update. A row sequence can be useful to simplify lost update detection. You can also generate a PL/SQL API which provides a package per table and provides procedures to query, insert, update, and delete rows. Some applications benefit by having each change captured in a history table, logging of old and new values, which is also an option.

4.3.1.3 Accessing Quick SQL

Access the Quick SQL page by clicking Utilities and then Quick SQL.

To access the Quick SQL page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.

> **Note:**
>
> You can also access Quick SQL through SQL Scripts page by clicking Quick SQL button on the SQL Scripts page.

> **See Also:**
>
> "About SQL Scripts (page 5-1)"

4.3.2 Using Quick SQL Help

- Accessing Quick SQL Help (page 4-25)
- About Quick SQL Help (page 4-25)

4.3.2.1 Accessing Quick SQL Help

To access the Quick SQL Help page:

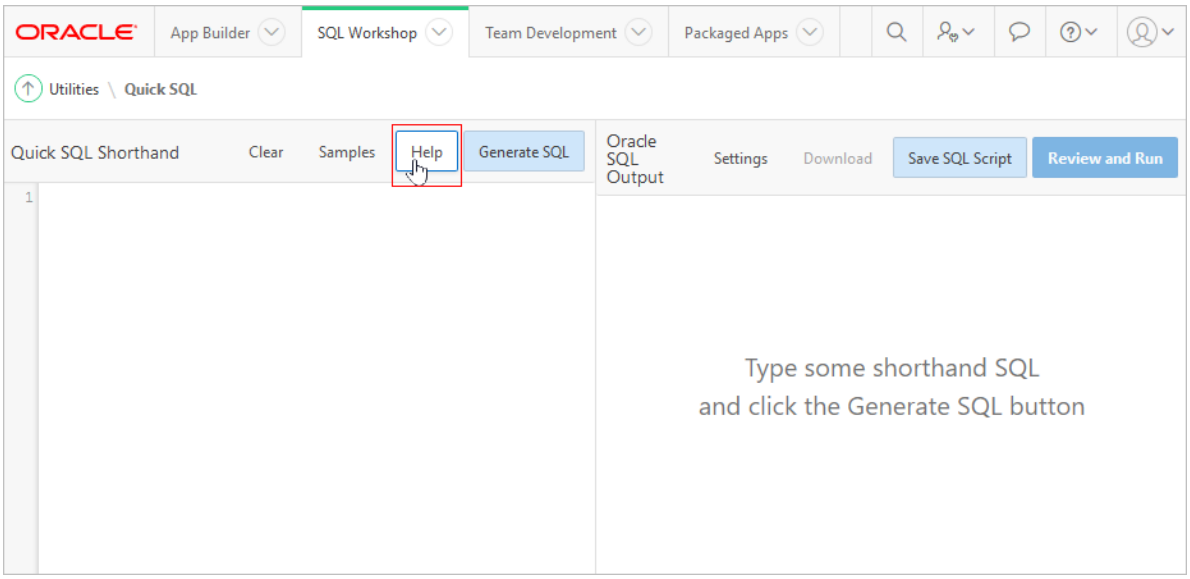
1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.

> **Note:**
>
> You can also access Quick SQL through SQL Scripts page by clicking Quick SQL button on the SQL Scripts page.

4. Click **Help** on the Quick SQL Shorthand pane.

   The Help page appears.



4.3.2.2 About Quick SQL Help

Quick SQL Help has the following:

- **About** - Overview of Quick SQL together with tips on getting started and steps to create the DB objects.
- **Data Types** - List of data types that can applied to each column.
- **Table Directives** - Commands used to modify tables.
- **Column Directives** - Commands used to extend column definitions.
- **Views** - Syntax for defining a view.

- **Settings** - Inline settings to explicitly set SQL syntax generation options.
- **Samples** - Samples of data models that can be used to populate the Quick SQL Shorthand.

4.3.3 Generating Quick SQL with SQL Shorthand

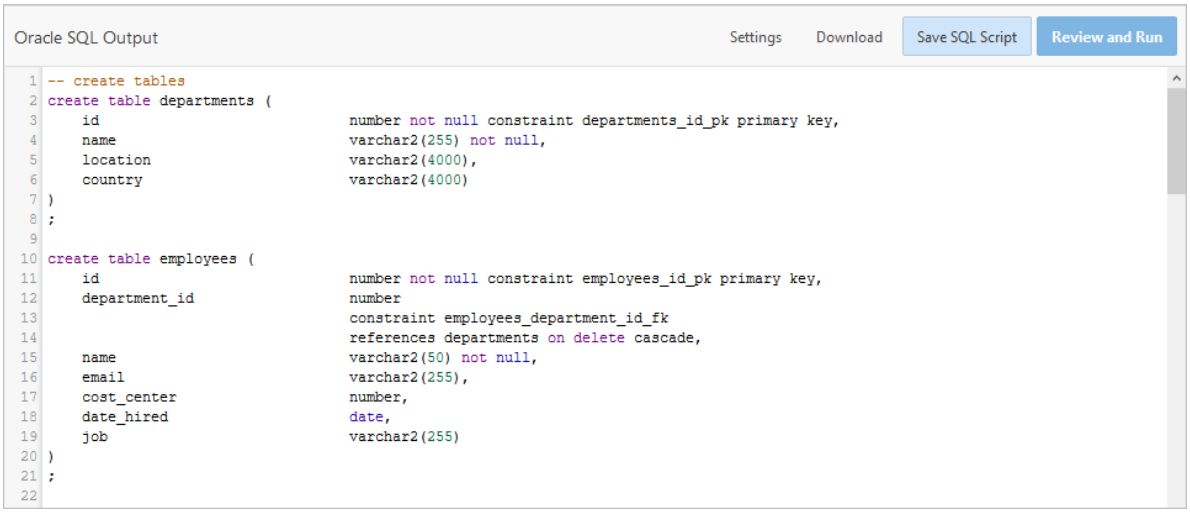
To generate the Quick SQL with SQL Shorthand:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.
4. In the Quick SQL Shorthand pane, enter SQL shorthand syntax.

5. Click **Generate SQL**.

   Displays the generated SQL in Oracle SQL Output pane.



> **Note:**
>
> You can save the generated script or review and run the script.

6. Click **Save SQL Script**.

   The Save Script dialog appears.

7. In the Save Script dialog:

   a. In Script Name, enter a name.

   b. Click **Save Script**.

      The script is saved under SQL Scripts.

> **Note:**
>
> To create an application from script, go to SQL Scripts page and run the saved script. You can also review and run the script by clicking **Review and Run** button.

> **See Also:**
>
> " Using SQL Scripts (page 5-1)"

4.3.4 Saving Scripts

- Saving a SQL Script (page 4-27)
  You can create and save a SQL script.
- Downloading SQL Script (page 4-28)
  Download a generated SQL locally using the Download button on the Oracle SQL Output pane.

4.3.4.1 Saving a SQL Script

You can create and save a SQL script.

To save a SQL script:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.

> **Note:**
>
> You can also access Quick SQL through SQL Scripts page by clicking Quick SQL button on the SQL Scripts page.

4. Enter the SQL in Quick SQL Shorthand or select the sample from **Samples**.
5. Click **Generate SQL**.

   The generated SQL appears on the Oracle SQL output pane.
6. Click **Save SQL Script**.

   The Save Script page appears.
7. Click **Save SQL Script**.
8. In the Save Script dialog:
   a. In Script Name , enter a name.
   b. Click **Save Script**.

      The SQL script is saved under SQL Scripts.

> **Note:**
>
> To create an application from script, go to SQL Scripts page and run the saved script. You can also review and run the script by clicking **Review and Run** button.

> **See Also:**
>
> "About SQL Scripts (page 5-1)"

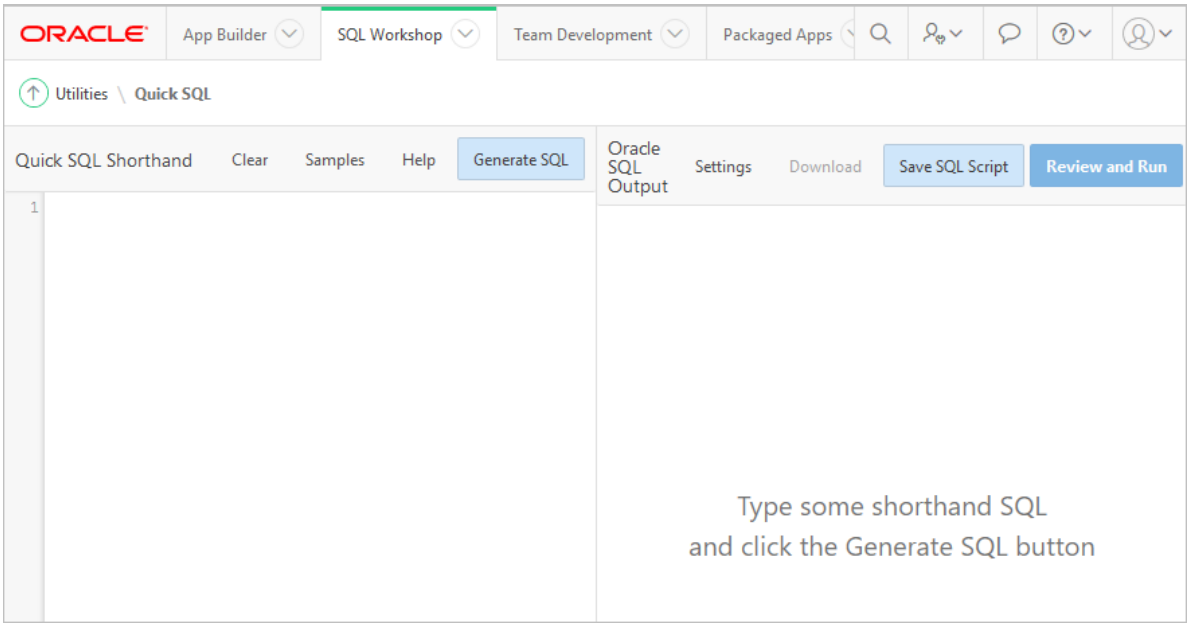
4.3.4.2 Downloading SQL Script

Download a generated SQL locally using the Download button on the Oracle SQL Output pane.

To download SQL Shorthand:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.



> **Note:**
>
> You can also access Quick SQL through SQL Scripts page by clicking Quick SQL button on the SQL Scripts page.

4. Enter the SQL in Quick SQL Shorthand or select the sample from **Samples**.
5. Click **Generate SQL**.

   The generated SQL appears on the Oracle SQL output pane.

> **Note:**
>
> If you have specified any settings, navigate to the bottom of the Oracle SQL Output pane and copy the line starting with `# settings =` to the top of the Quick SQL Shorthand pane.

6. Click **Download** on the Oracle SQL Output pane to save the SQL file to your computer.

4.3.5 Creating an Application from a Generated SQL Script

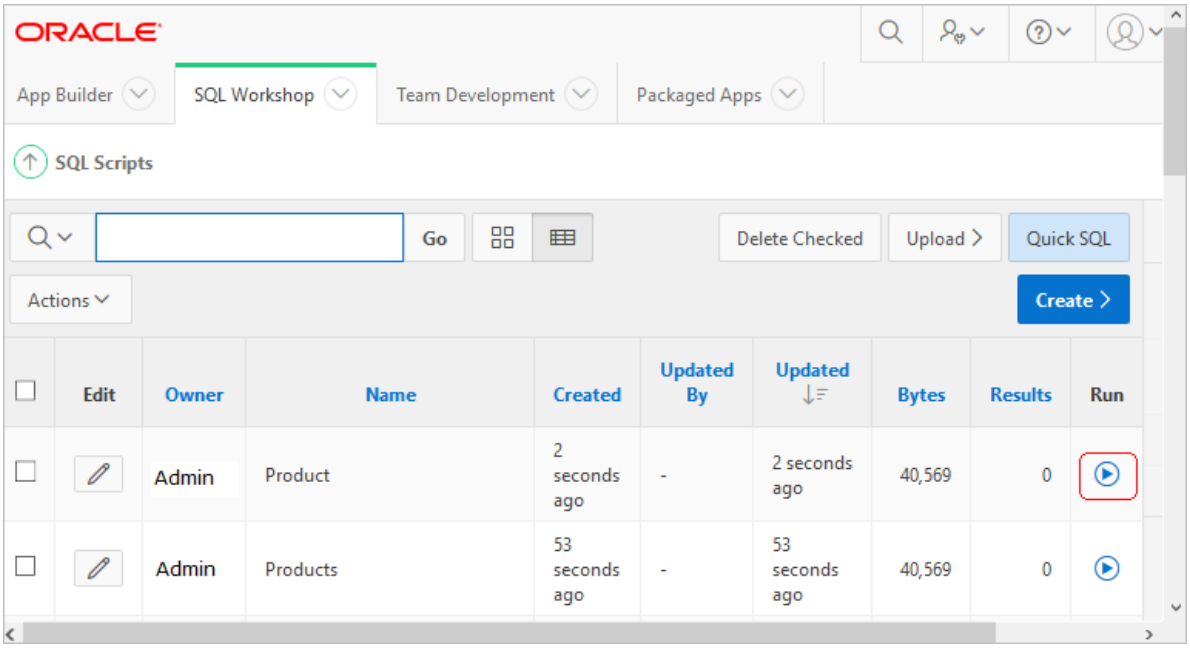
When you save a script it saves it to SQL Scripts and you can directly run the Create Application Wizard from SQL Scripts page.

To create an application from script:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **SQL Scripts**.

The SQL Script page appears.

3. Locate the script and click **Run**.



The Run Script page appears.

4. Click **Run Now**.

   The Results page appears.

5. To create an application, click **Create App from Script**.

   The Create Application Wizard appears.

> **Note:**
>
> Creating a database application is a multiple step process. Once you specify the application name and appearance, you add pages, select features, and configure application settings.

6. In the Create Application Wizard:

   a. Enter the application name, select an appearance, add pages, select features, and configure application settings.

   > **Tip:**
   >
   > To learn more, see online Help.

   b. Click **Create Application**.

> **See Also:**
>
> "Understanding Page Types, Features, and Settings"

4.3.6 Using Quick SQL Samples

- About Quick SQL Samples (page 4-31)
  Quick SQL includes samples to assist you in learning to use SQL Shorthand syntax.
- Accessing Quick SQL Samples (page 4-31)
  Use SQL Shorthand syntax to generate SQL.

4.3.6.1 About Quick SQL Samples

Quick SQL includes samples to assist you in learning to use SQL Shorthand syntax.

You have several options when generating Quick SQL. Enables you to use a shorthand format and generate SQL.

Available samples are:

- **Departments and Employees** - Create a **DEPARTMENTS** and **EMPLOYEES** table with a SQL join VIEW.
- **Employee Skills** - Create a **DEPARTMENTS**, **EMPLOYEES**, and **SKILLS** table using indentation to identify parent and child tables.
- **Product Sales** - Create a star schema with **PRODUCTS**, **CUSTOMERS**, **CHANNELS**, **PROMOTIONS**, and **SALES** tables.
- **Project Management** - Create a **PROJECT_MANAGEMENT** and four detail tables.

4.3.6.2 Accessing Quick SQL Samples

Use SQL Shorthand syntax to generate SQL.

To access the Quick SQL page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.
4. Click **Samples** on the Quick SQL Shorthand pane.

   The Samples page appears.
5. Choose a sample and click **Load Model**.

   The Sample loads on Quick SQL Shorthand pane and generated SQL is displayed in Oracle SQL Output pane .

> **Note:**
>
> You can save the generated script or review and run the script.

4.3.7 Configuring Settings

- About Settings (page 4-32)
  Learn how to set SQL syntax generation options.
- Configuring Settings Declaratively (page 4-32)
  You can configure settings declaratively to explicitly set SQL syntax generation options.
- Configuring Settings Inline (page 4-34)
  You can configure settings inline to explicitly set SQL syntax generation options.

4.3.7.1 About Settings

Learn how to set SQL syntax generation options.

You can use inline settings to explicitly set SQL syntax generation options by either entering settings directly into the Quick SQL Shorthand pane, or by clicking the **Settings** button at the top of the Oracle SQL Output pane.

4.3.7.2 Configuring Settings Declaratively

You can configure settings declaratively to explicitly set SQL syntax generation options.

To configure settings declaratively:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Quick SQL**.

   The Quick SQL page appears.
4. Click **Settings** on the Oracle SQL Output pane.

   The Settings page appears. Configure the appropriate attributes.

   > **Tip:**
   >
   > To learn more about an attribute, see field-level Help.

5. Under Table:
   - Object Prefix - Database object prefix. Prefixes all object names with the provided value .
   - Schema - Prefix object names with a schema name. The default is no schema prefix for object names.

- On Delete - This setting controls how foreign key behaves On Delete settings. Valid values include: Cascade, Restrict, Set Null
- Compression - When enabled creates all tables compressed. Default is not compressed.
- Generate Inserts - Turn on the when you are ready to generate data.

6. Under Column:
   - Primary Keys - Determines how the primary key is set.
   - Date Data Type - Defines what type of date to use to store columns identified as dates. Options include: Date,Timestamp,Timestamp with local time zone.
   - Semantics- Choose no column semantics, or BYTE or CHAR semantics.
7. Under Additional Columns:
   - Include - Additional columns to automatically add to your table. Options include: Audit Columns, Row Key, Row Version Number, Security Group ID.
8. Under Options:
   - Data Language - Controls the language used for generating the data using the /INSERT flag.
   - PL/SQL API - Generates PL/SQL APIs on all tables for create, insert, update, delete and query.
   - Include Drops - Includes SQL commands to drop each database object created.
   - History Tables - Maintains history of column value changes in a history table.
   - Auto Primary Key - Prefixes primary key database table columns with name of table.
   - Longer Varchars - Allows longer identifiers to be used for database object names.
   - Editionable - When generating PL/SQL objects including triggers and packages make these editionable.
   - APEX Enabled - Controls the syntax generated to support audit columns.
   - Tags Framework - Choose Yes or No.
   - Compatibility - Ensures generated SQL is compatible with the database release selected.
9. Under Preferences:
   - Prefix primary keys with table name - Prefix primary key database table columns with name of table.
   - Created Column Name - Determines the created column name.
   - Created By Column Name - Determines who created the column.
   - Updated Column Name - Determines the updated column name.
   - Updated By Column Name - Determines who updated the column name.
10. Click **Save Changes** to save the configuration.

    Click **Reset** to reset to default settings.

4.3.7.3 Configuring Settings Inline

You can configure settings inline to explicitly set SQL syntax generation options.

Entering settings directly into the Quick SQL Shorthand pane ensures the same SQL generation options are utilized even if you download the script and later paste it back. For example, enter the following to prefix all table names with TEST and generate for schema OBE:

```
# settings = { prefix: "test", schema: "OBE" }.
Alternatively, enter each setting on a separate line for the same result:
# prefix: "test"
# schema: "OBE"
```

> **Note:**
>
> The settings must start on a new line and begin with `# settings =` to enter multiple settings, or `#` to enter a single setting per line. All values are case insensitive. Brackets, spaces, and commas can be added for clarity but are ignored. To have all settings generated use `# verbose: true`.

> **Tip:**
>
> You can find the setting inline, by clicking **Help**, **Settings**.

4.3.8 About Database Terminology

Common database terms referenced in Quick SQL include:

- **BLOB** – A binary large object or `CLOB`. BLOBs store binary data and do not impose any character set encoding. `BLOB` size can store 4GB or more depending on system configuration.
- **Check Constraint** – A check constraint is a declarative way to limit the values a given column may contain.

  ```
  create table projects (
                id     number primary key,
                name   varchar2(50),
                status varchar2(30) check (status in ('OPEN','CLOSED'))
                );
  ```

- **CLOB** – A character large object or `CLOB`. The maximum size of a `VARCHAR2` datatype is 32,767 bytes, however a `CLOB` size can be store up to 4GB of character data.
- **Column** – RDBMS tables store data in tables that are comprised of rows and columns. Each column is named. For example, `first_name`.
- **Database** – A structured set of data held in a computer, especially one that is accessible in various ways.

- **Data Model** – A Data Model is a method of organizing elements of data and defining how they relate to one another. A model typically organizes data into entities that when they are transformed into a physical database model are converted into tables and columns. This tool is a very rudimentary tool that creates physical data models from indented text. For serious data modeling it is recommended that you use a proper data modeling tool Oracle SQL Data Modeler. It is a free graphical tool that enhances productivity and simplifies data modeling tasks.
- **Data Type** – Each column in a table has a data type. Data types are frequently character (`VARCHAR2`), `number`, or `dates`.
- **DML** – Data Manipulation Language. SQL commands that include the following:
  - `INSERT` - SQL statement used to add new rows to a table.
  - `UPDATE` - SQL statement used to update one or more rows of data within a table.
  - `DELETE` - SQL statement used to remove one or more rows of data from a table.
- **Foreign Key** – A column(s) in a table that references another table. For example, if employees are organized by departments, a `department_id` in the employees table could be a foreign key to the primary key of the `employee` table. By defining foreign keys you are declaratively defining a relationship between tables.
- **Join** – A join is a SQL query that combines rows from two or more tables or views. A query is a join query whenever multiple tables appear in the `FROM` clause. A major benefit of the relational database model is that you can store data that repeats in one location. For example, assume we have two table departments, and employees. Here is a query that selects data from both tables.

  ```
  select d.name, d.location, e.name, e.job
          from   departments d, employees e
          where  e.department_id = e.id
  ```

  Note if the department name or location changes it is changed in only one place.
- **Relational Database** – A database model that organizes data in a relational model. The relational model organizes data into one or more tables, also known as relations, of columns and rows. The standard language to access data within a relational database and to define database objects is SQL.
- **RDBMS** – Relational Database Management System.
- **Object** – A database object is a named widget used to store or manage data. For example `tables`, `views`, `triggers`, and `synonyms` are all considered database objects.
- **PL/SQL** – Oracle procedural extension to the SQL language. PL/SQL stands for Procedural Language SQL. PL/SQL is a server-side, stored procedural language that can be used to perform procedural operations on Oracle database data.
- **Query** – To access / query data in your relational database you use a SQL `SELECT` statement. In SQL a `SELECT` statement can query data from one or more tables or views.
  - Query of 3 columns from a table called projects

    ```
    select id, name, status
        from projects
        where status = 'OPEN';
    ```
  - Aggregate Query

```
select count(*) count_of_projects,
           sum(case when status = 'OPEN' then  1 else 0) open_projects,
           sum(case when status = 'CLOSED' then  1 else 0) closed_projects
    from projects;
```

- **Reserved Word** – A name that cannot be used as an object identifier. For example `SELECT` cannot be used as a table name.
- **Row** – RDBMS tables store data in tables that are comprised of rows and columns. Each row is made up of one or more columns.
- **Sequence** – Sequences are database objects from which multiple users can generate unique integers. The sequence generator generates sequential numbers, which can help to generate unique primary keys automatically, and to coordinate keys across multiple rows or tables. A sequence can be created with the following Syntax:

  ```
  create sequence project_seq;
  ```

  The sequence values can be referenced using `project_seq.currval` (current value) or `project_seq.nextval` (next value). Each time the `nextval` is access the sequence increments. The default increment is`1`.
- **Schemas** – A schema is a namespace for creating database objects such as tables. You do not create database tables in databases you create them within schemas within a database. Each database can store any number of schemas. Schemas are also sometimes called users. When you have a new database you first create a schema (or user, they are the same thing) then create your tables within the schema or user. You can create schemas using the SQL `create schema` or `create user` command. Schemas are a way of organizing database objects within a database.
- **SQL** – Structure Query Language. The language used to define database tables, and to manipulate the data within the tables.
- **Structured Data** – Relational database data is frequently referred to as structured because data is organized cleanly into rows and columns. Unstructured typically refers to data that has no structure imposed on it.
- **Table** – A database object that stores related data organized into rows and columns.
  - Creating a table

    ```
    create table projects (
            id     number primary key,
            name   varchar2(50),
            status varchar2(30)
            );
    ```
  - Inserting rows into a table

    ```
    insert into projects (id, name, status)
          values (1, 'My Project','OPEN');
    ```
  - Querying the data with the table

    ```
    select id, name, status
          from projects
          where status = 'OPEN';
    ```
- **Trigger** – A database trigger is procedural code that is automatically executed in response to certain events on a particular table. For example before or after insert of a row. The following trigger forces a column to lower case.

```
create or replace trigger PROJECTS_BIU
            before insert or update
            on PROJECTS
            for each row
        begin
            :new.OWENER_EMAIL := LOWER(:new.OWENER_EMAIL);
        end;
    /
```

- **View** – A database view is a virtual table based on the result-set of an SQL statement. A view is also known as a stored query. You can create a view that joins multiple tables. Once the view is created you can query it as if it was a table. Following is an example of the SQL to create a view:

```
create view department_employees as
select d.name, d.location, e.name, e.job
    from   departments d, employees e
    where  e.department_id = e.id;
```

4.3.9 Recommendations When Using Quick SQL

There are many valid approaches to creating an effective data model. Oracle recommends the following method:

- **Table Naming** – Define your tables as plural and columns as singular. Since tables are designed to store many rows it makes sense to have them plural. For example `Employees`.
- **Column Naming** – Define your column names as singular. For example `First Name`.
- **Primary Keys** – Every table should have a primary key, primary keys are used to uniquely identify each row and can be used to associate child table data.
- **Primary Key Column Naming** – Every table will have a primary key and it should be called `ID` and it should be numeric.
- **Table Constraints** – The best way to ensure data integrity is to have the database only allow proper data. For example, if a column is required ensure you have a `not null` constraint. If a column is unique ensure you have a unique constraint.
- **Referential Integrity** – Most relational database models have parent child, also known as one too many relationships. They frequently also have many to many relationships. Using declarative primary key and foreign key syntax you can have the database enforce these relationships to ensure the integrity of your data. By defining the referential integrity you ensure all child tables do in fact relate to parent tables.

4.3.10 About Supported Data Types

Quick SQL supports the following data types:

- `VARCHAR2`
- `NUMBER`
- `INTEGER`
- `DATE`

- `TIMESTAMP`
- `TIMESTAMP WITH TIME ZONE`
- `TIMESTAMP WITH LOCAL TIME ZONE`
- `CLOB`
- `BLOB`

4.4 Using Sample Datasets

Use Sample datasets to build sample applications.

- About Sample Datasets (page 4-38)
  Sample datasets enable you to easily build sample applications. Each dataset includes sufficient data to build pages with various different components.
- Accessing Sample Datasets (page 4-38)
  Access the Sample Dataset page by clicking Utilities and then Sample Datasets.
- Installing a Sample Dataset (page 4-39)
  You can install sample dataset to your workspace.
- Removing a Sample Dataset (page 4-40)
  You can remove the installed dataset.
- Refreshing a Sample Dataset (page 4-41)
  You can refresh the installed dataset.
- Changing the Schema for a Sample Dataset (page 4-41)
  You can change the schema for a installed dataset.

4.4.1 About Sample Datasets

Sample datasets enable you to easily build sample applications. Each dataset includes sufficient data to build pages with various different components.

Install, refresh, or replace sample datasets within one of the schemas associated with your workspace. The database objects can also be easily removed.Each dataset includes various database objects and sample data. The sample data is available in one or more languages.

> **Note:**
>
> Only a single language can be loaded into the sample database objects. If you select a different language, the currently loaded data is replaced.

4.4.2 Accessing Sample Datasets

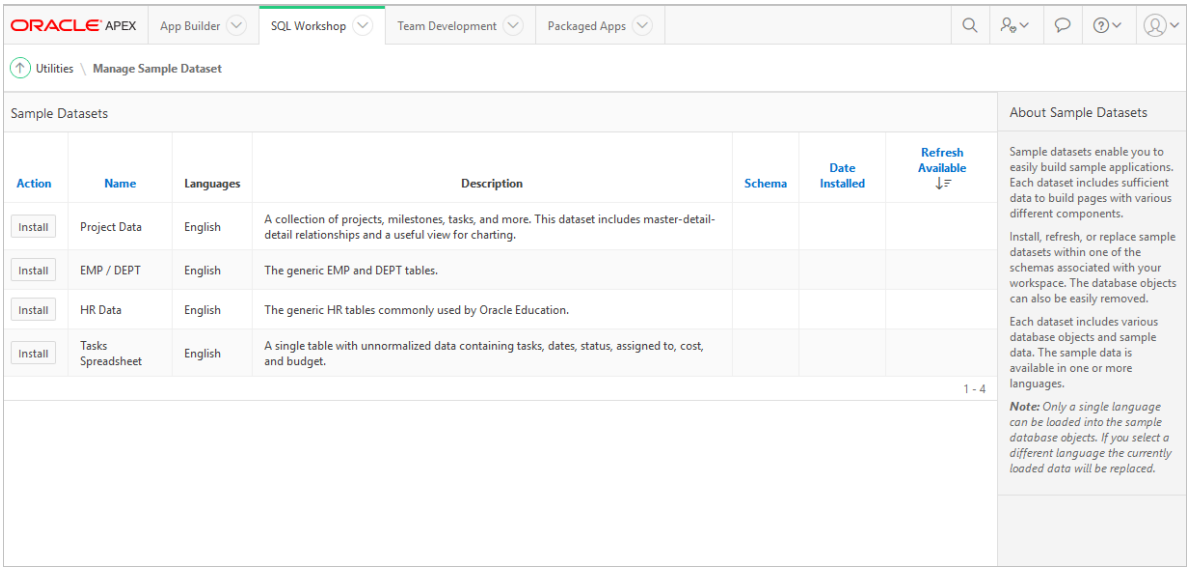
Access the Sample Dataset page by clicking Utilities and then Sample Datasets.

To access the Sample Datasets page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.

3. Click **Sample Datasets**.

   The Sample Datasets page appears with a list of datasets.



4.4.3 Installing a Sample Dataset

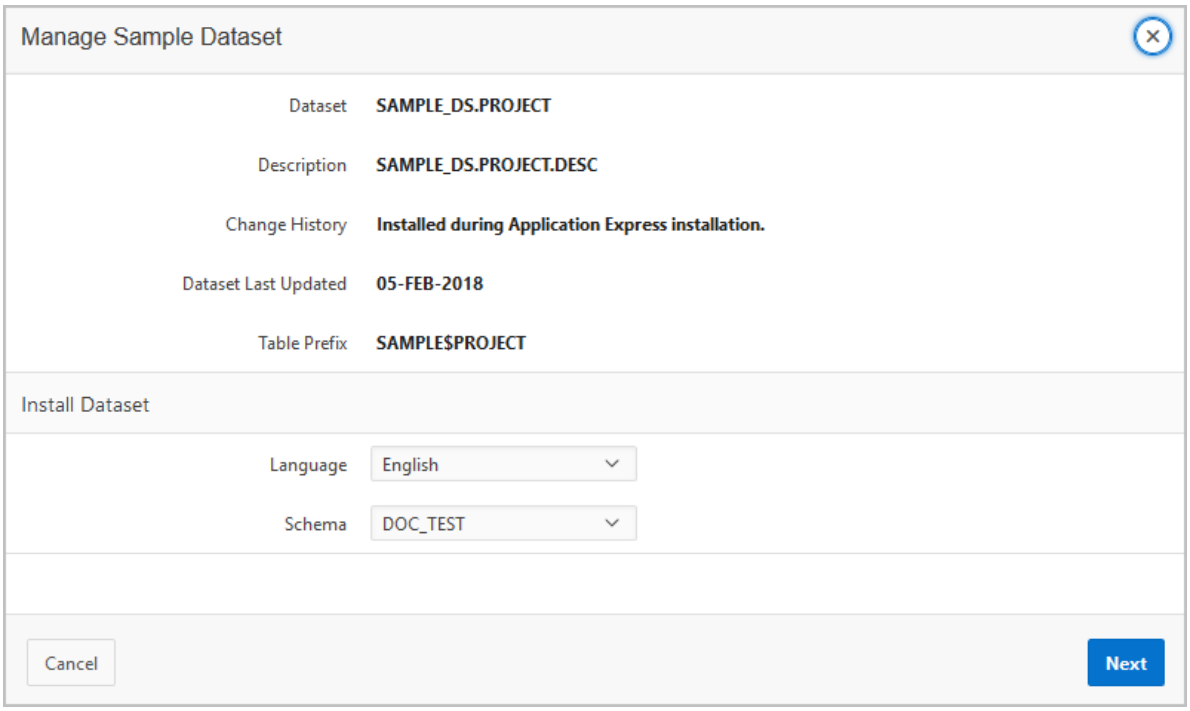
You can install sample dataset to your workspace.

To install the Sample Datasets:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Sample Datasets**.

   The Sample Datasets page appears with a list of datasets.
4. Select the sample dataset and click **Install** under Action.

   The Manage Sample Dataset page appears.



5. Click **Next**.

   The Load Sample Dataset page appears with details of database objects.

6. Click **Install Dataset**.

   The Load Sample Dataset - Result page appears with the list of installed database objects.

7. On Load Sample Dataset
   - Click **Exit** to exit the wizard.
   - Click **Create Application** to create a new application based on this database.

> **See Also:**
>
> "Creating a Database Application Based on a Table or Query"

4.4.4 Removing a Sample Dataset

You can remove the installed dataset.

To remove installed dataset:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Sample Datasets**.

   The Sample Datasets page appears with a list of datasets. Installed Datasets displays details of Schema and Date Installed.

4. Click **Update** under Action.

The Manage Sample Dataset page appears.

5. Click **Remove Dataset**.

4.4.5 Refreshing a Sample Dataset

You can refresh the installed dataset.

To install the Sample Datasets:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Sample Datasets**.

   The Sample Datasets page appears with a list of datasets. Installed Datasets displays details of Schema and Date Installed.
4. Click **Update** under Action.

   The Manage Sample Dataset page appears.
5. Click **Next**.
6. Select **Refresh Existing Dataset**.

4.4.6 Changing the Schema for a Sample Dataset

You can change the schema for a installed dataset.

To change the schema for a Sample Datasets:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **Sample Datasets**.

   The Sample Datasets page appears with a list of datasets. Installed Datasets displays details of Schema and Date Installed.
4. Click **Update** under Action.

   The Manage Sample Dataset page appears.
5. Select **Schema**.

   > **Note:**
   >
   > This option is displayed if you have more than one schema associated with your workspace.

6. Click **Next**.

4.5 Generating DDL

With Oracle Application Express, you can generate data definition language (DDL) statements from the Oracle data dictionary.

You can use these scripts to create or re-create database schema objects. The scripts can be generated to display inline or saved as a script file. You can generate the create scripts for all objects for a specific schema, specific object types, or specific objects.

To generate a DDL statement:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Generate DDL**.

   The Generate DDL page appears.
4. Click **Create Script**.

   The Generate DDL Wizard appears.
5. Select a database schema and click **Next**.
6. Define the object type:
   a. Output - Specify an output format. Select either **Display Inline** or **Save As Script File**.
   b. **Check All** - Select this option to include all object types for which to generate DDL.
   c. **Object Type** - Select the object types for which to generate DDL.
   d. To select object names for the selected object types, click **Next** and follow the on-screen instructions.
7. Click **Generate DDL**.

4.6 Managing User Interface Defaults

Managing user interface defaults enables you to assign default interface properties, create default wizard, modify the table user interface defaults, add new attributes, and export the user interface defaults.

- About User Interface Defaults (page 4-43)
  User Interface Defaults enable you to assign default user interface properties for regions and items.
- Creating User Interface Defaults for a Table (page 4-43)
  If no defaults have been created for a table, use the Table Dictionary Create Defaults wizard to automatically generate defaults based on the database table and column definitions. After the initial defaults are created, you can modify the individual default values.
- Modifying Table User Interface Defaults (page 4-44)
  Learn how to modify existing table user interface defaults.

- Creating User Interface Attributes (page 4-45)
  Use the Attribute Dictionary to define and add new attributes.
- Modifying User Interface Attributes (page 4-45)
  Modify existing user interface attributes on the Grid Edit page or by selecting a specific column.
- Exporting and Importing User Interface Defaults (page 4-46)
  You export user interface defaults in the same way you export any related application file.

4.6.1 About User Interface Defaults

User Interface Defaults enable you to assign default user interface properties for regions and items.

This wizard enables you to specify whether you want to use User Interface Defaults if they exist. When you create a form or report using a wizard, the wizard uses this information to create default values for region and item properties. Utilizing user interface defaults can save valuable development time and has the added benefit of providing consistency across multiple pages in an application. User Interface Defaults are divided into two categories, the Table Dictionary and the Attribute Dictionary.

- The Table Dictionary enables you to specify defaults for tables and columns that are initialized from the database definition.
- The Attribute Dictionary enables you to create defaults based on attribute or column names, thus being usable across all tables. Attribute definitions can also have synonyms, allowing more than one attribute to share a common definition.

When you use User Interface Defaults during creation of pages and regions, the Table Dictionary takes priority over the Attribute Dictionary. If a table and column combination exists, that is used rather than an attribute definition of the same name. This can be useful, for example, when you want to have a specific label or Help text for the `CREATED_BY` column in the `emp` table but then use more generic defaults for `CREATED_BY` in another table.

4.6.2 Creating User Interface Defaults for a Table

If no defaults have been created for a table, use the Table Dictionary Create Defaults wizard to automatically generate defaults based on the database table and column definitions. After the initial defaults are created, you can modify the individual default values.

To create the initial user interface defaults for tables:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**.

   The Dashboard page for User Interface Defaults appears.
4. Click **Table Dictionary**.

   The Table Dictionary page displays a report of all objects in your workspace that includes a Defaults Exist column indicating whether defaults have been created for each object.

5. Click the object name for which you want to create defaults.

   The Create Table Dictionary Defaults page appears.

6. Click **Create Defaults**.

   The Table Dictionary page displays a report showing that defaults exist for the object you selected.

> **Note:**
>
> Synchronizing defaults with the database data dictionary creates table based defaults for any tables in the current schema that do not have defaults yet and adjusts those of tables that do to match the database definition.

> **See Also:**
>
> "Modifying Table User Interface Defaults (page 4-44)"

4.6.3 Modifying Table User Interface Defaults

Learn how to modify existing table user interface defaults.

To modify existing table defaults:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**.

   The Dashboard page for User Interface Defaults appears.

4. Click **Table Dictionary**.

   The Table Dictionary page displays a report of all objects in your workspace. The Defaults Exist column indicates whether defaults have been created for an object.

5. Click the object name you want to modify.

   The Table and Column Properties page displays a report of column defaults.

6. Click the column name you want to modify.

   The Column Defaults page displays.

7. Make modifications and click **Apply Changes**.

   See field-level help for a description of the individual settings on this page.

> **Note:**
>
> A column can be removed from the Table Dictionary, thus allowing the Attribute Dictionary defaults to be used during the creation process. For example, when you have auditing columns where you may want the exact same labels and help across every instance of `CREATED_BY` and `CREATED_ON`, regardless of which table they come from, you could simply remove the column defaults from each table for those columns.

4.6.4 Creating User Interface Attributes

Use the Attribute Dictionary to define and add new attributes.

To create an attribute:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**.

   The Dashboard page for User Interface Defaults appears.
4. Click **Attribute Dictionary**.

   The Attribute Dictionary page displays a report of all attributes in your workspace.
5. Click **Create**.

   The Column Details page appears.
6. Enter attribute specifications and click **Create**.

   See field-level help for a description of the individual settings on this page.

4.6.5 Modifying User Interface Attributes

Modify existing user interface attributes on the Grid Edit page or by selecting a specific column.

To modify an existing attribute:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **User Interface Defaults**.

   The Dashboard page for User Interface Defaults appears.

   The current schema displays to the right of the breadcrumb menu.
4. Click **Attribute Dictionary**.

   The Attribute Dictionary page displays a report of all attributes in your workspace.
5. To edit select attributes for all displayed columns:
   a. Click **Grid Edit**.
   b. Edit the appropriate attributes and click **Apply Changes**.
6. To edit a specific column, click the edit icon for the attribute you want to modify.

The Column Details page displays.

> **Note:**
>
> If you select to edit a synonym, identified by having the `Synonym of` column populated, you are taken to the definition for the base column, because the base column and all synonyms share one definition. To delete the synonym, select the synonym within the Synonyms region. Deleting the base column deletes the column and all synonyms.

7. Make modifications and click **Apply Changes**.

   See field-level help for a description of the individual settings on this page.

4.6.6 Exporting and Importing User Interface Defaults

You export user interface defaults in the same way you export any related application file.

Exporting user interface defaults from one development instance to another involves the following steps:

1. Export the user interface defaults using the Export User Interface Defaults utility.
2. Import the exported file into the target Oracle Application Express instance.
3. Install the exported file from Export Repository.

When you export user interface defaults, all user interface defaults for the selected schema are exported to a single script. The file contains an API call to create table hints by making calls to the application PL/SQL API. You can use this file to import user interface defaults to another database and Oracle Application Express instance.

4.7 Comparing Schemas

You can run a report that compares database objects in two schemas, displaying differences between them. You can compare all objects in the schemas or limit your report to specific objects.

To compare two schemas, both must be available to your workspace.

Examples:

- Compare `DEMO_ objects` by searching for that naming convention. The report indicates if the object exists in each of the two schemas.
- Analyze the object details in the two schemas to determine why one implementation is different. For example, the report might show that an index in one schema has an additional column or a column with a different data type.

To compare schemas:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Schema Comparison**.

4. On the Schema Comparison page, make the appropriate selections to run the comparison:
   a. **Schema 1 and Schema 2** - Select the schemas to compare.
   b. **Compare** - Restrict the report to show one object type or select **All** to show all database objects.
   c. **Search** - Enter a case insensitive query for the object name.
   d. **Rows** - To change the number of rows that appear in the report, make a selection from the Display list.
   e. **Compare** - Click **Go** to find the results matching your selections.
   f. **Show Differences Only or Show Details** - Select the type of information you want to review.
   g. Click **Compare**.

4.8 Managing Methods on Tables

The Managing Methods on Tables utility enables you to create an application programming interface, or package, based on a specified table.

To create a package:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Methods on Tables**.

   The Create Table API page appears.
4. Enter a Package name. To have the final package name match the case entered in the Package Name field, click **Preserve Case** and click **Next**.
5. Select the table(s) for which you want to generate the PL/SQL package-based application program interface (API) and click **Next**.
6. A confirmation page appears, which displays the subprogram(s) and their description. You have the option to show or download the specification or body. To confirm, click **Create Package**.

4.9 Using the Recycle Bin

You can use the Recycle Bin to view and restore dropped database objects.

When you drop a table, the space associated with the table is not immediately removed. The Oracle database renames the table and places it and any associated objects in the Recycle Bin. You can recover objects in the Recycle Bin at a later time.

> **Note:**
>
> The Recycle Bin feature is only available if you run with an Oracle 10g or later database.

- Managing Objects in the Recycle Bin (page 4-48)
  Once you select an object and view the Object Details page, you can choose to purge the object or restore the object by clicking the appropriate button.
- Emptying the Recycle Bin Without Viewing the Objects (page 4-48)
  You can empty the Recycle Bin without viewing the objects by using Purge Recycle Bin option.

4.9.1 Managing Objects in the Recycle Bin

Once you select an object and view the Object Details page, you can choose to purge the object or restore the object by clicking the appropriate button.

You can view objects in the Recycle Bin on the Dropped Objects page.

To view objects in the Recycle Bin:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Recycle Bin.**
4. Click **Dropped Objects**.

   The Dropped Objects page appears.
5. To filter the report, select an object type, enter the object name in the Original Name field, and click **Go**.
6. To view object details, click the object name.

   The Object Details page appears.
7. To restore the current object, click **Restore Object**.
8. To permanently delete the current object, click **Purge Object**.

4.9.2 Emptying the Recycle Bin Without Viewing the Objects

You can empty the Recycle Bin without viewing the objects by using Purge Recycle Bin option.

To empty the Recycle Bin without viewing the objects:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Recycle Bin**.
4. Click **Purge Recycle Bin**.

   The Purge Recycle Bin page appears.
5. Confirm your request by clicking **Purge Recycle Bin** again.

4.10 Viewing Object Reports

SQL Workshop Utilities includes a variety of object reports to help you better manage the objects in your database.

- Viewing Table Reports (page 4-49)
  Use the Table reports to view specific details about the tables within your database.
- Viewing PL/SQL Reports (page 4-49)
  Use the PL/SQL reports to view program unit arguments or unit line counts and also to search PL/SQL source code.
- Viewing Exception Reports (page 4-51)
  Use the Exception Reports to view unindexed foreign keys and tables without primary keys, indexes, or triggers.
- Viewing Security Reports (page 4-51)
  Use the Security reports to view object or column privileges granted on database objects owned by other schemas.
- Viewing All Object Reports (page 4-52)
  Use the All Object reports to view objects for the selected schema.

4.10.1 Viewing Table Reports

Use the Table reports to view specific details about the tables within your database.

To view the Table reports:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. Under Table Reports, select a report to view:
   - **Table Columns**
   - **Table Comments**
   - **Table Constraints**
   - **Table Statistics**
   - **Table Storage Sizes**
5. To filter a report, enter search criteria in the field provided and click **Go**.
6. To view reports for a different schema, select another schema from the Schema list.

4.10.2 Viewing PL/SQL Reports

Use the PL/SQL reports to view program unit arguments or unit line counts and also to search PL/SQL source code.

- Viewing Package Input and Output Parameters (page 4-50)
  Use the Program Unit Arguments report to view package input and output parameters.
- Viewing the Number of Lines of Code for an Object (page 4-50)
  Use the Unit Line Counts report to view the number of lines of code for each object.

- Searching PL/SQL Source Code (page 4-50)
  Use the Search PL/SQL Source code page to search the text within your PL/SQL code.

4.10.2.1 Viewing Package Input and Output Parameters

Use the Program Unit Arguments report to view package input and output parameters.

To view the PL/SQL Unit Arguments report:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. Under PL/SQL Reports, click **Program Unit Arguments**.
5. To filter a report, enter search criteria in the field provided and click **Go**.
6. To view reports for a different schema, select another schema from the Schema list.

4.10.2.2 Viewing the Number of Lines of Code for an Object

Use the Unit Line Counts report to view the number of lines of code for each object.

Use this report to identify larger PL/SQL program units.

To view the Unit Line Counts report:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. Under PL/SQL Reports, click **Unit Line Counts**.

   The PL/SQL Unit Line Counts report appears.
5. To view reports for a different schema, select another schema from the Schema list.

4.10.2.3 Searching PL/SQL Source Code

Use the Search PL/SQL Source code page to search the text within your PL/SQL code.

Use this report to find references to tables or functions you might be thinking of deleting. You can also use this page to locate code when you can only recall a code snippet.

To search for PL/SQL source code:

1. On the Workspace home page, click the **Utilities** icon.
2. Click **Object Reports**.

The Object Reports page appears.

3. Under PL/SQL Reports, click **Search PL/SQL Source Code**.
4. To filter the report:
   a. In From Line and To Line, enter the range of lines you want to search.
   b. Click **Set**.
   c. To filter the report, enter an object name and click **Go**.
5. To view reports for a different schema, select another schema from the Schema list.

4.10.3 Viewing Exception Reports

Use the Exception Reports to view unindexed foreign keys and tables without primary keys, indexes, or triggers.

To view Exception reports:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. In the Exception Reports section, click the report you want to view.
   - **Tables without Primary Keys**
   - **Tables without Indexes**
   - **Unindexed Foreign Keys**
   - **Tables without Triggers**
5. To filter the report, enter a table name and click **Go**.
6. To view reports for a different schema, select another schema from the Schema list.

4.10.4 Viewing Security Reports

Use the Security reports to view object or column privileges granted on database objects owned by other schemas.

You can also use these reports to view database role and system privileges.

To view the Security reports:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. Under Security Reports, click the report you want to view:

- **Object Grants** - View the privileges for an existing schema and also understand what privileges have been granted from the selected schema to other schemas.
- **Column Privileges** - View the privileges for columns within your schema and also understand what privileges have been granted from the selected schema to other schemas.
- **Role Privileges** - View the database roles that have been granted to a selected schema. Roles are collections of various privileges.
- **System Privileges** - View the database privileges that have been granted to a selected schema.

5. To filter a report, enter search criteria in the field provided and click **Go**.
6. To view reports for a different schema, select another schema from the Schema list.

4.10.5 Viewing All Object Reports

Use the All Object reports to view objects for the selected schema.

To view the All Object reports:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Object Reports**.

   The Object Reports page appears.
4. Under All Object Reports, select a report to view:
   - **All Objects** - List all objects in the current schema. Displays creation date and when the last DDL (data definition language) was performed.
   - **Invalid Objects** - View all invalid objects in the database by object type.
   - **Object Creation Calendar** - View all objects in a calendar format based on the date each database object was created.
   - **Object Counts by Type** - View the number of database objects by type for the selected schema.
   - **Data Dictionary** - View the data dictionary for this database.

     An Oracle data dictionary is a set of tables and views used as a read-only reference about the database. For example, a data dictionary stores information about both the logical and physical structure of the database.

     A data dictionary also stores information about valid Oracle database users, integrity constraints for tables in the database, and the amount of space allocated for a schema object and the amount being used.
5. To filter a report, enter search criteria in the field provided and click **Go**.
6. To view reports for a different schema, select another schema from the Schema list.

4.11 Viewing Database Details

If you have an account that has been granted a database administrator role, you can view details about your database on the About Database page.

To access details about your database:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.
3. Click **About Database**.
4. If prompted, enter the appropriate administrator user name and password and click **Login**.

   The About Database page appears.
5. To view additional information about installed options, currently used features, or National Language Support, select one of the following check boxes and click **Go**:
   - **Version**
   - **Settings**
   - **Options**
   - **National Language Support**
   - **CGI Environment**
   - **Parameters**

4.12 Monitoring the Database

The reports available on the Database Monitor page provide a database-wide view of the database sessions, system statistics, SQL statements, and longer operations.

You can use these reports to identify poorly performing SQL and to better understand the workload of the database.

> **Tip:**
>
> To access any of the icons on the Database Monitor page, you must have an account that has been granted a database administrator role.

- Accessing Session Page Reports (page 4-54)
  A session is the connection of a user to an Oracle database instance.
- System Statistics (page 4-56)
  Displays the System Statistics page statistics.
- Viewing System Statistics (page 4-57)
  Learn how to view the System Statistics page.

- About Top SQL Page (page 4-57)
  The top SQL statements represent the SQL statements that are executed most often, that use more system resources than other SQL statements, or that use system resources more frequently than other SQL statements.
- Viewing Top SQL Page (page 4-58)
  Use the Top SQL page to identify poorly performing SQL.
- About Long Operations Page (page 4-58)
  The Long Operations page displays the status of various operations that run for longer than 6 seconds (in absolute time).
- Viewing the Long Operations Page (page 4-58)
  Learn how to view the Long Operations page.

4.12.1 Accessing Session Page Reports

A session is the connection of a user to an Oracle database instance.

A session lasts from the time the user connects until the time the user disconnects or exits the database application.

- Accessing Reports on the Session Page (page 4-54)
  You must have database administrator privileges to access the Sessions page.
- Creating Sessions Reports (page 4-55)
  The Sessions Report displays information about the current sessions in the database.
- Controlling the Number of Rows for Locks Reports (page 4-55)
  The Locks report displays a report of sessions which have locks that are blocking other session(s).
- Creating Waits Reports (page 4-55)
  The Waits report displays the wait events for each session.
- Creating I/O Reports (page 4-56)
  The I/0 report displays details about the I/O for each session.
- Creating SQL Reports (page 4-56)
  The SQL report displays details about the current or last SQL statement executed for each session.
- Creating Open Cursors Report (page 4-56)
  The Open Cursors report displays details about the number of open cursors for each session.

4.12.1.1 Accessing Reports on the Session Page

You must have database administrator privileges to access the Sessions page.

A session is the connection of a user to an Oracle database instance. A session lasts from the time the user connects until the time the user disconnects or exits the database application.

To access reports on the Sessions page:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.

3. Click **Database Monitor**.
4. Click **Sessions**.
5. If prompted, enter the appropriate database administrator user name and password and click **Login**.

   The Sessions page appears.
6. To view a report, select one of the following tabs at the top of the page:
   - **Sessions**
   - **Locks**
   - **Waits**
   - **I/O**
   - **SQL**
   - **Open Cursors**

4.12.1.2 Creating Sessions Reports

The Sessions Report displays information about the current sessions in the database.

Use the controls at the top of page to narrow the view:

- **Search**. Enter search criteria and click **Go**. For search details, click the Search label.
- **Status**. Select a status and click **Go**.
- **Show**. Select how many columns to display and click **Go**.
- **Rows**. Select the number of rows to appear in the report and click **Go**.

To view session details, click the Session ID (SID). The Session Details page appears. To remove the current session, navigate to the Session Details page and click **Kill Session**.

4.12.1.3 Controlling the Number of Rows for Locks Reports

The Locks report displays a report of sessions which have locks that are blocking other session(s).

To control the number of rows that appear, make a selection from the Rows list and click **Go**.

4.12.1.4 Creating Waits Reports

The Waits report displays the wait events for each session.

Use the controls at the top of page to narrow the view:

- **Search**. Enter search criteria and click **Go**. For search details, click the Search label.
- **Status**. Select a status and click **Go**.
- **Show**. Select how many columns to display and click **Go**.
- **Rows**. Select the number of rows to appear in the report and click **Go**.

To view session details, click the Session ID (SID). The Session Details page appears. To remove the current session, click **Kill Session**.

4.12.1.5 Creating I/O Reports

The I/0 report displays details about the I/O for each session.

Use the controls at the top of page to narrow the view:

- **Search**. Enter search criteria and click **Go**. For search details, click the Search label.
- **Rows**. Select the number of rows to appear in the report and click **Go**.

To view session details, click the Session ID (SID). The Session Details page appears. To remove the current session, click **Kill Session**.

4.12.1.6 Creating SQL Reports

The SQL report displays details about the current or last SQL statement executed for each session.

Use the controls at the top of page to narrow the view:

- **Search**. Enter search criteria and click **Go**. For search details, click the Search label.
- **Status**. Select a status and click **Go**.
- **Show**. Select how many columns to display and click **Go**.
- **Rows**. Select the number of rows to appear in the report and click **Go**.

To view session details, click the Session ID (SID). The Session Details page appears. To remove the current session, click **Kill Session**.

4.12.1.7 Creating Open Cursors Report

The Open Cursors report displays details about the number of open cursors for each session.

Use the controls at the top of page to narrow the view:

- **Search**. Enter search criteria and click **Go**. For search details, click the Search label.
- **Status**. Select a status and click **Go**.
- **Rows**. Select the number of rows to appear in the report and click **Go**.

To view details about a specific open cursor count, click the numeric link under the Open Cursor Count column.

To view session details, click the Session ID (SID). The Session Details page appears. To remove the current session, click **Kill Session**.

4.12.2 System Statistics

Displays the System Statistics page statistics.

The System Statistics page displays statistics for:

- **Physical I/O**. A physical I/O is an I/O that requires disk access. This report displays disk access statistics for physical reads and writes.
- **Logical I/O**. A logical I/O is an I/O that is satisfied in memory or disk. Displays the sum of buffer reads which might be consistent gets or current mode gets. Redo is the buffer in the SGA that contains information about changes.
- **Memory Statistics**. Displays memory consumption of the database.
- **Time Statistics**. Shows various times consumed by the database.
- **SQL Cursor Statistics**. Displays statistics about the cursors in the Oracle database.
- **Transaction Statistics**. Shows the number of transactions performed.

4.12.3 Viewing System Statistics

Learn how to view the System Statistics page.

To view the System Statistics page:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Database Monitor**.
4. Click **System Statistics**.
5. If prompted, enter the appropriate administrator user name and password and click **Login**.

   The System Statistics page appears.

Additional controls on the System Statistics page include:

- **Refresh Report** - Refreshes the System Statistics report.
- **Save Statistics** - Saves the current report.
- **Show delta between current and saved values** - Click this check box to display actual statistic values, or display deltas between a saved value and the current value.

> **See Also:**
>
> "*Oracle Database Performance Tuning Guide*"

4.12.4 About Top SQL Page

The top SQL statements represent the SQL statements that are executed most often, that use more system resources than other SQL statements, or that use system resources more frequently than other SQL statements.

Use the Top SQL page to identify poorly performing SQL.

4.12.5 Viewing Top SQL Page

Use the Top SQL page to identify poorly performing SQL.

To view the Top SQL page:

1. On the Workspace home page, click the **SQL Workshop.**
2. Click **Utilities**.
3. Click **Database Monitor**.
4. Click **Top SQL**.
5. If prompted, enter the appropriate administrator user name and password and click **Login**.

   The Top SQL page appears. Use the search fields and lists and the top of the page and click **Go** to narrow the display. For details on each field or list, click the label.
6. To access the SQL Plan page, click the **View** icon.

   The SQL Plan page appears, containing the following sections:

   - **Query Plan** - Contains a color coded explain plan. Note that unindexed columns display in red.
   - **SQL Text** - Displays the full text of the SQL statement.
   - **Indexes** - Displays all indexes on the table in the query. There is a checkmark when that index is used in the query.
   - **Table Columns** - Shows all columns on all tables or views in the query.

4.12.6 About Long Operations Page

The Long Operations page displays the status of various operations that run for longer than 6 seconds (in absolute time).

These operations currently include many backup and recovery functions, statistics gathering, and query execution, and more operations are added for every Oracle release.

4.12.7 Viewing the Long Operations Page

Learn how to view the Long Operations page.

To view the Long Operations page:

1. On the Workspace home page, click the **SQL Workshop**.
2. Click **Utilities**.
3. Click **Database Monitor.**
4. Click **Long Operations**.
5. If prompted, enter the appropriate administrator user name and password and click **Login**.

See Also:

"V$SESSION_LONGOPS"in *Oracle Database Reference*

5
Using SQL Scripts

Use SQL scripts to create, edit, view, run, and delete script files.

- About SQL Scripts (page 5-1)
  A SQL script is a set of SQL commands saved as a file in SQL Scripts.
- Accessing SQL Scripts (page 5-2)
  Learn how to access SQL Scripts.
- About the SQL Scripts Page (page 5-3)
  The SQL Scripts page displays an interactive report of all SQL scripts created by the current user.
- About the Tasks List (page 5-4)
  A Tasks list displays on the right side of the SQL Scripts page.
- Creating a SQL Script (page 5-4)
  Learn about different ways of creating a SQL script.
- Using the Script Editor (page 5-5)
  You use the Script Editor to add content to a new script, to edit existing scripts, and to run and delete scripts in the script repository.
- Deleting a SQL Script (page 5-7)
  You can delete scripts from the Script Repository by deleting selected scripts from the SQL Scripts page, or deleting the current script in the Script Editor.
- Copying a SQL Script (page 5-8)
  You can copy a script in the Script Repository by saving it with a new name.
- Executing a SQL Script (page 5-8)
  You can execute scripts stored in the Script Repository. You can submit a script for execution either from the Script Editor or from the SQL Scripts page.
- Viewing SQL Script Results (page 5-9)
  You use the Manage Script Results page to view and delete script results.
- Exporting and Importing SQL Scripts (page 5-10)
  You can transfer scripts from your current Script Repository to a Script Repository in a different workspace by using the Export and Import tasks.
- Viewing Script and Result Quotas (page 5-12)
  You can view the script limits in the current workspace on the Script Quotas page.

5.1 About SQL Scripts

A SQL script is a set of SQL commands saved as a file in SQL Scripts.

A SQL script can contain one or more SQL statements or PL/SQL blocks. You can use SQL scripts to create, edit, view, run, and delete database objects.

When using SQL Scripts, remember the following:

- SQL*Plus commands in a SQL script are ignored at runtime.

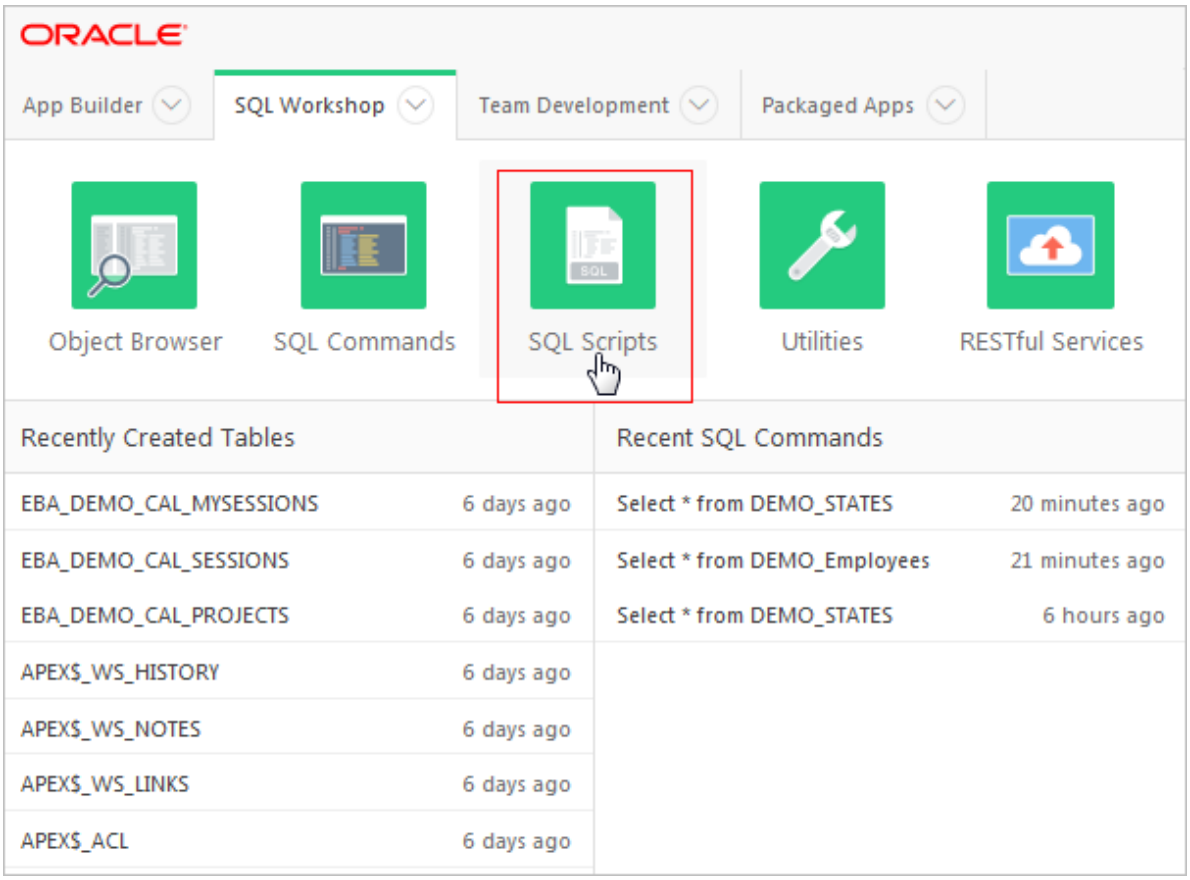
- There is no interaction between SQL Commands and SQL scripts.
- You can cut and paste a SQL command from the SQL Script editor to run it in SQL Commands.
- SQL scripts does not support bind variables.

5.2 Accessing SQL Scripts

Learn how to access SQL Scripts.

To access SQL Scripts:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.



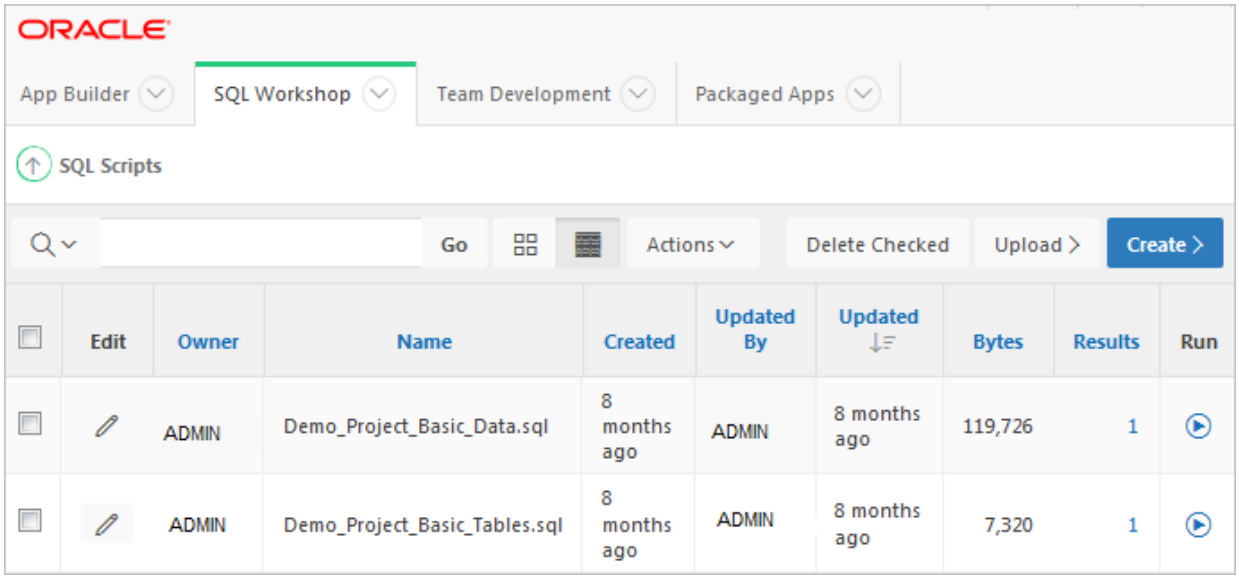
The SQL Scripts page appears.

> **Note:**
>
> If the instance administrator has disabled RESTful Services for this Application Express instance, RESTful Services are not available for this instance and the RESTful Services icon does not display.

5.3 About the SQL Scripts Page

The SQL Scripts page displays an interactive report of all SQL scripts created by the current user.

If you are a workspace administrator, this page displays all SQL scripts created by workspace users. You can alter the layout of report data by choosing the columns you are interested in, applying filters, highlighting, and sorting. You can also define breaks, aggregations, different charts, group bys, and add your own computations.



A search bar displays at the top of every interactive report and includes the following features:

- **Select Columns to Search** - This icon resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select All Columns.
- **View Icons** - Displays each script as an icon identified by the script name.
- **View Report** - Displays each script as a line in a report. Each line includes a check box to enable the selection of scripts for deletion, an edit icon to enable the script to be loaded into the script editor, the script name, the script owner, when the script was last updated and by who, the size in bytes, the number of times the script has been run linked to the run results, and an icon to enable the script to be run.

  Reports view offers the following additional controls:

  – **Delete Checked** - In Reports view, select the check box associated with the script you want to delete and click **Delete Checked**.

  – **Sort** - In Reports view, click a column heading to sort the listed scripts by that column.
- **Upload** - Click **Upload** to upload a script from your local file system into SQL Scripts.

- **Create** - Click **Create** to create a script in the Script Editor.

5.4 About the Tasks List

A Tasks list displays on the right side of the SQL Scripts page.

The Task list contains the following links:

- **Manage Results** - enables you to view, search, and display results.
- **Show Quotas** - displays the Script Quotas page. The Script Quotas page shows the maximum size of a single result, the maximum size of all results, the quota used and the quota free. It also shows the maximum size of a SQL Script.
- **Export** - enables you to export multiple scripts from the current SQL Script Repository for import into SQL Scripts in a different workspace. The scripts you select to export are encoded in a single export script written to your local file system. The export script is named *workspace_name*_script.sql by default.
- **Import** - enables you to import a script exported by this workspace, or a different workspace. **Import** only imports scripts encoded in an export script created using **Export**. The export script to import must be accessible on your local file system.

5.5 Creating a SQL Script

Learn about different ways of creating a SQL script.

You can create a script in the Script Repository by:

- Creating a SQL Script in the Script Editor (page 5-4)
  Learn how to create SQL script in the Script Editor.
- Uploading a SQL Script (page 5-5)
  Learn how to upload a SQL script from your local file system.

5.5.1 Creating a SQL Script in the Script Editor

Learn how to create SQL script in the Script Editor.

To create an SQL script in the Script Editor:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.
2. Click the **Create** button.

   The Script Editor appears.
3. In Script Name, enter a name for the script. Script name extensions are optional.
4. Enter the SQL statements, PL/SQL blocks you want to include in your script.

   Remember that SQL*Plus commands are ignored at runtime.
5. Click **Create**.

   The SQL Scripts page appears listing your newly saved script.

5.5.2 Uploading a SQL Script

Learn how to upload a SQL script from your local file system.

To upload a script from your local file system:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. Click the **Upload** button.

   The Upload Script dialog appears.

3. For Upload Script:

   a. File - Browse to locate the script you want to upload.

   b. Script Name - Optionally enter a new name in the Script Name field. This name displays in the Script Repository.

   c. File Character Set - Select the character set in which the script is encoded.

   d. Click **Upload**.

   The SQL Scripts page appears listing your newly uploaded script.

> **Tip:**
>
> The script is parsed during runtime. When parsed, ignored statements such as SQL*PLUS commands are listed. Any invalid SQL is identified only in results. If a script of the same name exists, you are prompted to rename it.

5.6 Using the Script Editor

You use the Script Editor to add content to a new script, to edit existing scripts, and to run and delete scripts in the script repository.

- Editing an Existing Script (page 5-5)
  Learn how to edit an existing SQL script.
- Script Editor Controls (page 5-6)
  Controls and buttons available within the Script Editor.

5.6.1 Editing an Existing Script

Learn how to edit an existing SQL script.

To edit a SQL script:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. You can load a script into the editor as follows:

- In Icons view, click the script icon.
- In Report view, click the **Edit** icon.

The Script Editor appears.

3. Edit the script.

   Note that new lines are automatically indented to the previous line start column. Other features of the Script Editor include:

   - **Undo** - Use to undo the last edit.
   - **Redo** - Use to reapply the last undone edit.
   - **Find** - Click the **Find** icon (which resembles a magnifying glass) to perform a basic search.
   - **Replace** - Click the **Replace** icon (which resembles a two-sided arrow) to replace existing code.
   - **Cut, Copy, and Paste** - Use standard edit controls to cut, copy and paste content in the Script Editor.
   - **Auto Complete** - Click the **Auto Complete** icon to complete the command.

4. Click **Apply Changes** to save your script.

   The SQL Scripts page appears.

You can test your script during editing by running the script to reveal errors.

5.6.2 Script Editor Controls

Controls and buttons available within the Script Editor.

Table 5-1 (page 5-6)describes the buttons and controls available within the Script Editor

Table 5-1 Buttons and Controls within the Script Editor

Button	Descriptions
Cancel	Cancel the editing session and exit the Script Editor without saving changes made since the last save.
Download	Saves a copy of the current script to your local file system. Enter a name for the script on your local file system and a directory path.
Delete	Removes the current script from the Script Repository.
Save	Saves the currently script. Save is disabled if you edit the Script Name. Instead use **Apply Changes**.
Apply Changes	Saves the script and returns to the SQL Scripts report.
Run	Submits the script for execution.
Undo (Ctrl+Z)	Removes, or undoes, the most recent line edit made in the Script Editor.
Redo (Ctrl+Y)	Repeats the most recent line edit made in the Script Editor.
Find (Ctrl+F)	Click the Find icon (which resembles a magnifying glass) to perform a basic search.

Table 5-1 (Cont.) Buttons and Controls within the Script Editor

Button	Descriptions
Replace (Ctrl+Shift+F)	Click the Replace icon (which resembles a two-sided arrow) to replace existing code.

5.7 Deleting a SQL Script

You can delete scripts from the Script Repository by deleting selected scripts from the SQL Scripts page, or deleting the current script in the Script Editor.

- Deleting Scripts from the SQL Scripts Page (page 5-7)
  Learn how to delete scripts from the SQL Scripts page.
- Deleting a Script in the Script Editor (page 5-7)
  Learn how to delete a script in the Script Editor.

5.7.1 Deleting Scripts from the SQL Scripts Page

Learn how to delete scripts from the SQL Scripts page.

To delete scripts from the SQL Scripts page:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.
2. Click the **View Report** icon.

   Reports view appears.
3. Select the scripts to be deleted.

   To select individual scripts, click the check box to the left of the script name. To select all scripts visible in the current page, click the check box in the column heading.
4. Click **Delete Checked** to permanently remove the selected scripts from the Script Repository.
5. When prompted to confirm the delete action, click **OK**.

5.7.2 Deleting a Script in the Script Editor

Learn how to delete a script in the Script Editor.

To delete a script in the Script Editor:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.
2. Open the script you want to delete in the Script Editor.
3. Click **Delete** to permanently remove the script from the Script Repository.
4. When prompted to confirm the delete action, click **OK**.

5.8 Copying a SQL Script

You can copy a script in the Script Repository by saving it with a new name.

To copy a script:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.
2. Open the script into the Script Editor.
3. In Script Name, enter a new script name.
4. Click **Apply Changes**.

   The SQL Scripts page appears listing the new script.

5.9 Executing a SQL Script

You can execute scripts stored in the Script Repository. You can submit a script for execution either from the Script Editor or from the SQL Scripts page.

When you submit a script for execution, the Run Script page appears. It displays the script name, when it was created and by whom, when it was last updated and by whom, the number of statements it contains, and its size in bytes. It also lists unknown statements such as SQL*Plus commands that are ignored during execution.

If a script does not contain any runnable statements, the Run control does not display.

- Executing a SQL Script in the Script Editor (page 5-8)
  Learn how to execute a script in the Script Editor.
- Executing a SQL Script from the SQL Scripts Page (page 5-9)
  Learn how to execute a script from the SQL Scripts page.

> **See Also:**
>
> "About Long Operations Page (page 4-58)"

5.9.1 Executing a SQL Script in the Script Editor

Learn how to execute a script in the Script Editor.

To execute a script in the Script Editor:

1. Open the script you want to execute in the Script Editor.
2. Click **Run** in the Script Editor.

   The Run Script page appears. This page displays information about the script and lists statements in error preventing execution, or statements such as SQL*Plus commands that are ignored when the script is executed.

Available options on the Run Script page include:

- **Cancel** - returns you to the SQL Scripts page without executing the script.
- **Run Now** - to submit the script for execution.
- **Run in Background** - runs the script using `DBMS_SCHEDULER` rather than submitting for execution. This option is useful for long running scripts.

3. Click **Run Now** to submit the script for execution.

   The Manage Script Results page appears listing script results.

4. To view script results, click **View Results**.

5.9.2 Executing a SQL Script from the SQL Scripts Page

Learn how to execute a script from the SQL Scripts page.

To execute a script from the SQL Scripts page:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. Click the **View Report** icon.

   Reports view appears.

3. In the far right column, click the **Run** icon for the script you want to execute.

   > **Tip:**
   >
   > If the script does not contain any runnable statements, an error icon displays.

   The Run Script page appears. This page displays information about the script and lists statements in error preventing execution, or statements such as SQL*Plus commands that are ignored when the script is executed.

   Available options on the Run Script page include:

   - **Cancel** - returns you to the SQL Scripts page without executing the script.
   - **Run Now** - to submit the script for execution.

4. Click **Run Now** to submit the script for execution.

   The Manage Script Results page appears listing available results for the script.

5. Click the **View Results** icon in the far right column.

5.10 Viewing SQL Script Results

You use the Manage Script Results page to view and delete script results.

To view script results:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

The SQL Scripts page appears.

2. On the Tasks list, click **Manage Results**.

   The Manage Script Results page appears, listing available results for the script.

3. Click the **View Results** icon in the far right column.

   The Results page appears.

   The Results page displays the script name and status (Complete, Canceled, Executing or Submitted), and lists the statements executed.

4. On the Results page, select a View option and click **Go**:

   - **Detail** - Displays the Results in a report.
   - **Summary** - Displays a Summary page. From the Rows list, select the number of Rows to display. Select a column heading to sort the listed values by that column.

5. Click **Edit Script** to load the script into the Script Editor.

5.11 Exporting and Importing SQL Scripts

You can transfer scripts from your current Script Repository to a Script Repository in a different workspace by using the Export and Import tasks.

Exported scripts are encoded in a single file on your local file system. Once exported, you then log in to another workspace and import the file. During import, the file is run to re-create the scripts in the current Script Repository.

By default, the Export SQL Scripts page lists all scripts in the Script Repository created by the current user. There are two panes on the Export SQL Scripts page, the Scripts pane and the Scripts to Export pane. You use the Scripts pane to select scripts to export. Then, you use the Scripts to Export pane to finalize the scripts to export, to choose a name for the export script, and to save the selected scripts in the export script on your local file system. You use the Import Scripts pane to select the export script containing the scripts to import.

- Copying Scripts to an Export Script (page 5-10)
  Learn how to copy scripts to an export script.
- Importing Scripts (page 5-11)
  Learn how to import the listed scripts into the current Script Repository.

5.11.1 Copying Scripts to an Export Script

Learn how to copy scripts to an export script.

To copy scripts to an export script:

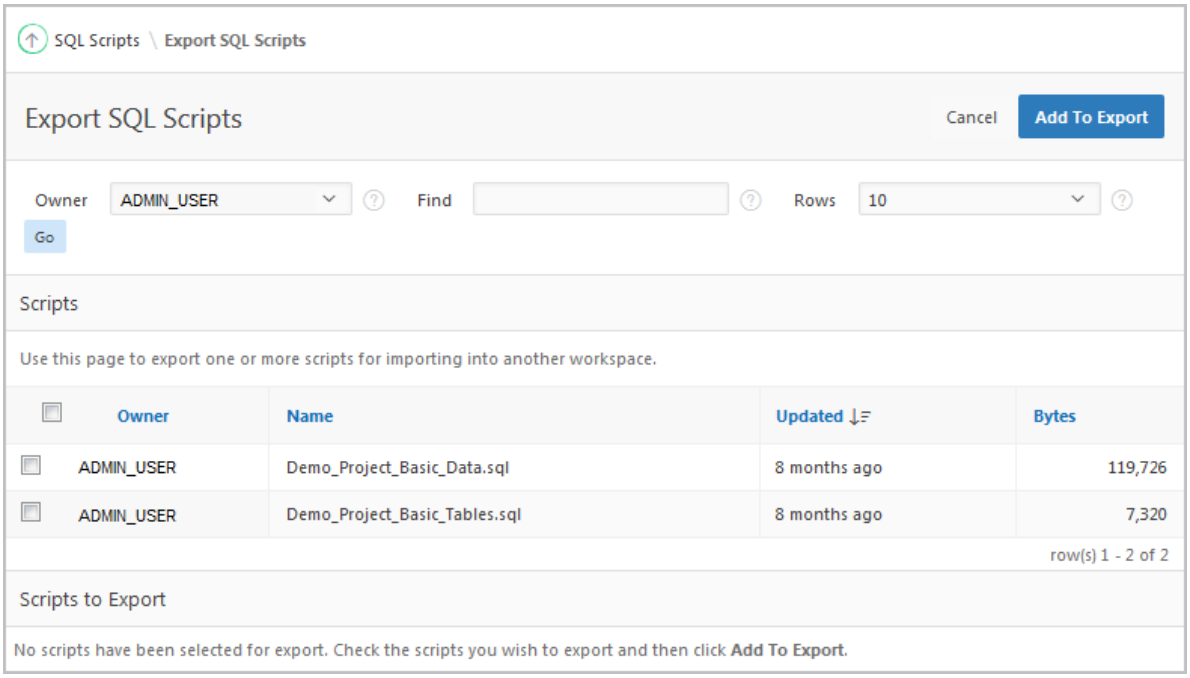
1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. On the Tasks list, click **Export**.

   The Export SQL Scripts page appears.

3. Select the scripts you want to export. To select all displayed scripts, click the column head check box.



4. Click **Add to Export**.

   The selected scripts are added to the list of scripts in the Scripts to Export pane.

5. Enter a name for the export script in the File Name field.

   The default script name is `workspace_name_script.sql`.

6. Click **Export All** to export the scripts to the export script.

   You are prompted to enter the directory where you want to save the export script.

5.11.2 Importing Scripts

Learn how to import the listed scripts into the current Script Repository.

To import scripts:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. On the Tasks list, click **Import**.

   The Import Scripts pane appears.

3. Browse to export script you want to import to the Script Repository.

4. Click **Next** to list the scripts in the export script.

   The Import Scripts page appears. The Action column indicates whether the imported script is new, or whether it replaces an existing script of the same name.

5. Click **Import Script(s)** to import the listed scripts into the current Script Repository.

   The SQL Scripts page appears listing the imported scripts.

5.12 Viewing Script and Result Quotas

You can view the script limits in the current workspace on the Script Quotas page.

To view the Script Quotas page:

1. On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.

   The SQL Scripts page appears.

2. On the Tasks list, click **Show Quotas**.

   The Script Quotas page appears and displays the following limits:

   - Result Quota in Bytes:
     - **Maximum Script Result Size** - The maximum size in bytes of a single script result. This size is set by an Oracle Application Express instance administrator and cannot be changed from within the workspace.
     - **Quota for All Script Results** - The maximum size in bytes of all results in this workspace. This size is set by an Oracle Application Express instance administrator and cannot be changed from within the workspace.
     - **Used** - The number of bytes currently used in this workspace.
     - **Free** - The number of bytes currently free in this workspace.
     - **Quota** - A usage bar illustrating the percentage of quota currently used.
   - Script Quota in Bytes:
     - **Maximum Script Size** - The maximum size in bytes of a single script. The size is set by the Oracle Application Express instance administrator and cannot be changed within the Workspace.

3. Click **Cancel** to return to the SQL Scripts page.

6

Enabling Data Exchange with RESTful Services

RESTful Services enable the declarative specification of RESTful services used to access the database.

RESTful Services enable the creation of a set of standards-based API's on top of database objects available for a workspace. Without connecting directly to the underlying data source these API's allow external systems to interact securely with the data by querying, inserting, updating or deleting.

Prior to release 18.1, the definitions of RESTful services created within Oracle Application Express were stored within the metadata tables of the core Application Express schema. With release 18.1, Oracle recommends migrating of all RESTful services to the Oracle REST Data Services (ORDS) repository. This makes it easier to manage RESTful services in a single place using a multitude of tools, including Application Express, SQL Developer, SQL Plus, SQLcl, and so on.

This section introduces you to the concepts of RESTful services and describes how to create, edit and delete RESTful Services using Application Express.

> **Note:**
>
> Your workspace administrator controls whether RESTful services are enabled. If disabled, the RESTful Services icon does not display.

- What Is REST? (page 6-2)
  Representational State Transfer (REST) introduces the concepts of REST and discusses the difference between publishing and consuming REST services.
- About Oracle RESTful Services (page 6-3)
  Discusses Oracle's implementation of REST and the terminology used to describe the various components of RESTful services.
- RESTful Service Requirements (page 6-4)
  Outlines the requirements for exposing RESTful services to external systems.
- How to Access RESTful Services (page 6-6)
  Access RESTful Services from SQL Workshop.
- How to Create the ORDS Based RESTful Services Module Example (page 6-10)
  Learn how to create the ORDS Based RESTful Services module example.
- Managing RESTful Service Modules (page 6-18)
  Learn how to install sample RESTful service module, You can create ,edit, and delete an ORDS RESTful service module.

- Managing Resource Templates (page 6-26)
  Resource templates are a means of grouping URIs and associating them with a particular RESTful Service Module.
- Managing Resource Handlers (page 6-28)
  A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method.

> **See Also:**
>
> "Controlling RESTful Services for a Workspace" in *Oracle Application Express Administration Guide*

6.1 What Is REST?

Representational State Transfer (REST) introduces the concepts of REST and discusses the difference between publishing and consuming REST services.

Representational State Transfer (REST) is a way of providing interoperability between computer systems over the Internet. You can define the RESTful services to allow the querying and manipulation of data without the need for direct access to the underlying data store.

A service is described as RESTful when it conforms to the tenets of REST. Although a full discussion of REST is outside the scope of this document, a RESTful Service has the following characteristics:

- Data is modeled as a set of resources. Resources identified by URLs and accessed over the http or https web protocols.
- A small set of operations are used to manipulate resources (for example, PUT, POST, GET, DELETE).
- REST services are, by nature, stateless as there is no connection kept open between the client and serving computer systems, and no client context is stored between requests.
- Requests to a RESTful service always elicit a response. This response is in the form of XML, JSON, HTML, or some other defined format.
- Depending upon the operation, responses provide details of some sort of alteration to the underlying data, error messages, and hypertext links to other related resources.

By using a stateless protocol and standard operations, REST resources are created to optimize for fast performance, reliability, portability, and growth. This is achieved by creating reusable underlying components that can be managed and updated without affecting the system, even while REST is running.

RESTful services, is divided into two sections: Consuming RESTful services and Providing RESTful services.

While consuming RESTful Services uses the standard HTTP operations, processing the data returned through a RESTful service is different. Oracle Application Express 18.1 provides a comprehensive set of components to make consuming RESTful services easier.

6.2 About Oracle RESTful Services

Discusses Oracle's implementation of REST and the terminology used to describe the various components of RESTful services.

This section introduces RESTful services as implemented by Oracle REST Data Services and provides clarification on the terminology used within Application Express and other Oracle tools to reference the different components that make up a RESTful Service.

Common terms are:

- **Schema Alias**: When enabling an Oracle schema for use with RESTful services, an alias is required. The case sensitive Schema Alias is used to form the URL and appended to the server path prior to the Base Path and URI pattern.
- **Resource Module**: A container that is used to group related RESTful Services together.
- **Base Path:**
  A case sensitive base path that is used to access a RESTful Service. This base path for the module is appended to the server path and the Schema Alias for the underlying Oracle schema. The base path must be unique within the schema.
- **RESTful Service**: An HTTP web service that conforms to the tenets of the RESTful architectural style.
- **Resource Template**: An individual RESTful Service that is able to service requests for some set of Universal Resource Identifiers (URIs). The URI Pattern of the Resource Template defines the set of URIs.
- **URI pattern**: A case sensitive path used to access the specific resource and can be either a Route Pattern or a URI Template. The URI Pattern for the resource is appended to the server path and Resource Module Base Path. The URI Pattern may also include a bind variable appended after a forward slash. This allows a value to be passed to the service as part of the URI. The bind variable syntax prefixes the variable name with a colon (:).
  - **Route Pattern**: A pattern that focuses on decomposing the path portion of a URI into its component parts. For example, a pattern of `/:object/:id?` will match `/emp/101` (matches a request for the item in the `emp` resource with id of `101`) and also match `/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` matches any URI whose path begins with `employees/`, such as `employees/2560`.
- **Resource handler**: Provides the logic required to service a specific HTTP operation for a specific resource template.
- **HTTP operation**: HyperText Transport Protocol (HTTP) defines standard methods that can be performed on resources: GET, POST, PUT, and DELETE.

In general, for each RESTful service you want your application to provide, you must configure a Resource Module and a child Resource Template. For each method associated with the same Resource Template you must define a Resource Handler.

For instance, if you want to provide a method to return data and another to store data, you must define a Resource Handler for each operation.

A Resource Module is a container used to group a set of related RESTful Services together. A Resource Module not only provides a way to identify the group uniquely, but also defines the unique Base Path used on a URL to access the set of services defined within the module.

For example, a Resource Module that allows you to access information about employees may be named `oracle.example.hr` and a typical base path value for the service is `/hr/` .

A Resource Template exists within the confines of a Resource Module and defines an individual service that can be called. Each Resource Template defines a URI Pattern where it can be reached and implements at least one Resource Handler. A given Resource Template may only implement one of each type of Resource Handler (GET, POST, PUT, DELETE) but may choose not to implement all of them.

Each Resource Handler implements one (and only one) HTTP operation. It is important to understand how Resource Handler implementations are mapped to traditional Create, Read, Update, Delete (CRUD) operations.

- **GET** - Retrieves a representation of a resource, equivalent to a SQL SELECT statement.
- **POST** - Creates a new resource or adds a resource to a collection, equivalent to a SQL INSERT statement.
- **PUT** - Updates the values an existing resource, equivalent to a SQL UPDATE statement.
- **DELETE** - Deletes an existing resource, equivalent to a SQL DELETE statement.

A Resource Template is considered functionally incomplete if it does not implement at least one Resource Handler.

An example of a functionally complete Resource Template might be one that implements the ability to retrieve a specific employee's details by passing in the employee id. The Resource Templates URI Pattern looks like `employees/:id`. A Resource Handler is implemented using the GET operation and the following logic:

```
select empno, ename, dept from emp where empno = :id
```

When calling the RESTful service, access the service using the server path, base path and URI template and providing a static value for the employee id as:

```
http://<<server_address>>/<<schema_alias>>/hr/employees/7978
```

When this particular URL is called using an HTTP GET, the service returns the result of the SQL Statement listed above. The format in which the data is returned depends upon the full definition of the Resource Handler.

6.3 RESTful Service Requirements

Outlines the requirements for exposing RESTful services to external systems.

To provide RESTful services from within Oracle Application Express, you must have Oracle REST Data Services release 2.0 or later as your web listener and enabled RESTful Services either at the workspace or instance-level by an administrator.

In order for the RESTful Service utility to be available and function properly, the following requirements must be met:

- Your configuration uses Oracle REST Data Services release 2.0 or later as your web listener.
- An administrator has enabled RESTful Services either at the workspace or instance-level.
- The Oracle REST Data Services version 2.0 or higher must be used.

> **Tip:**
>
> The Oracle XML DB HTTP Server with the embedded PL/SQL gateway and Oracle HTTP Server with `mod_plsql` do not support RESTful Services.

Versions of Oracle Application Express prior to 18.1 maintains the metadata about APEX based RESTful Services defined in the core Oracle Application Express metadata repository.

As Oracle moves toward a unified RESTful Services repository based on Oracle REST Data Services (ORDS), you need to allow for migration of RESTful Services previously defined in Application Express to the ORDS based repository.

Oracle Application Express release 18.1 requires Oracle REST Data Services release 17.4.1 or later to provide full integration with the ORDS Metadata repository. Functionality of the REST Services section of Application Express provides for the following scenarios.

- When running Oracle Application Express release 18.1 with a version Oracle REST Data Services prior to release 17.4.1, only the traditional Oracle Application Express based REST Services console is available. You can continue to create and maintain Oracle Application Express based RESTful services and they will be stored in the traditional Oracle Application Express metadata repository.
- When running Oracle Application Express release18.1 with a version of Oracle REST Data Services release 17.4.1 or later and the workspace currently contains Oracle Application Express based RESTful services.
  - You will be given read-only access to the traditional Oracle Application Express based RESTful Service pages. The current services continue to work , however, you are not be able to create new or edit existing Application Express based RESTful services.
  - Clicking **Migrate to ORDS RESTful Services** button migrates all existing Oracle Application Express based RESTful services within the workspace to the ORDS metadata repository.
- When running Oracle Application Express release 18.1 with a version of Oracle REST Data Services release 17.4.1 or later and the workspace currently does not contain any Oracle Application Express based RESTFUL services, you are navigated to the new Oracle REST Data Services (ORDS) Based Restful Services pages and will not have access to the traditional Oracle Application Express based RESTful services screens.

6.4 How to Access RESTful Services

Access RESTful Services from SQL Workshop.

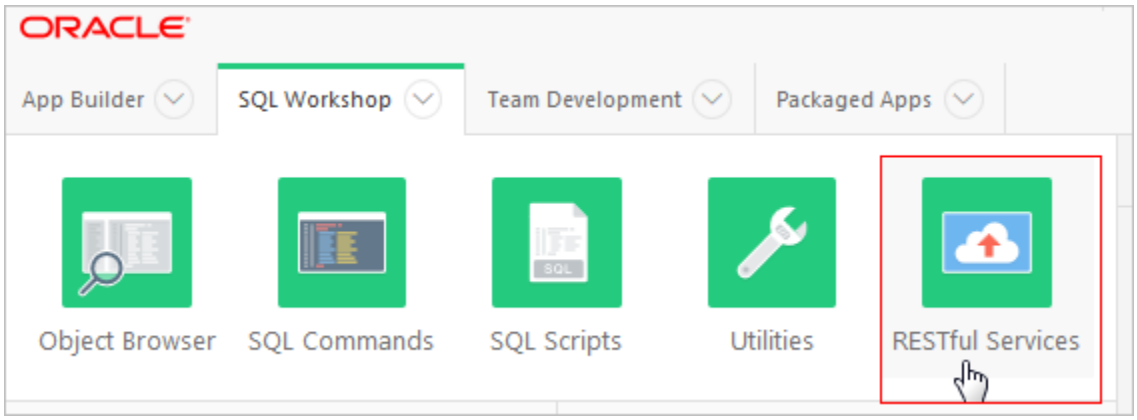
- Accessing RESTful Services (page 6-6)
  Access RESTful Services from the Workspace home page by clicking SQL Workshop and then RESTful Services.
- Registering Schema with ORDS (page 6-7)
  If you are accessing ORDS Restful Services in a new workspace, you must register the schema with ORDS RESTful Data Services.
- About ORDS RESTful Services Dashboard (page 6-8)
  Learn about ORDS RESTful Services dashboard.

6.4.1 Accessing RESTful Services

Access RESTful Services from the Workspace home page by clicking SQL Workshop and then RESTful Services.

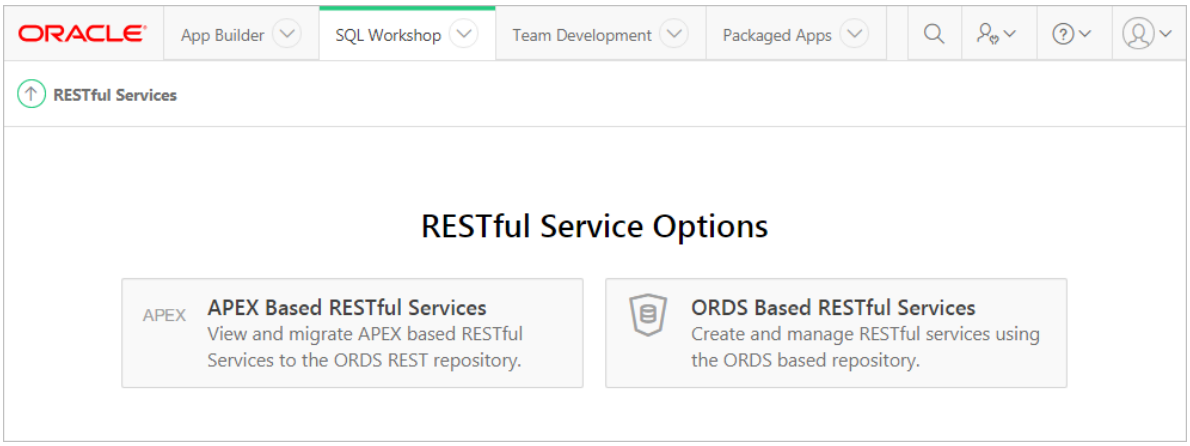
To access RESTful Services:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **RESTful Services**.



The RESTful Services home page appears.

In a new workspace, the bottom of the page displays two regions: **APEX Based RESTful Services** and **ORDS Based RESTful Services**. Click these regions to create an APEX Based RESTful Services or ORDS Based RESTful Services.



6.4.2 Registering Schema with ORDS

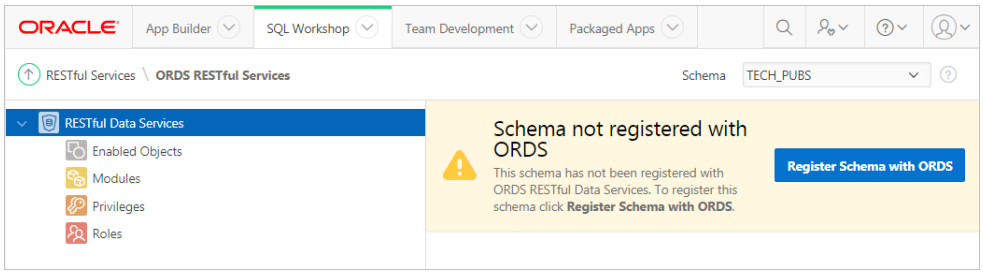
If you are accessing ORDS Restful Services in a new workspace, you must register the schema with ORDS RESTful Data Services.

To register a schema with ORDS RESTful Data Services:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **RESTful Services**.

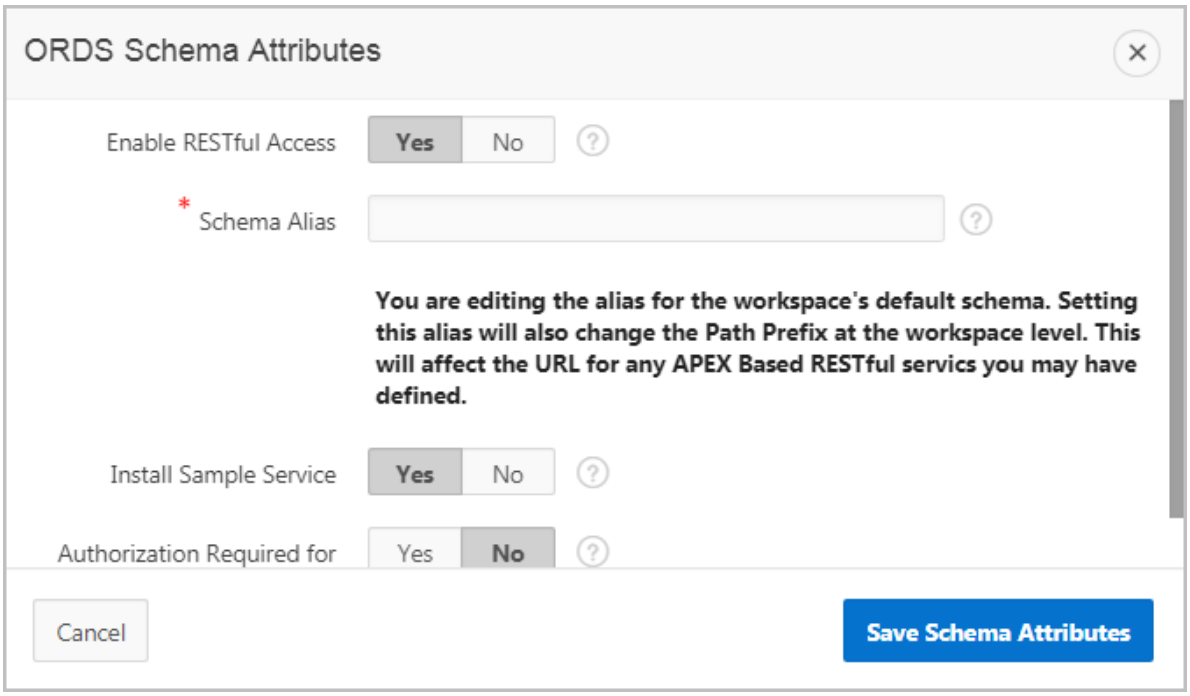
   The RESTful Services home page appears.
3. Click **ORDS Based RESTful Services**.

   ORDS RESTful Services page appears. If your schema is not registered with ORDS RESTful Data Services, the following message appears.



4. Click **Register Schema with ORDS**.

   ORDS Schema Attributes dialog appears.



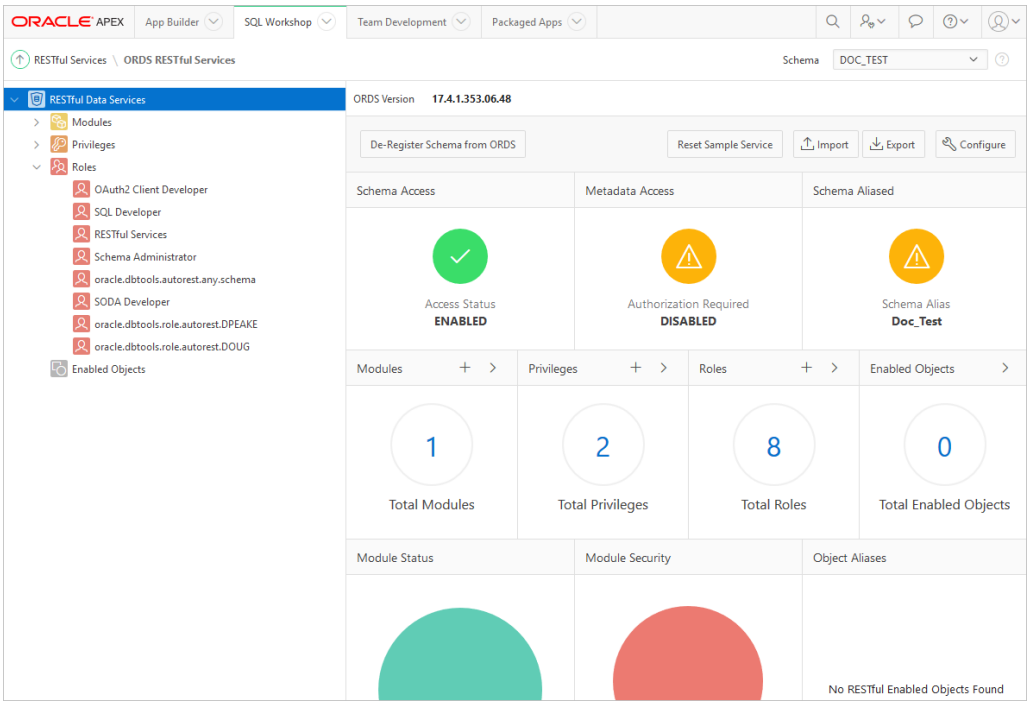
5. Enter the attribute details:
   a. **Enable RESTful Access** - Click **Yes** to enable or click **No** to disable the access to any REST services defined in the schema.
   b. **Schema Alias** - Enter the name of the schema alias to be used in the formation of the URL referencing any RESTful service within the schema.
   c. **Install Sample Service** - Click **Yes** or **No**. Clicking **Yes** installs sample `oracle.example.hr` RESTful service.
   d. **Authorization Required for Metadata Access** - Click **Yes** or **No**. Clicking **Yes** requires Oracle REST Data Services user authorization before allowing access to the ORDS metadata catalog of the schema.
6. Click **Save Schema Attributes**.

   ORDS RESTful page appears with a message Schema enabled for use with ORDS RESTful Services and sample RESTful Service successfully installed.

6.4.3 About ORDS RESTful Services Dashboard

Learn about ORDS RESTful Services dashboard.

Upon registering the schema with ORDS, the ORDS RESTful services page appears displaying the ORDS version.



The following table details the information available in the ORDS RESTful Services Dashboard:

Table 6-1 ORDS RESTful Services Dashboard Parameters

Name	Description
ORDS Version	Version of ORDS.
De- Register Schema from ORDS	Click to de-register schema.
Reset Sample Service	Click to reset the installed sample service.
Import	Click to import ORDS based RESTful service definitions from a SQL script generated by SQL Developer or Oracle Application Express.
Export	Click to export ORDS based RESTful Service metadata to SQL script file format.
Configure	Click to configure ORDS schema attributes.
Schema Access	Displays whether access to the RESTful services defined within the schema is enabled or disabled.
Metadata Access	Displays whether Oracle REST Data Services should require user authorization before allowing access to the ORDS metadata catalog of this schema.
Schema Aliased	Displays alias name for the current schema.
Modules	Displays total number of modules defined within the schema. Click to view list of defined modules.

Table 6-1 (Cont.) ORDS RESTful Services Dashboard Parameters

Name	Description
Privileges	Displays total number of privileges defined. Click to view list of defined privileges.
Roles	Displays total number of roles associated. Click to view the associated roles.
Enabled Objects	Displays the total number of REST enabled database objects in the schema. Click to view list of REST enabled objects.
Module Status	Displays the status of published and unpublished modules.
Module Security	Displays the status of protected and unprotected modules.
Object Aliases	Displays the status of Aliased and Unaliased objects.

6.5 How to Create the ORDS Based RESTful Services Module Example

Learn how to create the ORDS Based RESTful Services module example.

Oracle Application Express workspace by default has an example RESTful Service Module, `oracle.example.hr`. You can install the sample service while registering the schema with ORDS.

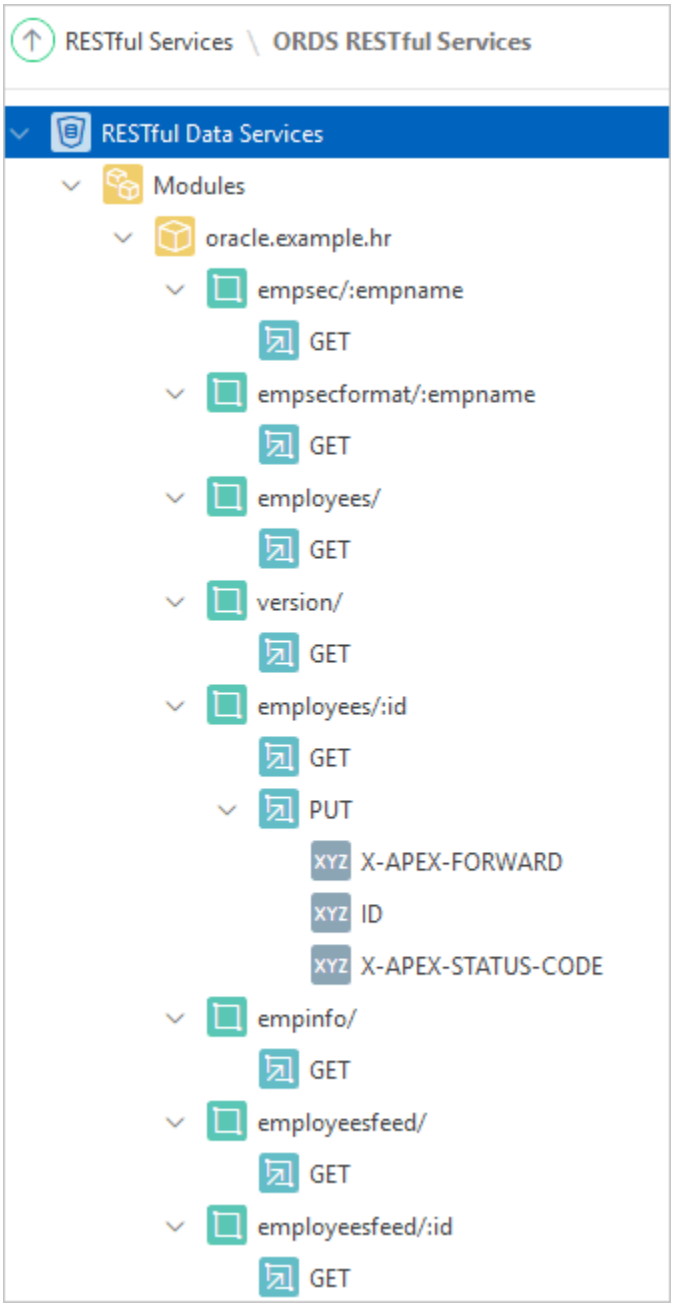
- About the Example RESTful Service Module (oracle.example.hr) (page 6-11)
  In ORDS Based RESTful Services, you can choose to install the sample service `oracle.example.hr`, when registering schema with ORDS.
- Returning the Result Set Based on a PL/SQL Block (empsecformat/:empname) (page 6-13)
  You can add and test your own resource template that uses a PL/SQL block to format results.
- Returning the Result Set in JSON Format with Pagination Set (employees/) (page 6-14)
  You can add and test your own resource template that retrieves data in JSON format and demonstrates pagination.
- Retrieving Data Based on a Parameter (employees/:id) (page 6-15)
  You can add and test your own resource template that retrieves employee data based on a parameter.
- Returning the Result Set in CSV Format (empinfo/) (page 6-16)
  You can build and test your own example RESTful Service Module that retrieves data in CSV format.
- Returning the Result Set as a Feed (employeesfeed/) (page 6-17)
  You can add and test your own resource template that displays return data as a feed.

6.5.1 About the Example RESTful Service Module (oracle.example.hr)

In ORDS Based RESTful Services, you can choose to install the sample service `oracle.example.hr`, when registering schema with ORDS.

This module provides example resource templates and resource handlers that implement several different methods to retrieve and display employee information from the `emp` table. To access the `oracle.example.hr` module, see " Accessing RESTful Services".

The `oracle.example.hr` module consists of several resource templates. Each resource template demonstrates a different way of retrieving information and formatting the returned results.



The following results are expected for each `oracle.example.hr` resource template when the respective GET resource handler is invoked:

- `empsec/:empname` - Employee information for a specified employee name, `empname`, is returned and displayed in JSON.
- `empsecformat/:empname` - Employee information for a specified employee name, `empname`, is returned and formatted based on the execution of a PL/SQL block.
- `employees/` - Employee information from the employees `emp` table is displayed in JSON format with Pagination Size set to 7. A first, previous and next link is provided to page through the results.
- `version/`- Returns the version of the RESTful service module.

- `employees/:id` - Employee information from the employees `emp` table for a specified `id` is returned.
- `empinfo/` - Employee information from the employees `emp` table is returned in CSV format.
- `employeesfeed/` - Employee information from the employees `emp` table is returned as a JSON Feed.
- `employeesfeed/:id` - Employee information from the employees `emp` table for a specified `id` is displayed in JSON format.

The following result is expected for the `employees/:id` resource template when the PUT resource handler is invoked:

- `employees/:id` - Employee information from the employees `emp` table for a specified id is updated.

6.5.2 Returning the Result Set Based on a PL/SQL Block (`empsecformat/:empname`)

You can add and test your own resource template that uses a PL/SQL block to format results.

The `oracle.example.hr` module includes the `empsecformat/:empname` resource template to demonstrate a RESTful Service that returns the results of an SQL Query formatted by a PL/SQL block.

To add and test your own resource template that uses a PL/SQL block to format results:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click the RESTful Service Module you want to add the resource template to.

   The Module Definition page appears.
4. Under Resource Templates block on the left panel, click **Create Template**.

   The ORDS Template Definition page appears.
5. For URI Template, enter `empsecformat/:empname`.
6. Click **Create Template**.
7. Under the `empsecformat/:empname` resource template, click **Create Handler**.

   The ORDS Handler Definition page appears.
8. Make the following selections and entries:
   - **Method** - Select **GET**.
   - **Source Type** - Select **PL/SQL**.
   - **Source** - enter the following:

     ```
     DECLARE
       prevdeptno      number;
     ```

```
  total_rows     number;
  deptloc        varchar2(20);
  deptname       varchar2(20);
  CURSOR         getemps is select * from emp
                             start with ename = :empname
                            connect by prior empno = mgr
                             order siblings by deptno, ename;
BEGIN
  sys.htp.htmlopen;
  sys.htp.headopen;
  sys.htp.title('Hierarchical Department Report for Employee '||:empname);
  sys.htp.headclose;
  sys.htp.bodyopen;

  for emprecs in getemps
  loop

      if l_employee.deptno != prevdeptno or prevdeptno is null then
          select dname, loc
           into deptname, deptloc
           from dept
          where deptno = l_employee.deptno;

         if prevdeptno is not null then
             sys.htp.print('</ul>');
         end if;
        sys.htp.print('Department ' || apex_escape.html(deptname) || '
located in ' || apex_escape.html(deptloc) || '<p/>');
        sys.htp.print('<ul>');
      end if;

      sys.htp.print('<li>' || apex_escape.html(l_employee.ename) || ', '  ||
apex_escape.html(l_employee.empno) || ', ' ||
                        apex_escape.html(l_employee.job) || ', ' ||
apex_escape.html(l_employee.sal) || '</li>');

      prevdeptno := l_employee.deptno;
      total_rows := getemps%ROWCOUNT;

   end loop;
   if total_rows > 0 then
       sys.htp.print('</ul>');
   end if;
   sys.htp.bodyclose;
   sys.htp.htmlclose;
END;
```

9. Click **Create Handler**.

   The GET resource handler appears under the `empsecformat/:empname` resource template on the left panel.

6.5.3 Returning the Result Set in JSON Format with Pagination Set (`employees/`)

You can add and test your own resource template that retrieves data in JSON format and demonstrates pagination.

The `oracle.example.hr` module includes the `employees/` resource template to demonstrate a RESTful Service that returns data in JSON format. This example is

returning the Result Set in JSON Format with Pagination Set Using Pagination Bind Variables. The Pagination bind variables are `:row_offset` and `:row_count`.

For other example resource templates that return results in JSON format see the `empsec/{empname}` resource template for the `oracle.example.hr` Service Module.

To add and test your own resource template that retrieves data in JSON format and demonstrates pagination:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click the RESTful Service Module you want to add the resource template to.

   The Module Definition page appears.
4. Under Resource Templates block on the left panel, click **Create Template**.

   The ORDS Template Definition page appear.
5. For URI Template, enter `employees/`.
6. Click **Create Template**.

   The `employees/` resource template displays on the left panel.
7. Under the `employees/` resource template, click **Create Handler**.

   The ORDS Handler Definition page appear.
8. Make the following selections and entries:
   - **Method** - Select **GET**.
   - **Source Type** - Select **Query**.
   - **Format** - Select **JSON**.
   - **Source** - Enter the following:

     ```
     select empno "$uri", rn, empno, ename, job, hiredate, mgr, sal, comm, deptno
       from (
            select emp.*
                 , row_number() over (order by empno) rn
              from emp
            ) tmp
     ```
9. Click **Create Handler**.

   The GET Handler appears under the `employees/` resource template on the left panel.

6.5.4 Retrieving Data Based on a Parameter (`employees/:id`)

You can add and test your own resource template that retrieves employee data based on a parameter.

The `oracle.example.hr` module includes the `employees/:id` resource template to demonstrate a RESTful Service that returns one row of data in JSON format based on the specified `id`.

For other example resource templates that return results based on a specified parameter see the `employeesfeed/:id`, `empsec/:empname` and `empsecformat/:empname` resource templates for the `oracle.example.hr` Service Module.

To add and test your own resource template that retrieves employee data based on a parameter:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appear.
3. Click the RESTful Service Module you want to add the resource template to.

   The Module Definition page appears.
4. Under Resource Templates block on the left panel, click **Create Template**.

   The ORDS Template Definition page appear.
5. For URI Template, enter `employees/:id`.
6. Click **Create Template**.

   The `employees/:id` resource template displays on the left panel.
7. Under the `employees/:id` resource template, click **Create Handler**.

   The ORDS Handler Definition page appear.
8. Make the following selections and entries:
   - **Method** - select **GET**.
   - **Source Type** - select **Query One Row**. Executes a SQL Query returning one row of data into a JSON representation. This option is only available when the HTTP method GET has been selected
   - Pagination Size - Leave blank. In this case, there's no need to set pagination since only one record is retrieved.
   - **Source** - enter the following:

     ```
     select * from emp  where empno = :id
     ```
9. Click **Create Handler**.

6.5.5 Returning the Result Set in CSV Format (`empinfo/`)

You can build and test your own example RESTful Service Module that retrieves data in CSV format.

The `oracle.example.hr` module includes the `empinfo/` resource template to demonstrate a RESTful Service that returns data in CSV format. In this example, all the employee data from the `emp` table is returned by using a SQL Query.

To build and test your own example RESTful Service Module that retrieves data in CSV format:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click ORDS Based RESTful services.

The ORDS RESTful services page appear.

3. Click the RESTful Service Module you want to add the resource template to.

   The Module Definition page appears.

4. Under Resource Templates block on the left panel, Click **Create Template**.

   The ORDS Template Definition page appear.

5. For URI Template, enter `empinfo/`.

6. Click **Create Template**.

   The `empinfo/` resource template displays on the left panel.

7. Under the empinfo/ resource template, click **Create Handler**.

8. Make the following selections and entries:

   - **Method** - select **GET**.
   - **Source Type** - Select **Query**. Executes a SQL Query and transforms the result set into either a JavaScript Object Notation (JSON) or CSV representation, depending on the format selected. This option is only available when the HTTP method GET has been selected
   - **Format** - Select **CSV**.
   - **Source** - Enter the following:

     ```
     select * from emp
     ```

9. Click **Create Handler**.

6.5.6 Returning the Result Set as a Feed (`employeesfeed/`)

You can add and test your own resource template that displays return data as a feed.

The `oracle.example.hr` module includes the `employeesfeed/` resource template to demonstrate a RESTful Service that returns `empno` and `ename` values from the `emp` table and displays them as a feed.

A feed resource handler executes a SQL query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the result set must be a unique identifier for the row and is used to form a hyperlink of the form: path/to/feed/{id}, with the value of the first column being used as the value for {id}. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate resource template for the full representation of the resource should also be defined.

To add and test your own resource template that displays return data as a feed:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.

2. Click ORDS Based RESTful services.

   The ORDS RESTful services page appear.

3. Click the RESTful Service Module you want to add the resource template to.

   The Module Definition page appears.

4. Under Resource Templates block on the left panel, click **Create Template**.

   The ORDS Template Definition page appear.
5. For URI Template, enter `employeesfeed/`.
6. Click **Create Template**.
7. Under the `employeesfeed/` resource template, click **Create Handler**.

   The ORDS Handler Definition page appear.
8. Make the following selections and entries:
   - **Method** - select **GET**.
   - **Source Type** - select **Feed**.
   - **Pagination Size** - enter `25`.
   - **Source** - enter the following:

     ```
     select empno, ename from emp  order by deptno, ename
     ```
9. Click **Create Handler**.

6.6 Managing RESTful Service Modules

Learn how to install sample RESTful service module, You can create ,edit, and delete an ORDS RESTful service module.

A RESTful Service Module is a grouping of common templates, known as resource templates, under a common Uniform Resource Identifier (URI) prefix.

The specification of a URI prefix is optional. If defined for the RESTful Service Module, it is prepended to all resource templates. A priority value is used to choose between resource templates when their patterns both match a single request URI. In such cases, the resource template with the highest priority value is chosen.

- Installing a Sample RESTful Service Module with Schema ORDS not Enabled (page 6-19)
  Learn how to create a Sample RESTful Service Module with Schema ORDS not Enabled.
- Installing a Sample RESTful Service Module with Schema ORDS Enabled (page 6-20)
  Learn how to create a Sample RESTful Service Module with Schema ORDS Enabled.
- Creating an ORDS RESTful Service Module (page 6-21)
  Learn how to create an ORDS RESTful Service Module.
- Editing an ORDS RESTful Service Module (page 6-21)
  An ORDS RESTful Service Module can have associated resource templates and resource handlers. During the edit process for a particular RESTful Service Module, you can make changes to any of the associated resource templates and resource handlers.
- Deleting an ORDS RESTful Service Module (page 6-22)
  Deleting an ORDS RESTful service module removes the module along with all resource templates and resource handlers associated with this module.

- Managing an ORDS RESTful Service Privileges (page 6-22)
  Privileges to access RESTful Service Modules can be restricted in Application Express.
- Assigning Privileges to an ORDS RESTful Service Modules (page 6-23)
  Learn how to assign privileges to an ORDS RESTful Service Modules.
- Creating an ORDS RESTful Service Privilege (page 6-23)
  You can create an ORDS RESTful Service Privilege to restrict access to specified users for one or more RESTful Service Modules.
- Editing an RESTful Service Privilege (page 6-24)
  You can modify an ORDS RESTful Service Privilege. Changes to the specified Protected Modules will be reflected in the Required Privilege setting on the RESTful Service Module page for affected modules.
- Deleting an ORDS RESTful Service Privilege (page 6-24)
  You can remove an ORDS RESTful Service Privilege. After deleting the privilege, any RESTful Service Modules that were protected by this privilege will be unprotected until reassigned another privilege.
- Exporting an ORDS RESTful Service Module (page 6-25)
  An ORDS RESTful Service Module metadata can be exported to a file in SQL script format. The exported file can be imported on the same instance or another compatible instance of Application Express.
- Importing an ORDS RESTful Service Module (page 6-25)
  An ORDS RESTful Service Module metadata that has been exported to a file in SQL script format can be imported. The imported module must have been exported on the same instance or another compatible instance of Application Express.
- Generating a Swagger Document (page 6-26)
  Learn how to generate a Swagger document.

6.6.1 Installing a Sample RESTful Service Module with Schema ORDS not Enabled

Learn how to create a Sample RESTful Service Module with Schema ORDS not Enabled.

> **Note:**
>
> You can install a sample RESTful service only when you create new workspace.

To create a Sample RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appear.
3. Click **Register Schema with ORDS**.

The ORDS Schema Attributes page appears.

4. Enter the attribute details:
   - **Enable RESTful Access**- Click **Yes** to enable or click **No** to disable the access to any REST services defined in the schema.
   - **Schema Alias** - Enter the name of the schema alias to be used in the formation of the URL referencing any RESTful service within the schema.
   - **Install Sample Service** - Click **Yes**. Clicking **Yes** installs sample `oracle.example.hr` RESTful service.
   - **Authorization Required for Metadata Access** - Click **Yes**. Clicking **Yes** requires Oracle REST Data Services user authorization before allowing access to the ORDS metadata catalog of this schema.
5. Click **Save Schema Attributes**.

   The ORDS RESTful page appears. Sample RESTful Services appears under Modules.

6.6.2 Installing a Sample RESTful Service Module with Schema ORDS Enabled

Learn how to create a Sample RESTful Service Module with Schema ORDS Enabled.

> **Note:**
>
> You can install a sample RESTful service only when you create new workspace.

To create a Sample RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appear.
3. Click **Install Sample Service** on the right pane.

   The Sample service is installed and appears under RESTful Services Modules.

   > **Note:**
   >
   > Click **Reset Sample Service** button to reset the sample service. Click **Delete** button to delete the sample service from Module definition page.

6.6.3 Creating an ORDS RESTful Service Module

Learn how to create an ORDS RESTful Service Module.

To create an ORDS RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appears.
3. Click **Modules** under RESTful Data Services tree.
4. Click **Create Module** on the right pane.

   The ORDS Module Definition page appears.
5. Specify the following RESTful Service Module settings:
   - **Module Name** - Enter the name of the RESTful Service Module.
   - **Base Path** - Enter module base path.
   - **Is Published** - Select **Yes** to make this service available for use, otherwise select **No**.
   - **Pagination Size** - Specify the size of the pagination window. For database queries, this is the number of rows to return.
   - **Origins Allowed**(optional) - Specify a comma-separated list of origins that are permitted to access the resource template.
   - **Comments**(optional) - Enter comments.
6. Click **Create Module**.

   The ORDS Module Definition page appears.

6.6.4 Editing an ORDS RESTful Service Module

An ORDS RESTful Service Module can have associated resource templates and resource handlers. During the edit process for a particular RESTful Service Module, you can make changes to any of the associated resource templates and resource handlers.

To edit an ORDS RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click the name of the module you want to edit.

   The ORDS Module Definition page appears.

> **Note:**
>
> Use the RESTful Service tree on the left pane to view, edit and create resource templates and resource handlers.

3. To edit RESTful Service Module settings, make changes to the fields on the right pane under ORDS Module Definition page.
   - To edit a resource template:
     a. Select the resource template you want to edit on the RESTful Service tree on the left panel. The resource template options appear on the right panel.
     b. Make your changes.
     c. Click **Apply Changes**.
   - To edit a resource handler:
     a. Select the resource handler you want to edit on the RESTful Service tree on the left panel. The resource handler options appear on the right panel.
     b. Make your changes.
     c. Click **Apply Changes**.
4. Click **Apply Changes**.

6.6.5 Deleting an ORDS RESTful Service Module

Deleting an ORDS RESTful service module removes the module along with all resource templates and resource handlers associated with this module.

To delete an ORDS RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click the name of the module you want to delete.

   The ORDS Module Definition page appears.
3. Click **Delete**.
4. Click **OK** to confirm.

   The RESTful Services home page appears.

6.6.6 Managing an ORDS RESTful Service Privileges

Privileges to access RESTful Service Modules can be restricted in Application Express.

To protect an ORDS RESTful Service Modules by assigning privileges:

1. Create a RESTful Service Privilege .
2. Specify which RESTful Service Modules are protected by this privilege.

> **See Also:**
>
> - "Using Groups to Manage Application Express Users" in *Oracle Application Express Administration Guide*
> - "Creating an ORDS RESTful Service Privilege (page 6-23)"
> - "Assigning Privileges to an ORDS RESTful Service Modules (page 6-23)"
> - "Editing an RESTful Service Privilege (page 6-24)"
> - "Deleting an ORDS RESTful Service Privilege (page 6-24)"

6.6.7 Assigning Privileges to an ORDS RESTful Service Modules

Learn how to assign privileges to an ORDS RESTful Service Modules.

You can assign privileges to an ORDS RESTful Service Modules during the creation or editing of an ORDS RESTful Service Privilege, you can specify the module or modules protected by this privilege.

> **Note:**
>
> After a RESTful Service Privilege is deleted, all RESTful Service Modules protected by that privilege are unprotected.

6.6.8 Creating an ORDS RESTful Service Privilege

You can create an ORDS RESTful Service Privilege to restrict access to specified users for one or more RESTful Service Modules.

A RESTful Service Privilege can include users belonging to one or more user groups. RESTful Service Modules protected by this privilege display the privilege name for Required Privilege on the RESTful Service Module page.

To create an ORDS RESTful Service Privilege:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appears.
3. Under RESTful Data Services tree on the left pane, click **Privileges**.
4. Click **Create Privilege** on the right pane.

   The ORDS Privileges Definition page appears.
5. Specify the following in the ORDS Privileges Definition page:
   - **Name**- Enter the name of the RESTful Service Privilege.

- **Title**- Enter the name of the RESTful Service Privilege title.
- **Description**- Enter the description of the RESTful Service Privilege.
- **Comments**- Enter comments.
- **Roles**- Select roles assigned to the privilege.
- **Protected Modules**- Use the arrows to select the modules this privilege is assigned to.
- **Protected Resources**- Adding a list of protection patterns to the Interactive Grid.

6. Click **Create Privilege**.

6.6.9 Editing an RESTful Service Privilege

You can modify an ORDS RESTful Service Privilege. Changes to the specified Protected Modules will be reflected in the Required Privilege setting on the RESTful Service Module page for affected modules.

To edit a ORDS RESTful Service Privilege:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appears.
3. Under RESTful Data Services tree on the left pane, click **Privileges**.

   The ORDS Privileges Definition page appears.
4. Make modifications.
5. Click **Apply Changes**.

6.6.10 Deleting an ORDS RESTful Service Privilege

You can remove an ORDS RESTful Service Privilege. After deleting the privilege, any RESTful Service Modules that were protected by this privilege will be unprotected until reassigned another privilege.

To delete an ORDS RESTful Service Privilege:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful Services page appears.
3. Under RESTful Data Services tree on the left pane, click **Privileges**.

   The ORDS Privileges Definition page appears.
4. Click **Delete**.

6.6.11 Exporting an ORDS RESTful Service Module

An ORDS RESTful Service Module metadata can be exported to a file in SQL script format. The exported file can be imported on the same instance or another compatible instance of Application Express.

To export an ORDS RESTful Service module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appear.
3. Click **Export** on the right pane.

   The Export ORDS RESTful Services page appears.
4. For RESTful Service, perform one of the following:
   - To export one module, select the module name.
   - To export all modules, select **All Services**.
5. Click **Export**.

   The Opening dialog appears.
6. Select export options.
7. Click **OK**.

6.6.12 Importing an ORDS RESTful Service Module

An ORDS RESTful Service Module metadata that has been exported to a file in SQL script format can be imported. The imported module must have been exported on the same instance or another compatible instance of Application Express.

> **Note:**
>
> If the imported module contains a ORDS RESTful Service name that exists in this workspace, the existing ORDS RESTful Service is deleted and replaced with the definition from the imported module.

To import an ORDS RESTful Service module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click **Import** on the right pane.

   The Import ORDS RESTful Services page appears.

4. For Import File, browse to and select the import file.
5. For File Character Set, select the character set used by the import file.
6. Click **Import**.

   The ORDS RESTful Services page appears with the imported data.

> **WARNING:**
>
> An error message appears that the file is of the wrong type, when you export an APEX Based RESTful Services and try to import it into the Import ORDS RESTful Services page.

6.6.13 Generating a Swagger Document

Learn how to generate a Swagger document.

Generate a swagger document by setting up the URL at the Oracle Application Express instance level. This allows you to define the URL for a Swagger server. Oracle Application Express sends generated swagger document to the server for rendering when the URL is setup at the instance level. Oracle Application Express just displays the JSON generated for Swagger when the URL is not specified.

To generate a Swagger a document:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Select the module for which you want to generate Swagger document.
4. Click **Generate Swagger Doc** on the left pane.
   - If a Swagger UI URL is defined at the instance level, then Oracle Application Express calls the URL and pass the ORDS Swagger Doc URL to it. Opens in a new browser tab and the swagger documentation appears .
   - If the Swagger UI is not defined, Oracle Application Express simply calls ORDS to produce the Swagger documentation JSON file and present that in a new tab.

6.7 Managing Resource Templates

Resource templates are a means of grouping URIs and associating them with a particular RESTful Service Module.

Resource templates can have one or more resource handlers. Only one resource handler per HTTP method is permitted. For example, a resource template can have only one GET method, only one DELETE method and so on.

- Adding a Resource Template (page 6-27)
  Learn how to add a resource template to a ORDS RESTful Service Module.

- Editing a Resource Template (page 6-27)
  Learn how to edit a resource template.
- Deleting a Resource Template (page 6-28)
  Deleting a resource template removes the template along with all resource handlers associated with this template.

6.7.1 Adding a Resource Template

Learn how to add a resource template to a ORDS RESTful Service Module.

To add a resource template to a ORDS RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click the ORDS RESTful Service module you want to ad the resource template to.

   The Module Definition Page appears.
4. Under Resource Templates block on the left panel, click **Create Template**.

   The ORDS Template Definition page appears.
5. Fill in the appropriate fields and select the appropriate options. Mandatory fields are marked with a red asterisk (*). To learn more about a specific option, see the field-level help.
6. Specify the following:
   - **URI Template** - Enter the URI template to identify your Uniform Resource Identifiers. A URI template is simple syntax for describing URIs, for example:

     `example/:id`
   - **Priority** - Specify the template's evaluation order. Higher numbers are evaluated first.
   - **HTTP Entity Tag** - Specify the type of Entity Tag to use to uniquely identify the resource version
7. Click **Create Template**.

6.7.2 Editing a Resource Template

Learn how to edit a resource template.

To edit a resource template:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click ORDS RESTful Service template you want to edit.

   The ORDS Template Definition page appears.

   The ORDS RESTful services page appears.
3. Make the changes you want.

4. Click **Apply Changes**.

6.7.3 Deleting a Resource Template

Deleting a resource template removes the template along with all resource handlers associated with this template.

To delete a resource template:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click ORDS RESTful Service template you want to delete.

   The ORDS Template Definition page appears.
4. Click **Delete**.
5. Click **OK** to confirm.
6. Click **Apply Changes**.

6.8 Managing Resource Handlers

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method.

Multiple resource handlers can be defined for a resource template, however only one resource handler can be defined for a HTTP method.

Parameters to a resource handler can be manually defined to bind HTTP headers to the resource handler, or to cast a URI template parameter to a specific data type. Parameters declared in the URI template are implicitly passed to the resource handler. For example, a resource handler might need to know the value of the HTTP Accept-Language header in order to localize the generated representation.

- Adding a Resource Handler (page 6-28)
  Learn how to add a resource handler to a resource template.
- Editing a Resource Handler (page 6-29)
  Learn how to edit a resource handler.
- Deleting a Resource Handler (page 6-29)
  Learn how to delete a resource handler.

6.8.1 Adding a Resource Handler

Learn how to add a resource handler to a resource template.

To add a resource handler to a resource template:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.
2. Click **ORDS Based RESTful Services**.

The ORDS RESTful services page appears.

3. Click the ORDS RESTful Service template you want to add the resource template to.

   The ORDS Template Definition page appears.

4. Under Resource Handlers on the right pane, click **Create Handler**.

   The ORDS Handler Definition page appears.

5. Fill in the appropriate fields and select the appropriate options. Mandatory fields are marked with a red asterisk (*). To learn more about a specific option, see the field-level help.

6. Under Resource Handler, specify the following:

   - **Method** - Specify the HTTP method to be used for the resource handler. See field-level help.
   - **Source Type** - Identify the strategy type used to generate the resource. See field-level help.
   - **Format** - Select the format to utilize for the result set.
   - **Pagination Size** - Identify the size of the pagination window. For database queries, this is the number of rows to return.

7. Under Source, enter the SQL query or PL/SQL block responsible for handling the selected HTTP method.

8. Click **Create Handler**.

6.8.2 Editing a Resource Handler

Learn how to edit a resource handler.

To edit a resource handler:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.

2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.

3. Click the ORDS RESTful Service Handler you want to edit.

   The ORDS Handler Definition page appears.

4. Make the changes you want.

5. Click **Apply Changes**.

6.8.3 Deleting a Resource Handler

Learn how to delete a resource handler.

To delete a resource handler:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.

   The RESTful Services home page appears.

2. Click **ORDS Based RESTful Services**.

   The ORDS RESTful services page appears.
3. Click the ORDS RESTful Service Handler you want to delete.

   The ORDS Handler Definition page appears.
4. Make the changes you want.
5. Click **Delete**.
6. Click **OK** to confirm.
7. Click **Apply Changes**.

Index

C

create
    generate, *4-22*
Create Lookup Table, *2-12*

D

data
    exporting from database, *4-8*
    loading, *4-2*
    loading into database, *4-5*
    unloading, *4-2*, *4-8*
data dictionary, browsing, *4-52*
Data Workshop, *4-2*
database
    loading data into, *4-2*
    monitoring, *4-53*
    unloading data from, *4-2*
database definition language, generating DDL statements, *4-42*
database link
    browsing, *2-40*
    creating, *2-40*
    testing, *2-41*
Database Monitor
    I/O, *4-56*
    Locks, *4-55*
    Long Operations, *4-58*
    Open Cursors, *4-56*
    Sessions, *4-54*, *4-55*
    SQL, *4-56*
    Top SQL, *4-57*
    Waits, *4-55*
database objects
    purging, *4-47*
    restoring, *4-47*
database security
    Column Privileges report, *4-51*
    Role Privileges report, *4-51*
    System Privileges report, *4-51*
DDL
    about, *4-42*
    generating, *4-42*

E

Explain Plan
    about, *3-10*
    using, *3-10*
Export XML Wizard, *4-10*
exporting data
    summary of available methods, *4-3*

F

function
    browsing, *2-33*
    compiling, *2-34*
    creating, *2-32*
    downloading, *2-35*
    dropping, *2-35*
    editing, *2-34*

I

I/O report, *4-56*
importing data, summary of methods, *4-3*
index
    browsing, *2-18*
    creating, *2-17*
    disabling, *2-18*
    dropping, *2-18*
    rebuilding, *2-18*
    statistics about, *2-18*

L

Load Data Wizard, *4-6*
Load XML Wizard, *4-8*
Locks report, *4-55*

M

materialized view
    browsing, *2-43*
    creating, *2-42*
    dependent objects, *2-43*
    dropping, *2-44*

materialized view *(continued)*
list of grants, *2-43*
report of data, *2-43*
sort by column, *2-43*
viewing object details, *2-43*

O

Object Browser
accessing, *2-2*
creating objects, *2-6*
Hide Objects control, *2-5*
hiding Object Selection pane, *2-5*
managing database links, *2-39*
managing functions, *2-31*
managing indexes, *2-17*
managing materialized views, *2-41*
managing packages, *2-23*
managing procedures, *2-28*
managing sequences, *2-19*
managing tables, *2-6*
managing triggers, *2-35*
managing types, *2-21*
managing views, *2-12*
searching for objects, *2-4*
synonyms, *2-44*
Object Reports
All Objects, *4-52*
All Objects report, *4-52*
Column Privileges report, *4-51*
Data Dictionary, *4-52*
Exception Reports, *4-51*
Invalid Objects, *4-52*
Object Calendar, *4-52*
Object Counts by Type, *4-52*
PL/SQL Source Code report, *4-50*
Role Privileges report, *4-51*
System Privileges report, *4-51*
objects
purging, *4-47*
restoring, *4-47*
Open Cursors report, *4-56*
Oracle Optimizer, Explain Plan, *3-10*
Oracle REST Data Services
registering, *6-7*
ORDS RESTful services
about dashboard, *6-8*
creating, *6-21*
deleting, *6-22*
editing, *6-21*
exporting, *6-25*
importing, *6-25*
managing privileges, *6-22*
managing resource templates, *6-26*

P

package
Auto Complete, *2-25*
compiling, *2-26*
creating, *2-24*
downloading, *2-25*, *2-27*
dropping, *2-27*
editing, *2-25*, *2-26*
Find, *2-25*
grants for, *2-26*
object dependencies, *2-26*
package body, *2-26*
related errors, *2-26*
Replace, *2-25*
viewing, *2-25*
procedure
browsing, *2-29*
compiling, *2-31*
creating, *2-28*
downloading, *2-31*
dropping, *2-31*
editing, *2-30*
editing manually, *2-30*
finding, *2-29*
replacing, *2-29*

Q

Query Builder
about, *4-12*
accessing, *4-12*
creating joins automatically, *4-19*
creating joins manually, *4-19*
filtering objects, *4-13*
hiding objects in Design Pane, *4-16*
Object Selection pane, *4-14*
removing objects in Design Pane, *4-16*
resizing panes, *4-16*
Saved SQL tab, *4-21*
searching for objects, *4-13*
supported column types, *4-15*
Table Actions, *4-19*
View Actions, *4-19*
Quick SQL, *4-22*

R

Recycle Bin
purging, *4-47*
searching, *4-47*
using, *4-47*
viewing objects, *4-47*
reports
Data Dictionary, *4-52*

reports *(continued)*
database objects, *4-52*
exception reports, *4-51*
PL/SQL Source Code, *4-50*
Representational State Transfer (REST), *6-2*
resource handler
adding, *6-28*
deleting, *6-29*
editing, *6-29*
resource template
adding, *6-27*
deleting, *6-28*
editing, *6-27*
managing, *6-26*
RESTful services
about, *6-3*
about oracle.example.hr, *6-11*
accessing, *6-6*
creating, *6-19*, *6-20*
example, *6-11*
managing, *6-18*
managing resource handlers, *6-28*
requirements, *6-4*
Schema ORDS Enabled, *6-20*
Schema ORDS not Enabled, *6-19*
using, *6-1*
results, *5-4*

S

schema
selecting, *1-2*
scripts
creating in editor, *5-4*
deleting from editor, *5-7*
deleting from SQL Scripts page, *5-7*
executing, *5-8*
executing from SQL Scripts page, *5-9*
executing in editor, *5-8*
export, *5-4*
exporting, *5-10*
importing, *5-10*
results, *5-4*
SQL Script editor, *5-5*
uploading to another account, *5-4*
uploading to SQL Scripts, *5-5*
viewing repository quotas, *5-12*
Security reports
Column Privileges report, *4-51*
Object Grants, *4-51*
Role Privileges, *4-51*
System Privileges, *4-51*
sequence
altering, *2-20*
browsing, *2-20*
sequence *(continued)*
creating, *2-19*
dependent objects, *2-20*
dropping, *2-20*
grants associated with, *2-20*
sequence, *2-21*
Session report, *4-55*
session, killing, *4-55*
SQL Commands
home page, *3-3*
running Explain Plan, *3-10*
transaction support, *3-6*
SQL report, *4-56*
SQL Scripts
accessing, *5-2*
creating script in editor, *5-4*
deleting script from editor, *5-7*
deleting script from SQL Scripts home page, *5-7*
executing script from SQL Scripts page, *5-9*
executing script in editor, *5-8*
exporting scripts, *5-10*
importing scripts, *5-10*
script editor, *5-5*
SQL Scripts page, *5-3*
uploading script, *5-5*
viewing quotas, *5-12*
SQL Scripts page
accessing Manage Script Results, *5-4*
Show Quotas, *5-4*
Task list, *5-4*
SQL Scripts page Task list
export scripts, *5-4*
uploading scripts, *5-4*
SQL Workshop
about, *1-1*
icons, *1-1*
overview, *1-1*
selecting a scheme, *1-2*
SQL Workshop home page
icons, *1-1*
selecting a schema, *1-2*
SQL:
shorthand syntax, *4-22*
SQL*Plus command support, *3-7*
SQL*Plus commands, unsupported, *3-7*
Swagger
document, *6-26*
synonym
creating, *2-44*
dropping, *2-45*
viewing, *2-45*

T

tab-delimited file
    loading, 4-6
tables
    add column, 2-11
    browsing, 2-9
    constraints, 2-10
    copy, 2-12
    creating, 2-7
    drop column, 2-11
    dropping, 2-12
    editing, 2-11
    exporting user interface defaults, 4-46
    exports data to spreadsheet, 2-10
    grants on, 2-10
    indexes associated, 2-10
    insert row, 2-9
    modify column, 2-11
    remove all rows, 2-12
    rename, 2-12
    rename column, 2-11
    report of number of rows, 2-10
    sort by column, 2-9
    statistics about, 2-10
    user interface defaults, 2-11
    viewing data, 2-10
text file, loading, 4-6
transaction support, 3-6
trigger
    browsing, 2-36
    compiling, 2-38
    creating, 2-36
    disabling, enabling, 2-39
    downloading, 2-38
    dropping, 2-39
    editing, 2-37
type
    browsing, 2-22
    creating, 2-22
    dropping, 2-23
    grants associated with, 2-23
    list of synonyms, 2-23

U

Unload to Text Wizard, 4-9
Unload to XML Wizard, 4-10
utilities
    about, 4-1
    comparing schemas, 4-1
    generating DDL, 4-1
    loading data, 4-1
    monitoring the database, 4-1
    restoring dropped objects, 4-1
    unloading data, 4-1
    viewing database details, 4-1
    viewing object reports, 4-1
Utilities, Data Dictionary, 4-52

V

view
    browsing, 2-14
    compiling, 2-16
    creating, 2-13
    dropping, 2-16
    editing, 2-15
    grants associated with, 2-15
    objects referenced by, 2-15
    report of data, 2-15
    user interface defaults, 2-15

W

Waits report, 4-55
wizards
    copying and pasting tab-delimited data, 4-6
    exporting contents of table, 4-9
    exporting text file, 4-9
    exporting XML, 4-10
    importing text, 4-6
    importing XML, 4-8

X

XML document
    exporting to, 4-10
    loading, 4-8