

Oracle® Application Express

SQL Workshop Guide

Release 4.2

E35128-03

March 2013

Oracle Application Express SQL Workshop Guide, Release 4.2

E35128-03

Copyright © 2003, 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Drue Swadener

Contributors: Marco Adelfio, Carl Backstrom, Christina Cho, Michael Hichwa, Christopher Jones, Joel Kallman, Sharon Kennedy, Syme Kutz, Terri Jennings, Kris Rice, Marc Sewtz, Scott Spadafore, Scott Spendolini, Jason Straub, and Simon Watt

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, 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 on 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. 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.

Contents

| | |
|--|------|
| Preface | ix |
| Topic Overview | ix |
| Audience | ix |
| Documentation Accessibility | x |
| Related Documents | x |
| Conventions | xi |
| Changes in This Release | xiii |
| Changes in Oracle Application Express Release 4.2 | xiii |
| 1 Managing Database Objects with Object Browser | |
| Accessing Object Browser | 1-1 |
| About Object Browser | 1-2 |
| Searching for and Browsing Database Objects | 1-3 |
| Searching For and Selecting Database Objects | 1-3 |
| Hiding the Object Selection Pane | 1-4 |
| About Creating Database Objects | 1-5 |
| Managing Tables | 1-5 |
| Creating a Table | 1-5 |
| Browsing a Table | 1-7 |
| Editing a Table | 1-9 |
| Dropping a Table | 1-10 |
| Managing Views | 1-10 |
| Creating a View | 1-11 |
| Browsing a View | 1-11 |
| Editing a View | 1-12 |
| Compiling a View | 1-13 |
| Dropping a View | 1-13 |
| Managing Indexes | 1-13 |
| Creating an Index | 1-13 |
| Browsing an Index | 1-14 |
| Dropping an Index | 1-14 |
| Managing Sequences | 1-15 |
| Creating a Sequence | 1-15 |
| Browsing a Sequence | 1-16 |

| | |
|---|-------------|
| Dropping a Sequence..... | 1-16 |
| Managing Types | 1-16 |
| Creating a Type | 1-17 |
| Browsing a Type..... | 1-17 |
| Dropping a Type | 1-17 |
| Managing Packages..... | 1-18 |
| Creating a Package..... | 1-18 |
| Viewing a Package | 1-19 |
| Editing a Package | 1-20 |
| Compiling a Package | 1-20 |
| Downloading a Package..... | 1-21 |
| Dropping a Package..... | 1-21 |
| Managing Procedures | 1-21 |
| Creating a Procedure | 1-22 |
| Browsing a Procedure | 1-22 |
| Editing a Procedure | 1-23 |
| Compiling a Procedure | 1-23 |
| Downloading a Procedure..... | 1-24 |
| Dropping a Procedure..... | 1-24 |
| Managing Functions | 1-24 |
| Creating a Function..... | 1-24 |
| Browsing a Function..... | 1-25 |
| Editing a Function..... | 1-26 |
| Compiling a Function..... | 1-26 |
| Downloading a Function..... | 1-26 |
| Dropping a Function..... | 1-26 |
| Managing Triggers | 1-27 |
| Creating Triggers | 1-27 |
| Browsing a Trigger..... | 1-27 |
| Editing a Trigger | 1-28 |
| Compiling a Trigger..... | 1-29 |
| Downloading a Trigger | 1-29 |
| Dropping a Trigger | 1-29 |
| Managing Database Links | 1-30 |
| Creating a Database Link..... | 1-30 |
| Browsing a Database Link | 1-30 |
| Dropping a Database Link..... | 1-31 |
| Managing Materialized Views..... | 1-31 |
| Creating a Materialized View..... | 1-31 |
| Browsing a Materialized View | 1-32 |
| Dropping a Materialized View..... | 1-33 |
| Managing Synonyms | 1-33 |
| Creating Synonyms..... | 1-34 |
| Viewing a Synonym..... | 1-34 |
| Dropping a Synonym | 1-34 |

2 Using SQL Commands

| | |
|--|------|
| What is SQL Commands? | 2-1 |
| Accessing SQL Commands | 2-2 |
| About the SQL Commands Home Page | 2-2 |
| Using the Command Editor | 2-5 |
| Running a SQL Command..... | 2-5 |
| About Transactions in SQL Commands | 2-5 |
| About Unsupported SQL*Plus Commands | 2-6 |
| About Command Termination..... | 2-6 |
| Using Bind Variables | 2-6 |
| Using the Find Tables Icon | 2-7 |
| Saving an SQL Command | 2-8 |
| Copying a Command | 2-9 |
| Viewing Results | 2-9 |
| Accessing the Results Pane | 2-9 |
| About the Results Pane | 2-9 |
| Using Explain Plan | 2-10 |
| Viewing an Explain Plan..... | 2-10 |
| About Explain Plan Pane | 2-11 |
| Using Saved Commands | 2-11 |
| Accessing Saved Commands..... | 2-11 |
| About the Saved SQL Pane..... | 2-12 |
| Using SQL Command History | 2-13 |
| Accessing a Command from Command History | 2-13 |
| About the History Pane..... | 2-13 |

3 Using SQL Scripts

| | |
|---|------|
| What is SQL Scripts? | 3-1 |
| Accessing SQL Scripts | 3-1 |
| About the SQL Scripts Page..... | 3-2 |
| Creating a SQL Script | 3-4 |
| Creating a SQL Script in the Script Editor..... | 3-4 |
| Uploading a SQL Script..... | 3-5 |
| Using the Script Editor | 3-5 |
| Editing an Existing Script..... | 3-5 |
| Searching and Replacing Text or Regular Expressions | 3-6 |
| Summary of Script Editor Controls | 3-7 |
| Deleting a SQL Script | 3-8 |
| Deleting Scripts from the SQL Scripts Page | 3-8 |
| Deleting a Script in the Script Editor..... | 3-8 |
| Copying a SQL Script | 3-8 |
| Executing a SQL Script | 3-9 |
| Executing a SQL Script in the Script Editor..... | 3-9 |
| Executing a SQL Script from the SQL Scripts Page..... | 3-9 |
| Viewing SQL Script Results | 3-10 |
| Viewing Results from the SQL Scripts Page | 3-10 |

| | |
|--|-------------|
| About the Results Page | 3-11 |
| Exporting and Importing SQL Scripts | 3-12 |
| Copying Scripts to an Export Script | 3-12 |
| Importing Scripts from an Export Script | 3-14 |
| Viewing Script and Result Quotas | 3-15 |

4 Using Oracle Application Express Utilities

| | |
|--|-------------|
| Building SQL Queries with Query Builder | 4-1 |
| About Query Builder | 4-2 |
| Using the Object Selection Pane | 4-4 |
| Selecting Objects | 4-5 |
| Specifying Query Conditions | 4-7 |
| Creating Relationships Between Objects | 4-9 |
| Working with Saved Queries | 4-11 |
| Viewing Generated SQL | 4-13 |
| Viewing Query Results | 4-13 |
| Using the Data Workshop to Manage Data | 4-14 |
| About Importing, Exporting, Loading, and Unloading Data | 4-14 |
| Accessing the Data Load/Unload Page | 4-16 |
| Loading Data | 4-17 |
| Unloading Data | 4-19 |
| Using Repository | 4-21 |
| Generating DDL | 4-21 |
| Managing Methods on Tables | 4-22 |
| Managing User Interface Defaults | 4-22 |
| Creating User Interface Defaults for a Table | 4-23 |
| Modifying Table User Interface Defaults | 4-23 |
| Creating User Interface Attributes | 4-24 |
| Modifying User Interface Attributes | 4-24 |
| About Exporting and Importing User Interface Defaults | 4-25 |
| Viewing Database Details | 4-25 |
| Comparing Schemas | 4-26 |
| Viewing Object Reports | 4-26 |
| Table Reports | 4-27 |
| Security Reports | 4-27 |
| PL/SQL Reports | 4-28 |
| Exception Reports | 4-29 |
| All Object Reports | 4-30 |
| Monitoring the Database | 4-30 |
| Sessions | 4-31 |
| About System Statistics | 4-33 |
| About Top SQL | 4-33 |
| About Long Operations | 4-34 |
| Using the Recycle Bin to View and Restore Dropped Objects | 4-34 |
| Managing Objects in the Recycle Bin | 4-35 |
| Emptying the Recycle Bin Without Viewing the Objects | 4-35 |

5 Using RESTful Services

| | |
|---|------|
| What Is REST? | 5-1 |
| About RESTful Services | 5-2 |
| RESTful Service Requirements | 5-2 |
| Accessing RESTful Services | 5-3 |
| About the Example RESTful Service Module (oracle.example.hr) | 5-3 |
| Returning the Result Set in CSV Format (empinfo/) | 5-5 |
| Retrieving Data Based on a Parameter (employees/{id}) | 5-7 |
| Returning the Result Set in JSON Format with Pagination Set (employees/) | 5-9 |
| Returning the Result Set as a Feed (employeesfeed/) | 5-12 |
| Returning the Result Set Based on a PL/SQL Block (empsecformat/{empname}) | 5-13 |
| Managing RESTful Service Modules | 5-16 |
| Creating a RESTful Service Module | 5-16 |
| Editing a RESTful Service Module | 5-17 |
| Deleting a RESTful Service Module | 5-19 |
| Managing RESTful Service Privileges | 5-19 |
| Exporting a RESTful Service Module | 5-21 |
| Importing a RESTful Service Module | 5-21 |
| Managing Resource Templates | 5-22 |
| Adding a Resource Template | 5-22 |
| Editing a Resource Template | 5-23 |
| Deleting a Resource Template | 5-23 |
| Managing Resource Handlers | 5-23 |
| Adding a Resource Handler | 5-24 |
| Editing a Resource Handler | 5-25 |
| Deleting a Resource Handler | 5-26 |

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.

Topics:

- [Topic Overview](#)
- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Topic Overview

This document contains the following chapters:

| Title | Description |
|---|--|
| Changes in This Release | Describes changes in this document for Oracle Application Express, release 4.2. |
| Managing Database Objects with Object Browser | Explains how Object Browser enables developers to browse, create, and edit objects in a database. |
| Using SQL Commands | Provides information on how to use SQL Commands to create, edit, view, run, and delete SQL commands. |
| Using SQL Scripts | Describes how to use SQL Scripts to create, edit, view, run and delete script files. |
| Using Oracle Application Express Utilities | Describes how to use Oracle Application Express utilities to load and unload data from an Oracle database, generate DDL, view object reports and restore dropped database objects. |
| Using RESTful Services | Describes how to use Oracle Application Express RESTful Services to enable the declarative specification of RESTful services used to access the database. |

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.

See Also: *Oracle Database 2 Day + Application Express Developer's Guide*

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 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.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Related Documents

For more information, see these Oracle resources:

- *Oracle Application Express Release Notes*
- *Oracle Application Express Installation Guide*
- *Oracle Database 2 Day + Application Express Developer's Guide*
- *Oracle Application Express Application Builder User's Guide*
- *Oracle Application Express API Reference*
- *Oracle Application Express Administration Guide*
- *Oracle Application Express Migration Guide*
- *Oracle Application Express End User's Guide*
- *Oracle Database Concepts*
- *Oracle Database Advanced Application Developer's Guide*
- *Oracle Database Administrator's Guide*

- *Oracle Database SQL Language Reference*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database PL/SQL Language Reference*
- *Oracle Application Express Listener Installation and Developer Guide*

For information about Oracle error messages, see *Oracle Database Error Messages*. Oracle error message documentation is available only in HTML. If you have access to the Oracle Database Documentation Library, you can browse the error messages by range. Once you find the specific range, use your browser's "find in page" feature to locate the specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle online documentation.

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

For additional documentation available on Oracle's Technology Network, visit the Oracle Application Express Web site located at

<http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html>

For additional application examples, go to the Learning Library. Search for free online training content, including Oracle by Example (OBE), demos, and tutorials. To access the Oracle Learning Library, go to:

<http://www.oracle.com/technetwork/tutorials/index.html>

Printed documentation is available for sale in the Oracle Store at

<http://shop.oracle.com/>

If you have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://www.oracle.com/technology/documentation/>

Conventions

The following text conventions are used in this document:

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

Changes in This Release

This preface contains:

- [Changes in Oracle Application Express Release 4.2](#)

Changes in Oracle Application Express Release 4.2

The following are changes in *Oracle Application Express SQL Workshop Guide* for Oracle Application Express release 4.2.

New Features

The following features are new in this release:

- RESTful Web Services

In SQL Workshop you can now define RESTful Web Service calls from the database. These definitions combined with the Oracle Application Express Listener 2.0 enables the database to emit RESTful Web Services rather than just being able to consume Web Services.

See "[Using RESTful Services](#)" on page 5-1.

Other Changes

The following are additional changes in the release:

- All content has been updated to reflect new functionality.
- Screen captures and graphics have been added and updated to reflect Oracle Application Express release 4.2 user interface enhancements.

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. See *Oracle Application Express API Reference* for details of the views defined.

Topics:

- [Accessing Object Browser](#)
- [About Object Browser](#)
- [Searching for and Browsing Database Objects](#)
- [About Creating Database Objects](#)
- [Managing Tables](#)
- [Managing Views](#)
- [Managing Indexes](#)
- [Managing Sequences](#)
- [Managing Types](#)
- [Managing Packages](#)
- [Managing Procedures](#)
- [Managing Functions](#)
- [Managing Triggers](#)
- [Managing Database Links](#)
- [Managing Materialized Views](#)
- [Managing Synonyms](#)

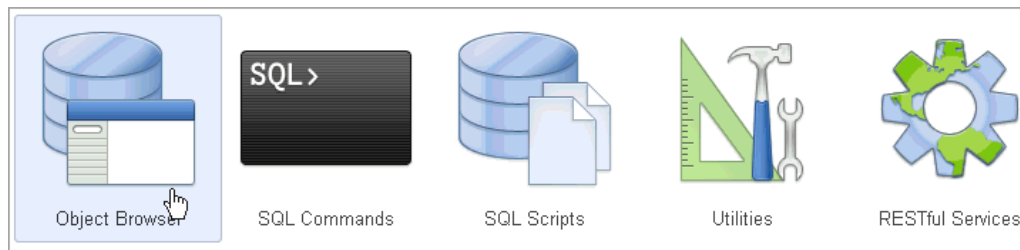
See Also: *Oracle Database SQL Language Reference*

Accessing Object Browser

To access Object Browser:

1. Log in to the Workspace home page.
2. Click **SQL Workshop**.

3. Click **Object Browser** to drill down to the Object Browser home page.



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 "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*.

About Object Browser

The Object Browser page is divided into two sections:

- **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.

| Column Name | Data Type | Nullable | Default | Primary Key |
|-----------------|----------------|----------|---------|-------------|
| CONSTRAINT_NAME | VARCHAR2(30) | No | - | 1 |
| MESSAGE | VARCHAR2(4000) | No | - | - |

See Also: ["Searching for and Browsing Database Objects"](#) on page 1-3 and ["About Creating Database Objects"](#) on page 1-5

Selecting a Schema

A schema is a logical container for database objects. To access objects in another schema, make a selection from the Schema list in the SQL Workshop home page. Only schemas assigned to your workspace are listed.

Switching to Another SQL Workshop Component

You can navigate to another SQL Workshop component by selecting the corresponding icon located on the upper right side of the page.



Available icons include:

- **Object Browser.** See ["Managing Database Objects with Object Browser"](#) on page 1-1.
- **SQL Commands.** See ["Using SQL Commands"](#) on page 2-1.
- **SQL Scripts.** See ["Using SQL Scripts"](#) on page 3-1.
- **Query Builder.** See ["Building SQL Queries with Query Builder"](#) on page 4-1.
- **Utilities.** See ["Using Oracle Application Express Utilities"](#) on page 4-1.
- **RESTful Services.** See ["Accessing RESTful Services"](#) on page 5-3.
- **Help.**

Searching for and Browsing Database Objects

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.

Topics:

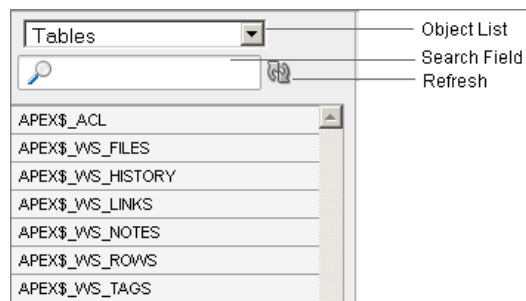
- [Searching For and Selecting Database Objects](#)
- [Hiding the Object Selection Pane](#)

Searching For and Selecting Database Objects

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.



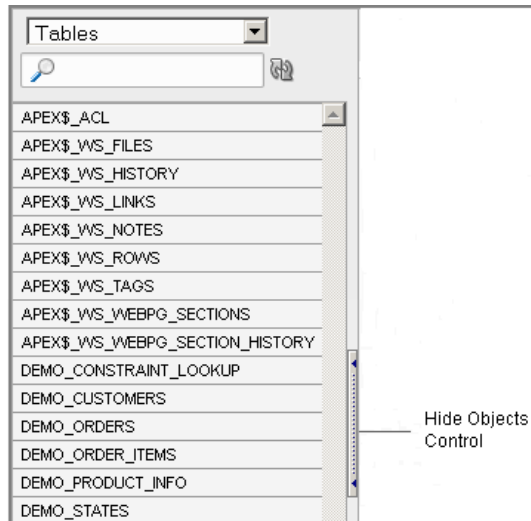
3. To search for an object name, enter a case insensitive search term in the Search field.

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.

Hiding the Object Selection Pane

You can hide the Object Selection pane by selecting the **Hide Objects** control. This control 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.



About 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 **Create**, located 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.

Managing Tables

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.

Topics:

- [Creating a Table](#)
- [Browsing a Table](#)

- [Editing a Table](#)
- [Dropping a Table](#)

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.

Creating a Table

To create a table:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
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 details for each column. For each column:
 - a. Enter the column name.
 - b. Select the column type.
 - c. Enter the following additional information as appropriate:
 - 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.
 - 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.
 - d. To specify a column must always have a value, select the check box in the **Not Null** column.
To change the order of previously entered columns, click the **Up** and **Down** arrows in the Move column. To add additional columns, click **Add Column**.

- e. If you are running Oracle Database 12c, 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.
- f. 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**:
 - **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).

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. To add a foreign key:
 - a. Name - Enter a name of the foreign key constraint that you are defining.
 - b. 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).
 - c. 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).
 - d. 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.
 - 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.

- 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**.

See Also: "Using the Table Finder" in *Oracle Application Express Application Builder User's Guide* and "Overview of Tables" in *Oracle Database Concepts* for information about tables.

Browsing 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the table. [Table 1-1](#) describes all available views.

Table 1–1 Available Views for Tables

| View | Description |
|-------------|---|
| Table | <p>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.</p> <p>To export the data as a comma-delimited file (.csv) file, click the Download link.</p> <p>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 Instance Settings" in <i>Oracle Application Express Administration Guide</i>.</p> <p>See Also: "Editing a Table" on page 1-9</p> |
| Data | <p>Displays a report of the data in the current table. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ 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. |
| Indexes | <p>Displays indexes associated with this table. Actions you can perform include Create and Drop.</p> <p>See Also: "Managing Indexes" on page 1-13</p> |
| Model | <p>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.</p> <p>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.</p> |
| Constraints | <p>Displays a list of constraints for the current table. Actions you can perform include Create, Drop, Enable, and Disable.</p> |
| Grants | <p>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.</p> |
| Statistics | <p>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.</p> |

Table 1–1 (Cont.) Available Views for Tables

| View | Description |
|--------------|---|
| UI Defaults | <p>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.</p> <p>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.</p> <p>See Also: "Managing User Interface Defaults" on page 4-22</p> |
| Triggers | <p>Displays a list of triggers associated with the current table. Actions you can perform include Create, Drop, Enable, and Disable.</p> <p>To view trigger details, click the trigger name.</p> <p>See Also: "Managing Triggers" on page 1-27</p> |
| 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. |

Editing 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 1–2](#).

Table 1–2 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. |
| Rename Column | Renames the selected column. |
| Drop Column | Drops the selected column. |
| Rename | Renames the selected table. |
| Copy | Copies the selected table. |
| Drop | Drops the selected table. |
| | See Also: "Using the Recycle Bin to View and Restore Dropped Objects" on page 4-34 |

Table 1–2 (Cont.) Edit Table Buttons

| Button | Description |
|---------------------|--|
| 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. |

Dropping a Table

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**.

See Also: ["Using the Recycle Bin to View and Restore Dropped Objects"](#) on page 4-34

Managing Views

A view is a logical representation of another table or combination of tables. 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
```

Topics:

- [Creating a View](#)
- [Browsing a View](#)
- [Editing a View](#)
- [Compiling a View](#)
- [Dropping a View](#)

See Also: *Oracle Database Administrator's Guide*

Creating a View

To create a view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **View**.
4. Define the view:
 - **View Name** - Enter a name for the View.
 - **Query** - Specify a query to define the view.
To access Query Builder or SQL Command Processor, click the appropriate link at the bottom of the page. The selected tool appears 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**.

See Also: ["Building SQL Queries with Query Builder"](#) on page 4-1 and ["Using SQL Commands"](#) on page 2-1

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports. [Table 1-3](#) describes all available views.

Table 1–3 Available Views for Views

| View | Description |
|--------------|--|
| View | <p>(Default) Displays the columns in the current view. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ Compile ■ Drop <p>See Also: "Editing a View" on page 1-12, "Compiling a View" on page 1-13, and "Dropping a View" on page 1-13</p> |
| Code | <p>Displays the code editor.</p> <p>See Also: "Editing a View" on page 1-12.</p> |
| Data | <p>Displays a report of the data in the columns in the view. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ 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 | <p>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.</p> |
| UI Defaults | <p>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.</p> <p>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.</p> <p>See Also: "Managing User Interface Defaults" on page 4-22</p> |
| Dependencies | <p>Displays a report showing objects referenced by this view, objects this view references, and synonyms for this view.</p> |
| SQL | <p>Displays the SQL necessary to re-create this view.</p> |

Editing a View

When you edit a view you can edit the code manually, perform a search and replace, and compile the view. Additionally, you can save the view as a file or drop it.

Editing a View Manually

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. See ["Compiling a View"](#) on page 1-13.

Using Find and Replace

Click **Find & Replace** to perform a basic search and replace.

Downloading a View

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

Compiling a View

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

Click **Save & Compile** to re-create the current view.

Dropping a View

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**.

Managing Indexes

An index is an optional structure associated with tables and clusters. 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 type, uniqueness, compression, prefix length, tablespace name, status, last analyzed and a listing of the indexed columns.

Topics:

- [Creating an Index](#)
- [Browsing an Index](#)
- [Dropping an Index](#)

Creating an Index

To create an index:

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

Browsing 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, type, table owner, table type and a listing of indexed columns.

Summary of Available Views

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

[Table 1–4](#) describes all available views.

Table 1–4 Available Views 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: <ul style="list-style-type: none"> ■ 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. |

Dropping an Index

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**.

Managing Sequences

A sequence generates a serial list of unique numbers for numeric columns of a database table. Database sequences are generally used to populate table primary keys.

Topics:

- [Creating a Sequence](#)
- [Browsing a Sequence](#)
- [Dropping a Sequence](#)

Creating a Sequence

To create a sequence:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Sequence**.
4. Specify the following and click **Next**:
 - **Sequence Name** - Enter the name of the sequence.
 - **Preserve Case** - To have the final sequence name match the case entered in the Sequence Name field, click **Preserve Case**.
 - **Start With** - Enter the number of the first sequence. The first reference to `sequence_name.nextval` returns this number.
 - **Minimum Value** - Enter the minimum value this sequence can return.
 - **Maximum Value** - Enter the maximum value this sequence can return.
 - **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.
 - **Cycle** - Select this option to restart the sequence number to the minimum value when the maximum value is reached. This is not recommended if using the sequence for primary key creation.
 - **Number to Cache** - For faster access, specify how many sequence values are stored in memory.
 - **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).

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

5. Click **Create Sequence**.

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the sequence. [Table 1–5](#) describes all available views.

Table 1–5 Available Views for Sequences

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

Dropping a Sequence

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**.
5. To confirm, click **Finish**.

Managing Types

A type is a user-specified object or collection definition. 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.

Topics:

- [Creating a Type](#)
- [Browsing a Type](#)
- [Dropping a Type](#)

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

Creating a Type

To create a collection type:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Type**.
4. Specify a name and click **Next**.
5. Select a type, data type, limit, and click **Next**.
A confirmation page appears, which displays the SQL used to create the type.
6. Click **Create Type**.

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports. [Table 1–6](#) describes all available views.

Table 1–6 Available Views 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. Actions you can perform include Grant and Revoke . |
| SQL | Displays the SQL necessary to re-create this type. |

Dropping a Type

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**.

Managing Packages

A package is a database object that groups logically related PL/SQL types, items, functions and procedures. Packages usually have two parts, a specification and a body. The **specification** is the interface to your application. The **body** implements the specification.

Topics:

- [Creating a Package](#)
- [Viewing a Package](#)
- [Editing a Package](#)
- [Compiling a Package](#)
- [Downloading a Package](#)
- [Dropping a Package](#)

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

Creating a Package

To create a package:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Package**.
4. Select the type of package you want to create, then click **Next**:
 - Specification
 - Body
 - Package with methods on database tables
5. If you select **Specification**:
 - a. Enter a name and click **Next**.
The wizard creates a dummy package specification and displays it for editing.
 - b. Edit the specification and click **Finish**.
6. If you select **Body**:
 - a. 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 **Finish**.
7. If you select **Package with methods on database tables**:
 - a. Enter a name and click **Next**.
 - 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 **Finish**.

Viewing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the package. [Table 1–7](#) describes all available views.

Table 1–7 Available Views for Packages

| View | Description |
|---------------|---|
| Specification | <p>Displays the package specification. This defines the interface to your application. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ Edit the code manually ■ Save & Compile ■ Find & Replace ■ Undo ■ Redo ■ Download Source ■ Drop |

Table 1–7 (Cont.) Available Views for Packages

| View | Description |
|--------------|---|
| Body | <p>Displays the package body, if one exists, for the selected package. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ Edit the code manually ■ Save & Compile ■ Find & Replace ■ 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. |

Editing a Package

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

Editing a Package Manually

To edit a package manually:

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. Click **Find & Replace** to perform a basic search and replace.

Compiling a Package

If you edit and make changes to a package, 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.

Downloading a Package

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.

Dropping a Package

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**.

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**.

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.

Topics:

- [Creating a Procedure](#)
- [Browsing a Procedure](#)
- [Editing a Procedure](#)
- [Compiling a Procedure](#)
- [Downloading a Procedure](#)
- [Dropping a Procedure](#)

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

Creating a Procedure

To create a procedure:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Procedure**.
4. Enter a procedure name and click **Next**.
5. Define the arguments by specifying the following information (optional):
 - Argument Name
 - In/Out (the parameter mode)
 - Argument Type (datatype)
 - Default (value)

To add additional arguments, click **Add Argument**.

6. Click **Next**.
7. Enter PL/SQL block you want to use as the procedure body and click **Next**.
A confirmation page appears, which displays the SQL used to create the procedure.
8. Click **Create Procedure**.

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the procedure. [Table 1–8](#) describes all available views.

Table 1–8 Available Views for Procedures

| View | Description |
|--------------|--|
| Code | <p>(Default) Displays the source code for the procedure. You can edit the code in this view. Actions you can perform in this view include:</p> <ul style="list-style-type: none"> ■ Edit the code manually ■ Save & Compile ■ Find & Replace ■ Undo ■ Redo ■ Download Source ■ Drop <p>See Also: "Editing a Procedure" on page 1-23, "Compiling a Procedure" on page 1-23, "Downloading a Procedure" on page 1-24, and "Dropping a Procedure" on page 1-24</p> |
| 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. |

Editing a Procedure

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

Editing a Procedure Manually

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. Click **Find & Replace** to perform a basic search and replace.

Compiling a Procedure

If you edit and make changes to a procedure, 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.

Downloading a Procedure

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.

Dropping a Procedure

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**.

Managing Functions

A function is a subprogram that can take parameters and return a single value.

Topics:

- [Creating a Function](#)
- [Browsing a Function](#)
- [Editing a Function](#)
- [Compiling a Function](#)
- [Downloading a Function](#)
- [Dropping a Function](#)

See Also: *Oracle Database SQL Language Reference* for information about PL/SQL functions and *Oracle Database PL/SQL Language Reference*

Creating a Function

To create a function:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Function**.

4. Enter a function name, specify the return datatype, and click **Next**.
5. Define the arguments and click **Next** (optional):
 - Argument Name
 - Argument Type (datatype)
 - Default (value)
 To add additional arguments, click **Add Argument**.
6. Enter PL/SQL block you want to use as the function body and click **Next**.
A confirmation page appears, which displays the SQL used to create the function.
7. Click **Create Function**.

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the function.
[Table 1–9](#) describes all available views.

Table 1–9 Available Views 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: <ul style="list-style-type: none"> ■ Edit the code manually ■ Save & Compile ■ Find & Replace ■ Undo ■ Redo ■ Download Source ■ Drop See Also: "Editing a Function" on page 1-26, "Compiling a Function" on page 1-26, "Downloading a Function" on page 1-26, and "Dropping a Function" on page 1-26 |
| 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. |

Editing a Function

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

Editing a Function Manually

To edit a function manually:

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. Click **Find & Replace** to perform a basic search and replace.

Compiling a Function

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.

Downloading a Function

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.

Dropping a Function

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**.

Managing Triggers

A database trigger is a stored subprogram associated with a database table, view, or event. 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.

Topics:

- [Creating Triggers](#)
- [Browsing a Trigger](#)
- [Editing a Trigger](#)
- [Compiling a Trigger](#)
- [Downloading a Trigger](#)
- [Dropping a Trigger](#)

See Also:

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

Creating Triggers

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 **Create**.
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**.

Browsing a 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**.

- From the Object Selection pane, select a trigger.

The Object Details view appears.

Summary of Available Views

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

[Table 1–10](#) describes all available views.

Table 1–10 Available Views for Triggers

| View | Description |
|----------------|---|
| Object Details | (Default) Lists of the details about the current trigger. Actions you can perform include: <ul style="list-style-type: none"> ■ 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: <ul style="list-style-type: none"> ■ Edit the code manually ■ Save & Compile ■ Find & Replace ■ Undo ■ Redo ■ Download Source ■ Drop <p>See Also: "Editing a Trigger" on page 1-28, "Compiling a Trigger" on page 1-29, "Downloading a Trigger" on page 1-29, and "Dropping a Trigger" on page 1-29</p> |
| Errors | Displays errors related to the current trigger. |
| SQL | Displays the SQL necessary to re-create the trigger. |

Editing a Trigger

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

Editing a Trigger Manually

To edit a trigger manually:

- On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
- From the Object list, select **Triggers**.
- From the Object Selection pane, select a trigger.
The Code view appears. You can edit the code in this view.
- Click **Find & Replace** to perform a basic search and replace.

Compiling a Trigger

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.

Downloading a Trigger

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.

Dropping a Trigger

To save 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**.
5. To confirm, click **Finish**.

Managing Database Links

A database link is a schema object in one database that enables you to access objects in another database. 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.

Topics:

- [Creating a Database Link](#)
- [Browsing a Database Link](#)
- [Dropping a Database Link](#)

Creating a Database Link

To create a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Database Link**.
4. Specify the following information and click **Next**.
 - Database Link Name
 - Connect To Schema
 - Password
 - Remote Hostname or IP
 - Remote Host Port
 - SID or Service NameA confirmation page appears.
5. Click **Create**.

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the database link. [Table 1–11](#) describes all available views.

Table 1–11 Available Views for Database Links

| View | Description |
|----------------|---|
| Object Details | (Default) Displays details about the database link. Actions you can perform include: <ul style="list-style-type: none"> ■ 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. |

Dropping a Database Link

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**.

Managing Materialized Views

A materialized view provides indirect access to table data by storing the results of a query in a separate schema object. 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.

Topics:

- [Creating a Materialized View](#)
- [Browsing a Materialized View](#)
- [Dropping a Materialized View](#)

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

Creating a Materialized View

To create a materialized view:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Materialized View**.
4. Define the materialized view:
 - a. **Materialized View Name** - Enter a name.
 - b. **Query** - Specify a query to define the view.
To access Query Builder or SQL Command Processor, click the appropriate link at the bottom of the page. The selected tool appears in a pop-up window. Once you generate 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**.

See Also: ["Building SQL Queries with Query Builder"](#) on page 4-1
and ["Using SQL Commands"](#) on page 2-1

Browsing a 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.

Summary of Available Views

Click the tabs at the top of the page to view different reports about the materialized view. [Table 1–12](#) describes all available views.

Table 1–12 Available Views for Materialized View

| View | Description |
|-------------------|--|
| Materialized View | (Default) Displays details about the columns in the materialized view, including: <ul style="list-style-type: none">■ Column Name■ Data type■ Nullable flag■ Default value■ Primary key Click Drop to delete the current materialized view. |

Table 1–12 (Cont.) Available Views for Materialized View

| View | Description |
|--------------|---|
| Data | <p>Displays a report of the data in the columns. Actions you can perform include:</p> <ul style="list-style-type: none"> ■ 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. Actions you can perform in this view include Grant and Revoke . |
| Dependencies | Displays a list of objects that use (or depend) upon this materialized view. |
| SQL | Displays the SQL necessary to re-create this materialized view. |

Dropping a Materialized View

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**.

Managing Synonyms

A synonym is an alias for a schema object. 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.

Topics:

- [Creating Synonyms](#)
- [Viewing a Synonym](#)
- [Dropping a Synonym](#)

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

Creating Synonyms

To create a synonym:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.
Object Browser appears.
2. Click **Create**.
3. From the list of object types, select **Synonym**.
4. Define the synonym:
 - 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**.

Viewing a Synonym

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

Dropping a Synonym

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**.

Using SQL Commands

This section provides information on how to use SQL Commands to create, edit, view, run, and delete SQL commands.

Topics:

- [What is SQL Commands?](#)
- [Accessing SQL Commands](#)
- [About the SQL Commands Home Page](#)
- [Using the Command Editor](#)
- [Saving an SQL Command](#)
- [Copying a Command](#)
- [Viewing Results](#)
- [Using Explain Plan](#)
- [Using Saved Commands](#)
- [Using SQL Command History](#)

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

What is 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.

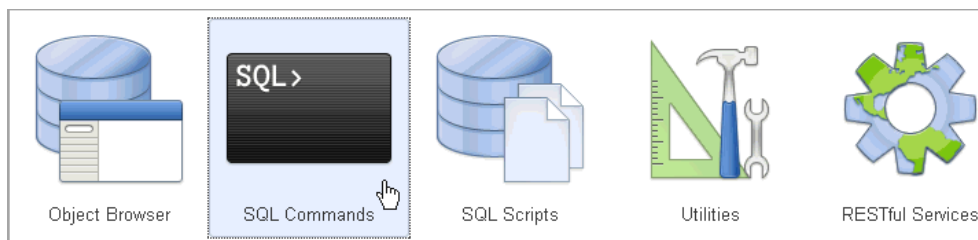
- There is no interaction between SQL Commands and SQL Scripts.
- You can cut and paste an SQL command from SQL Commands to run in the SQL Script Editor.

See Also: ["Using SQL Scripts"](#) on page 3-1

Accessing SQL Commands

To access SQL Commands:

1. Log in to the Workspace home page.
The Workspace home page appears.
2. Click **SQL Workshop**.
3. Click **SQL Commands**.

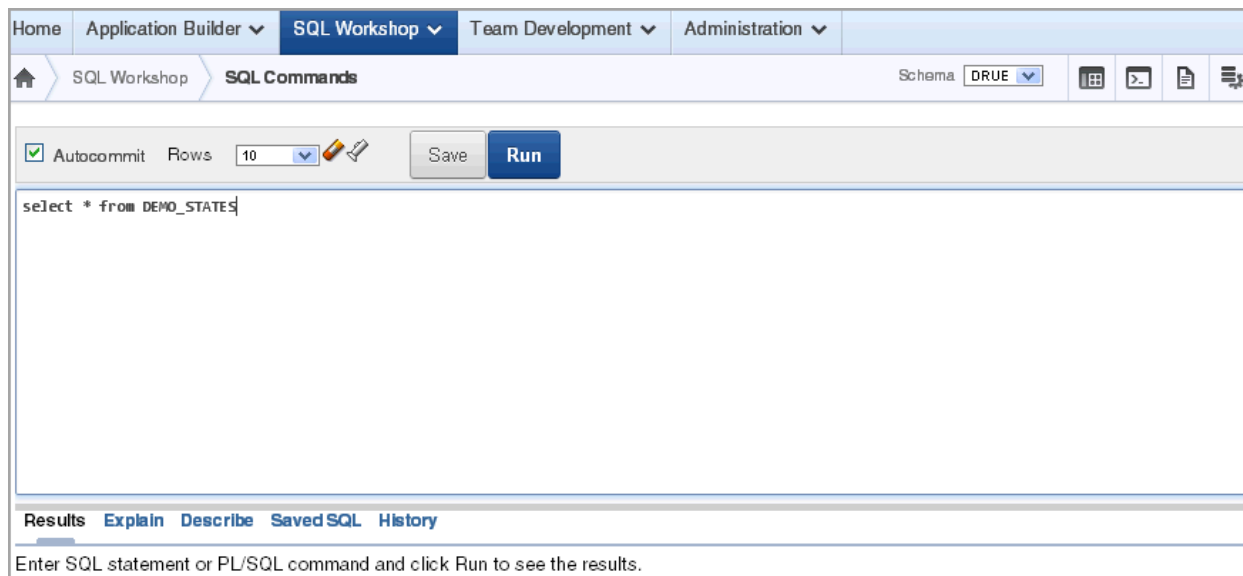


The SQL Commands home page appears.

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 "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*.

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. See ["About Transactions in SQL Commands"](#) on page 2-5.
- **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 icon.** The Clear Command icon resembles a pencil with an eraser. Use this icon to clear the text in the command editor.
- **Find Tables icon.** The Find Tables icon resembles a flashlight. Click this icon to view tables within the currently selected schema. See ["Using the Find Tables Icon"](#) on page 2-7.
- **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. See ["Saving an SQL Command"](#) on page 2-8.
- **Run.** Click the **Run** button to run the command in the command editor, or the currently highlighted command in the command editor. See ["Running a SQL Command"](#) on page 2-5.

See Also: ["Using the Command Editor"](#) on page 2-5

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. See ["Viewing Results"](#) on page 2-9.
- **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. See ["Using Explain Plan"](#) on page 2-10.

- **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. See ["Using Saved Commands"](#) on page 2-11
- **History.** Click the **History** tab to list your recently executed commands. Your last 200 executed commands are saved. See ["Using SQL Command History"](#) on page 2-13.

Selecting a Schema

A schema is a logical container for database objects. To access objects in another schema, make a selection from the Schema list in the SQL Workshop home page. Only schemas assigned to your workspace are listed.

Switching to Another SQL Workshop Component

You can navigate to another SQL Workshop component by making a selection from the Component list located on the upper right side of the page.



Available icons include:

- **Object Browser.** See ["Managing Database Objects with Object Browser"](#) on page 1-1.
- **SQL Commands.** See ["Using SQL Commands"](#) on page 2-1.
- **SQL Scripts.** See ["Using SQL Scripts"](#) on page 3-1.
- **Query Builder.** See ["Building SQL Queries with Query Builder"](#) on page 4-1.
- **Utilities.** See ["Using Oracle Application Express Utilities"](#) on page 4-1.
- **RESTful Services.** See ["Accessing RESTful Services"](#) on page 5-3.
- **Help.**

Using the Command Editor

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

Topics:

- [Running a SQL Command](#)
- [About Transactions in SQL Commands](#)
- [About Unsupported SQL*Plus Commands](#)
- [About Command Termination](#)

- [Using Bind Variables](#)
- [Using the Find Tables Icon](#)

See Also: ["About the SQL Commands Home Page"](#) on page 2-2

Running a SQL Command

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.

See Also: ["About the Results Pane"](#) on page 2-9, ["About Transactions in SQL Commands"](#) on page 2-5, ["About Unsupported SQL*Plus Commands"](#) on page 2-6, ["About Command Termination"](#) on page 2-6, and ["Using Bind Variables"](#) on page 2-6

About Transactions in 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.

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.

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. See "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*.

- The **CSV Export** option is not available.

See Also: ["Running a SQL Command"](#) on page 2-5, ["About Transactions in SQL Commands"](#) on page 2-5, ["About Unsupported SQL*Plus Commands"](#) on page 2-6, and ["Using Bind Variables"](#) on page 2-6

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.

See Also: ["Running a SQL Command"](#) on page 2-5, ["About Transactions in SQL Commands"](#) on page 2-5, ["About Unsupported SQL*Plus Commands"](#) on page 2-6, and ["Using Bind Variables"](#) on page 2-6

About Command Termination

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.

See Also: ["Running a SQL Command"](#) on page 2-5, ["About Transactions in SQL Commands"](#) on page 2-5, ["About Unsupported SQL*Plus Commands"](#) on page 2-6, and ["Using Bind Variables"](#) on page 2-6

Using Bind Variables

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
```

Run the following SQL command to check your Workspace ID:

```
SELECT v('WORKSPACE_ID') FROM dual
```

See Also: ["Running a SQL Command"](#) on page 2-5, ["About Transactions in SQL Commands"](#) on page 2-5, ["About Unsupported SQL*Plus Commands"](#) on page 2-6, and ["Using Bind Variables"](#) on page 2-6

Using the Find Tables Icon

The Find Tables icon resembles a flashlight. Click this icon 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 the **Find Tables** icon.

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.

Search ☐ Views Schema: DRUE Rows: 15 [Help](#)

Tables Columns

Table: DEMO_ORDERS

| Column | Data Type | Length | Precision | Scale | Nullable |
|-----------------|-----------------------------------|--------|-----------|-------|----------|
| ORDER_ID | NUMBER | 22 | - | - | No |
| CUSTOMER_ID | NUMBER | 22 | - | - | No |
| ORDER_TOTAL | NUMBER | 22 | 8 | 2 | Yes |
| ORDER_TIMESTAMP | TIMESTAMP(6) WITH LOCAL TIME ZONE | 11 | - | 6 | Yes |
| USER_NAME | VARCHAR2 | 100 | - | - | Yes |
| TAGS | VARCHAR2 | 4000 | - | - | Yes |
| 1 - 6 | | | | | |

```

select
ORDER_ID,
CUSTOMER_ID,
ORDER_TOTAL,
ORDER_TIMESTAMP,
USER_NAME,
TAGS
from DEMO_ORDERS

```

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. Click **Replace** to replace existing command in the command editor with the SQL from the page.

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.

Copying a Command

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.

Viewing Results

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

Topics:

- [Accessing the Results Pane](#)
- [About the Results Pane](#)

Accessing the Results Pane

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.

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.

| Results Explain Describe Saved SQL History | | | |
|--|------------------|------------|-------------|
| DEPARTMENT_ID | DEPARTMENT_NAME | MANAGER_ID | LOCATION_ID |
| 10 | Administration | 200 | 1700 |
| 20 | Marketing | 201 | 1800 |
| 30 | Purchasing | 114 | 1700 |
| 40 | Human Resources | 203 | 2400 |
| 50 | Shipping | 121 | 1500 |
| 60 | IT | 103 | 1400 |
| 70 | Public Relations | 204 | 2700 |
| 80 | Sales | 145 | 2500 |
| 90 | Executive | 100 | 1700 |
| 100 | Finance | 108 | 1700 |
| More than 10 rows available. Increase rows selector to view more rows. | | | |
| 10 rows returned in 0.00 seconds Download | | | |

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.
- **Export 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.

See Also: ["Accessing the Results Pane"](#) on page 2-9

Using Explain Plan

You can view the explain plan the Oracle Optimizer uses to run your SQL command. It is not necessary to execute the command to view the explain plan.

Topics:

- [Viewing an Explain Plan](#)
- [About Explain Plan Pane](#)

Viewing an Explain Plan

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.

About Explain Plan 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.

| Results Explain Describe Saved SQL History | | | | | | | | | |
|--|-------------|----------------------|-----------------------------------|-------------|------------|--------|---------------------|-------------------|--|
| Query Plan | | | | | | | | | |
| Operation | Options | Object | Rows | Time | Cost | Bytes | Filter Predicates * | Access Predicates | |
| SELECT STATEMENT | | | 10 | 1 | 3 | 260 | | | |
| TABLE ACCESS | FULL | DEMO_ORDERS | 10 | 1 | 3 | 260 | | | |
| * Unindexed columns are shown in red | | | | | | | | | |
| Index Columns | | | | | | | | | |
| Owner | Table Name | Index Name | Used in Plan | Columns | Uniqueness | Status | Index Type | Join Index | |
| DRUE | DEMO_ORDERS | DEMO_ORDER_PK | | ORDER_ID | UNIQUE | VALID | NORMAL | NO | |
| | | DEMO_ORD_CUSTOMER_IX | | CUSTOMER_ID | NONUNIQUE | VALID | NORMAL | NO | |
| Table Columns | | | | | | | | | |
| Table Owner | Table Name | Column Name | Data Type | | | | | | |
| DRUE | DEMO_ORDERS | ORDER_ID | NUMBER | | | | | | |
| | | CUSTOMER_ID | NUMBER | | | | | | |
| | | ORDER_TOTAL | NUMBER | | | | | | |
| | | ORDER_TIMESTAMP | TIMESTAMP(6) WITH LOCAL TIME ZONE | | | | | | |
| | | USER_NAME | VARCHAR2 | | | | | | |
| | | TAGS | VARCHAR2 | | | | | | |

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.

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.

Topics:

- [Accessing Saved Commands](#)
- [About the Saved SQL Pane](#)

Accessing Saved Commands

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.

See Also: ["About the Saved SQL Pane"](#) on page 2-12

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.

| Results Explain Describe Saved SQL History | | |
|--|--|--|
| Find <input type="text"/> | | |
| <input type="button" value="Go"/> | | |
| Time ▲ | SQL | Schema |
| 6 minutes ago | select ORDER_ID, CUSTOMER_ID, ORDER_TOTAL, ORDER_TIMESTAMP, USER_NAME, TAGS from DEMO_ORDERS | DRUE |
| 27 minutes ago | select * from DEMO_STATES | DRUE |
| 17 hours ago | select * from DEPT | DRUE |
| 17 hours ago | select * from DEMO_STATES | DRUE |
| 17 hours ago | select * from EMP | DRUE |
| | | 1 - 5 <input type="button" value="➤"/> |

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**.

Using SQL Command History

Commands you have executed are stored in the command history regardless of whether you explicitly save them. You use SQL Command History to access commands you have executed in SQL Commands.

Topics:

- [Accessing a Command from Command History](#)
- [About the History Pane](#)

Accessing a Command from Command History

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.

About the History Pane

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

| Results Explain Describe Saved SQL History | | |
|--|--|---------|
| Find <input type="text"/> | | |
| Go | | |
| Time ▲ | SQL | Schema |
| 6 minutes ago | select ORDER_ID, CUSTOMER_ID, ORDER_TOTAL, ORDER_TIMESTAMP, USER_NAME, TAGS from DEMO_ORDERS | DRUE |
| 27 minutes ago | select * from DEMO_STATES | DRUE |
| 17 hours ago | select * from DEPT | DRUE |
| 17 hours ago | select * from DEMO_STATES | DRUE |
| 17 hours ago | select * from EMP | DRUE |
| | | 1 - 5 > |

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.
- **Sort by time.** Click the Time column heading to sort the command history by least recent or most recent.

See Also: ["Accessing a Command from Command History"](#) on page 2-13

Using SQL Scripts

This section provides information on how to use SQL Scripts to create, edit, view, run, and delete script files.

Topics:

- [What is SQL Scripts?](#)
- [Accessing SQL Scripts](#)
- [Creating a SQL Script](#)
- [Using the Script Editor](#)
- [Deleting a SQL Script](#)
- [Copying a SQL Script](#)
- [Executing a SQL Script](#)
- [Viewing SQL Script Results](#)
- [Exporting and Importing SQL Scripts](#)
- [Viewing Script and Result Quotas](#)

What is 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.

See Also: ["Using SQL Scripts"](#) on page 3-1, ["Using SQL Commands"](#) on page 2-1 and ["About Long Operations"](#) on page 4-34

Accessing SQL Scripts

To access SQL Scripts:

1. Log in to the Workspace home page.
2. Click the **SQL Workshop** icon and then **SQL Scripts**.



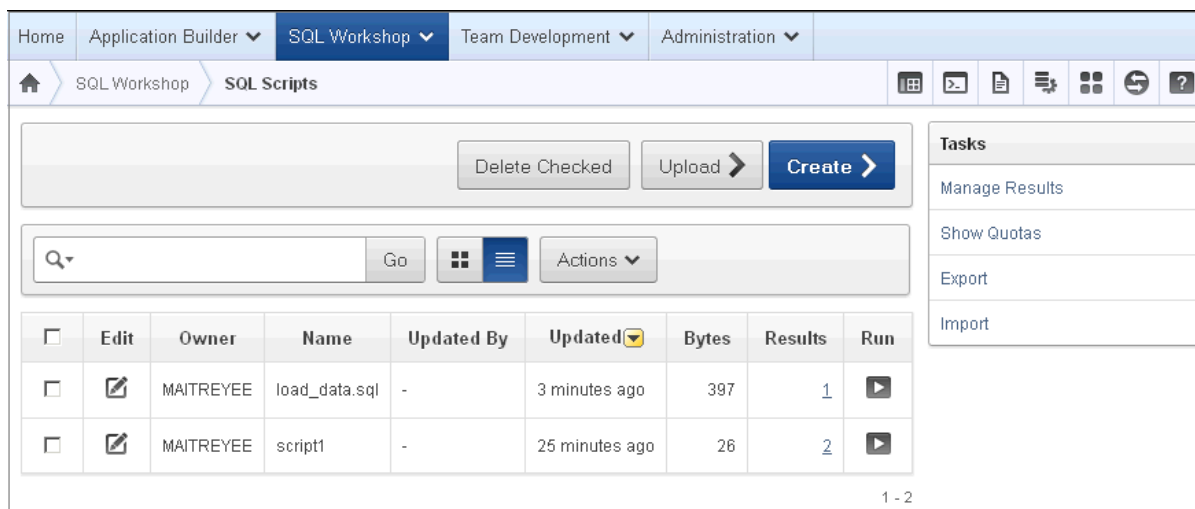
The SQL Scripts page appears.

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. See "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*.

See Also: ["About the SQL Scripts Page"](#) on page 3-2, ["Using SQL Commands"](#) on page 2-1 and ["About Long Operations"](#) on page 4-34

About the SQL Scripts Page

The SQL Scripts page displays an Interactive Report of all SQL scripts created by the current user. 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 See "Customizing Interactive Reports" in the *Oracle Application Express Application Builder User's Guide*. The Report view displays each script as a row in a report.



The SQL Scripts page features the following controls:

- **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**. See ["Deleting a SQL Script"](#) on page 3-8.
- **Sort.** In Reports view, click a column heading to sort the listed scripts by that column. For additional customization options, see ["Customizing Interactive Reports"](#) in the *Oracle Application Express Application Builder User's Guide*.
- **Upload.** Click **Upload** to upload a script from your local file system into SQL Scripts. See ["Creating a SQL Script"](#) on page 3-4.
- **Create.** Click **Create** to create a script in the Script Editor. See ["Creating a SQL Script"](#) on page 3-4.

Switching to Another SQL Workshop Component

You can navigate to another SQL Workshop component by making a selection from the Component list located on the upper right side of the page.



Available icons include:

- **Object Browser.** See ["Managing Database Objects with Object Browser"](#) on page 1-1.
- **SQL Commands.** See ["Using SQL Commands"](#) on page 2-1.
- **SQL Scripts.** See ["Using SQL Scripts"](#) on page 3-1.
- **Query Builder.** See ["Building SQL Queries with Query Builder"](#) on page 4-1.
- **Utilities.** See ["Using Oracle Application Express Utilities"](#) on page 4-1.
- **RESTful Services.** See ["Accessing RESTful Services"](#) on page 5-3.
- **Help.**

About the Tasks List

A Tasks list displays on the right side of the SQL Scripts page.

| Tasks |
|--------------------------------|
| Manage Results |
| Show Quotas |
| Export |
| Import |

The Task list contains the following links:

- **Manage Results** enables you to view, search, and display results. See ["Viewing SQL Script Results"](#) on page 3-10.
- **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. See ["Exporting and Importing SQL Scripts"](#) on page 3-12.
- **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. See ["Exporting and Importing SQL Scripts"](#) on page 3-12.

See Also: ["Accessing SQL Scripts"](#) on page 3-1

Creating a SQL Script

You can create a script in the Script Repository by:

- Creating a script in the Script Editor
- Uploading a script from your local file system

Topics:

- [Creating a SQL Script in the Script Editor](#)
- [Uploading a SQL Script](#)

Creating a 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. Enter a name for the script in the Script Name field.
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 **Save** to save your script to the repository.
The SQL Scripts page appears listing your newly saved script.

See Also: ["Uploading a SQL Script"](#) on page 3-5

Uploading a SQL Script

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. Use **Choose File** button to select the script you want to upload.
4. Optionally rename the script by entering the new name in the Script Name field.
This is the name given to the script in the Script Repository.
5. Click **Upload** to add the script to the Script Repository.

The SQL Scripts page appears listing your newly uploaded script.

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 in the Script Repository, you are prompted to rename it.

See Also: ["Creating a SQL Script in the Script Editor"](#) on page 3-4

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.

Topics:

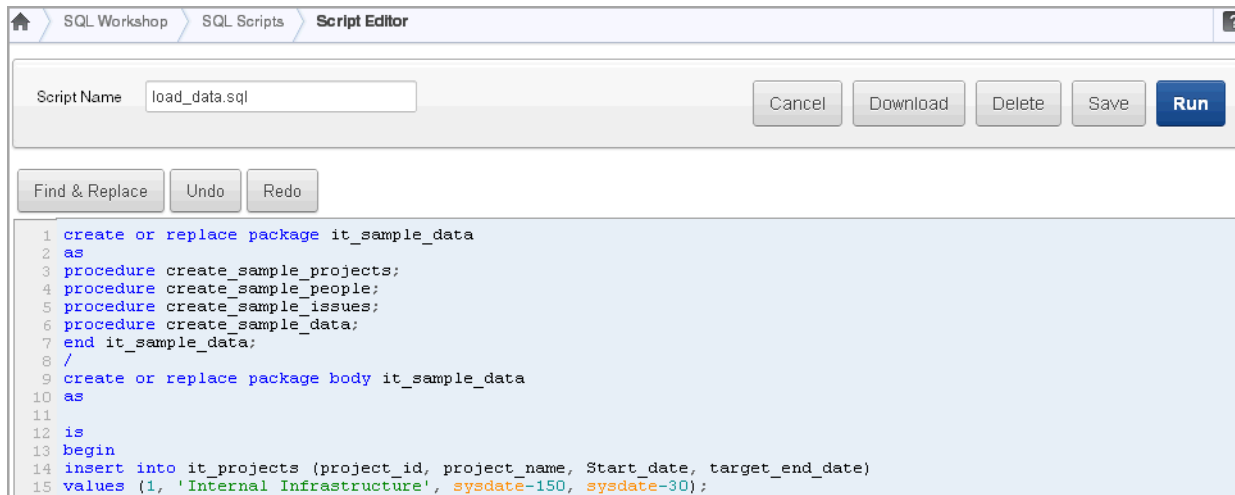
- [Editing an Existing Script](#)
- [Searching and Replacing Text or Regular Expressions](#)
- [Summary of Script Editor Controls](#)

Editing an Existing 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:

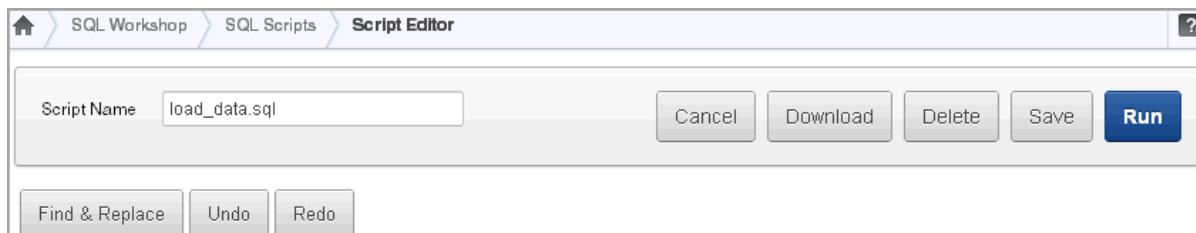
- **Search and Replace.** Click **Find & Replace** to display the text and JavaScript regular expression find and replace options. Click **Find & Replace** again to hide the options. See ["Searching and Replacing Text or Regular Expressions"](#) on page 3-6.
- **Cut, Copy, and Paste.** Use standard edit controls to cut, copy and paste content in the Script Editor.
- **Undo.** Use to undo the last edit.
- **Redo.** Use to reapply the last undone edit.

You can test your script during editing by running the script to reveal errors. The Run Script dialog and the Script Results pages enable you to resume editing the script. See ["Executing a SQL Script"](#) on page 3-9 and ["Viewing SQL Script Results"](#) on page 3-10.

4. Click **Save** to save your script to the Script Repository,
The SQL Scripts page appears.

Searching and Replacing Text or Regular Expressions

Clicking the **Find & Replace** button in the Script Editor displays the Find and Replace with fields at the top of the page. Use these fields to search for and replace text strings and JavaScript regular expressions within a script. To exit Find mode, click **Find** again.



To access Find mode in the Script Editor:

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

The SQL Scripts page appears.

2. Select a script.

The Script Editor appears.

3. Click the **Find & Replace** button.

The Find and Replace fields appear.

4. In the Find field, enter the string you want to find. In Replace with, enter the new string to be added and then click the appropriate button (Find Next, Replace, or Replace All.)

To further refine your search, select the Match Case check box.

5. To exit Find mode, click **Find & Replace**.

Summary of Script Editor Controls

[Table 3–1](#) describes the buttons and controls available within the Script Editor

Table 3–1 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. See Also: "Deleting a SQL Script" on page 3-8 |
| Save | Save your changes to the current script to the Script Repository. |
| Run | Submits the script for execution. See Also: "Executing a SQL Script" on page 3-9 |
| 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 & Replace | Click Find & Replace to access search and replace mode. Click Find & Replace again to exit Find mode. See Also: "Searching and Replacing Text or Regular Expressions" on page 3-6 |

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.

Topics:

- [Deleting Scripts from the SQL Scripts Page](#)
- [Deleting a Script in the Script Editor](#)

Deleting 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 Edit icon. 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. You are prompted to confirm this action before the script is deleted.
The message "Script(s) deleted" appears above the updated list of Scripts.

Deleting 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. You are prompted to confirm this action before the script is deleted.
The SQL Scripts page appears. The message "Script(s) deleted" appears above the updated list of scripts.

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. Rename the script to a name that does not exist in the repository.
4. Click **Save** to copy the script to a new name.

The SQL Scripts page appears listing the newly copied script.

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.

Topics:

- [Executing a SQL Script in the Script Editor](#)
- [Executing a SQL Script from the SQL Scripts Page](#)

See Also: ["About Long Operations"](#) on page 4-34

Executing a SQL 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. See ["Using the Script Editor"](#) on page 3-5.
2. Click **Run** in the Script Editor.
3. The Run Script page appears.

The Run Script 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.

The Run Script page has three controls:

- **Cancel** returns you to the SQL Scripts page without executing the script.
 - **Run in Background** runs the script using DBMS_JOB rather than submitting for execution. This option is useful for long running scripts.
 - **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 script results.
5. To view script results, click the **View** icon under View Results.

See Also: ["Viewing SQL Script Results"](#) on page 3-10

Executing a SQL 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. Click the **Run** icon for the script you want to execute. The Run icon is located on the far right side adjacent to the script name.
4. The Run Script page appears.

The Run Script 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. The Run Script page has three controls:

- **Cancel** returns you to the SQL Scripts page without executing the script.
- **Run in Background** runs the script using DBMS_JOB rather than submitting for execution. This option is useful for long running scripts.
- **Run Now** to submit the script for execution.

- Click **Run Now** to submit the script for execution.

The Manage Script Results page appears listing available results for the script.

- Click the View icon for the results you want to view. The View icon is at the right end of the scripts listed in the Manage Script Results page.

See Also: ["Viewing SQL Script Results"](#) on page 3-10

Viewing SQL Script Results

You use the Manage Script Results page to view and delete script results.

You can also select script results to view from the Icons view of the SQL Scripts page, and from the Results column of the SQL Scripts page Details view.

Topics:

- [Viewing Results from the SQL Scripts Page](#)
- [About the Results Page](#)

Viewing Results from the SQL Scripts Page

To view script results from the SQL Scripts page:

- On the Workspace home page, click **SQL Workshop** and then **SQL Scripts**.
The SQL Scripts page appears.
- On the Tasks list, click **Manage Results**.
- For the results you want to view, click the **View Results** icon.

The Manage Script Results page appears, listing available results for the script.

| <input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions"/> | | | | | | | | |
|---|-------------------------|-----------|----------------|---------|-----------|------------|-------|--------------|
| <input type="checkbox"/> Selected File <input checked="" type="checkbox"/> <input type="checkbox"/> | | | | | | | | |
| <input type="checkbox"/> | Script | Run By | Started | Elapsed | Status | Statements | Bytes | View Results |
| <input type="checkbox"/> | script1 | MAITREYEE | 56 minutes ago | 0.03 | Completed | 1 of 1 | 3,813 | |
| <input type="checkbox"/> | script1 | MAITREYEE | 79 minutes ago | 0.07 | Completed | 1 of 1 | 3,813 | |

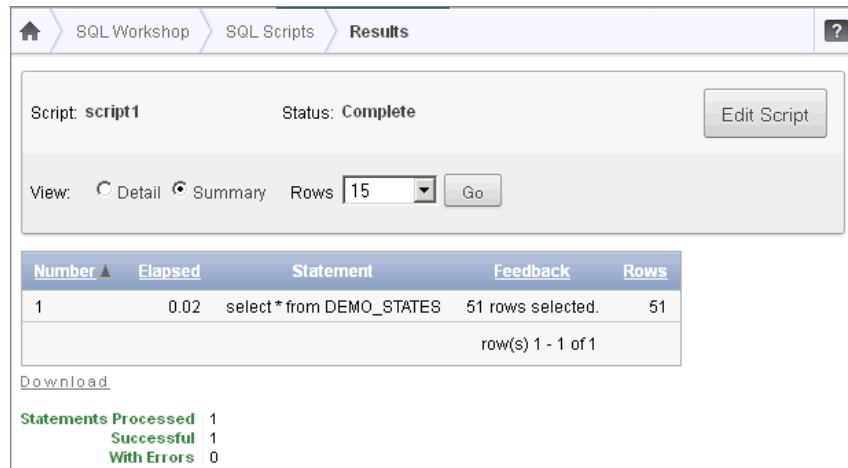
1 - 2

- Click the **View Results** icon for the results you want to view. The View icons display on the far right side of page under the heading View Results.

The Results page appears. See ["About the Results Page"](#) on page 3-11.

About the Results Page

The Results page displays the script name and status (Complete, Canceled, Executing or Submitted), and lists the statements executed.



On the Results page you can:

- **Choose the view.** Click the **Detail** or **Summary** option and click **Go** to specify whether to display complete or summarized information.
- **Choose the number of rows to display.** In Summary view, make a selection from the Display list and click **Go** to specify the number of rows displayed.
- **Sort the statement report.** In Summary view, select a column heading to sort the listed values by that column.
- **Edit the script.** Click **Edit Script** to load the script into the Script Editor. See ["Using the Script Editor"](#) on page 3-5.

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.

Topics:

- [Copying Scripts to an Export Script](#)
- [Importing Scripts from an Export Script](#)

Copying 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. Click the check box for each of the scripts you want to export. The check boxes display on the left side adjacent to the script name. To select all displayed scripts for export, click the column head check box.

4. Click **Add to Export** to create a list of scripts to be added to the export script.

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.

About the Scripts Pane

Scripts [Cancel] [Add To Export]

Use this page to export one or more scripts for importing into another workspace.

Owner: Find: Rows: [Go]

| <input type="checkbox"/> | Owner | Name | Updated ▼ | Bytes |
|--------------------------|-----------|---------------|--------------|-------|
| <input type="checkbox"/> | MAITREYEE | load_data.sql | 14 hours ago | 397 |

row(s) 1 - 1 of 1

Scripts to Export [Remove Checked] [Export All]

File Name:

| Remove | Name | Owner |
|--------------------------|----------|-----------|
| <input type="checkbox"/> | data.sql | MAITREYEE |

row(s) 1 - 1 of 1

In the Scripts pane you can:

- **Search for a script.** Enter a script name or partial name in the Find field and click **Go**. To view all scripts, leave the Find field blank, select **- All Users -** from the Owner list and click **Go**. You control how many rows display by making a selection from the Rows list.
- **Cancel the export.** Click **Cancel** to return to the SQL Scripts page without exporting any scripts, or to return to the SQL Scripts page after saving an export script.
- **Selecting scripts to export.** Click **Add to Export** to add scripts to the export script. Scripts added to the export script are no longer listed in the Script pane, but appear in the Scripts to Export pane.
- **Sort scripts.** Click a column heading to sort the listed scripts by that column.

About the Scripts to Export Pane

Scripts to Export Remove Checked Export All

File Name

| Remove | Name | Owner |
|--------------------------|----------|-----------|
| <input type="checkbox"/> | data.sql | MAITREYEE |

row(s) 1 - 1 of 1

In the Scripts to Export pane you can:

- **Rename the export script.** Enter a name for the export script in the File Name field or leave the default script name.
- **Remove scripts.** Click the check box adjacent to the scripts you want to remove and click **Remove Checked**. Scripts removed are no longer listed in the Scripts to Export pane, but appear in the Scripts pane.
- **Save the export script.** Click **Export All** to save the export script to your local file system. You are prompted to enter the directory where you want to save the export script.

Importing Scripts from an Export Script

To import scripts from 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 **Import**.

The Import Scripts pane appears. See ["About the Import Scripts Pane"](#) on page 3-14.

3. Enter the name and path to the export script you want to import to the Script Repository, or click **Browse** to locate the export script you want to import.

4. Click **Next** to list the scripts in the export script.

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.

About the Import Scripts Pane

In the Import Scripts pane you can:

- **Enter the export script.** Enter the name and path of the script to import in the Import file field, or click **Browse** to locate the script.
- **Cancel the import.** Click **Cancel** to return to the SQL Scripts page without importing scripts.
- **Proceed with the import.** Click **Next** to import the scripts in the specified export script. You can review the listed scripts to import.
- **Choose another export file.** Click **Previous** to return to the Import Scripts file selection page to choose a different export script.

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.
3. Click **Cancel** to return to the SQL Scripts page.

About the Script Quotas Page

The Script Quotas page 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.

Using Oracle Application Express Utilities

This section describes how to use Oracle Application Express 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.

Topics:

- [Building SQL Queries with Query Builder](#)
- [Using the Data Workshop to Manage Data](#)
- [Generating DDL](#)
- [Managing Methods on Tables](#)
- [Managing User Interface Defaults](#)
- [Viewing Database Details](#)
- [Comparing Schemas](#)
- [Viewing Object Reports](#)
- [Monitoring the Database](#)
- [Using the Recycle Bin to View and Restore Dropped Objects](#)

Building SQL Queries with Query Builder

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

Topics:

- [About Query Builder](#)
- [Using the Object Selection Pane](#)
- [Selecting Objects](#)
- [Specifying Query Conditions](#)
- [Creating Relationships Between Objects](#)
- [Working with Saved Queries](#)
- [Viewing Generated SQL](#)

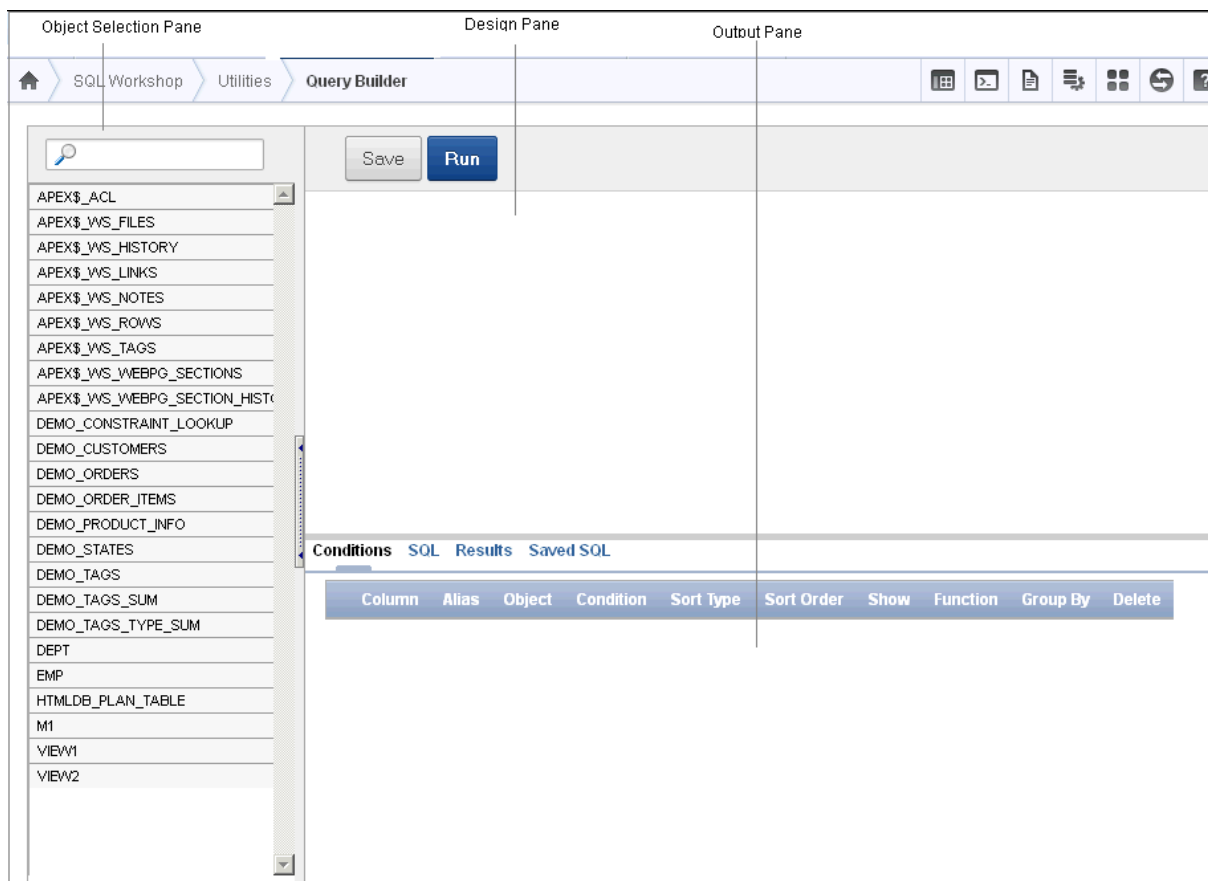
- [Viewing Query Results](#)

See Also: "Using the Query Finder" in *Oracle Application Express Application Builder User's Guide*

About Query Builder

The Query Builder page is divided into three sections:

- **Object Selection pane** displays on the left side of the page and contains a list of 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.



Selecting a Schema

A schema is a logical container for database objects. To access objects in another schema, make a selection from the Schema list in the SQL Workshop home page. Only schemas assigned to your workspace are listed.

Switching to Another SQL Workshop Component

You can navigate to another SQL Workshop component by making a selection from the Component list located on the upper right side of the page.



Available icons include:

- **Object Browser.** See ["Managing Database Objects with Object Browser"](#) on page 1-1.
- **SQL Commands.** See ["Using SQL Commands"](#) on page 2-1.
- **SQL Scripts.** See ["Using SQL Scripts"](#) on page 3-1.
- **Query Builder.** See ["Building SQL Queries with Query Builder"](#) on page 4-1.
- **Utilities.** See ["Using Oracle Application Express Utilities"](#) on page 4-1.
- **RESTful Services.** See ["Accessing RESTful Services"](#) on page 5-3.
- **Help.**

Topics:

- [Accessing Query Builder](#)
- [Understanding the Query Building Process](#)

Accessing Query Builder

To access Query Builder:

1. Log in to the Workspace home page.
2. Click **SQL Workshop**.
3. Click **Utilities**.
4. Click **Query Builder**.

The Query Builder home page appears.

Understanding the Query Building Process

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

1. Select objects from the Object Selection pane. See ["Using the Object Selection Pane"](#) on page 4-4.
2. Add objects to the Design pane and select columns. See ["Selecting Objects"](#) on page 4-5.
3. **Optional:** Establish relationships between objects. See ["Creating Relationships Between Objects"](#) on page 4-9.
4. **Optional:** Create query conditions. See ["Specifying Query Conditions"](#) on page 4-7.
5. Execute the query and view results. See ["Viewing Query Results"](#) on page 4-13.

See Also: ["Viewing Generated SQL"](#) on page 4-13 and ["Working with Saved Queries"](#) on page 4-11

Using the Object Selection Pane

The Object Selection pane displays on the left side of the Query Builder page and lists tables, views, and materialized views within the current schema.

Topics:

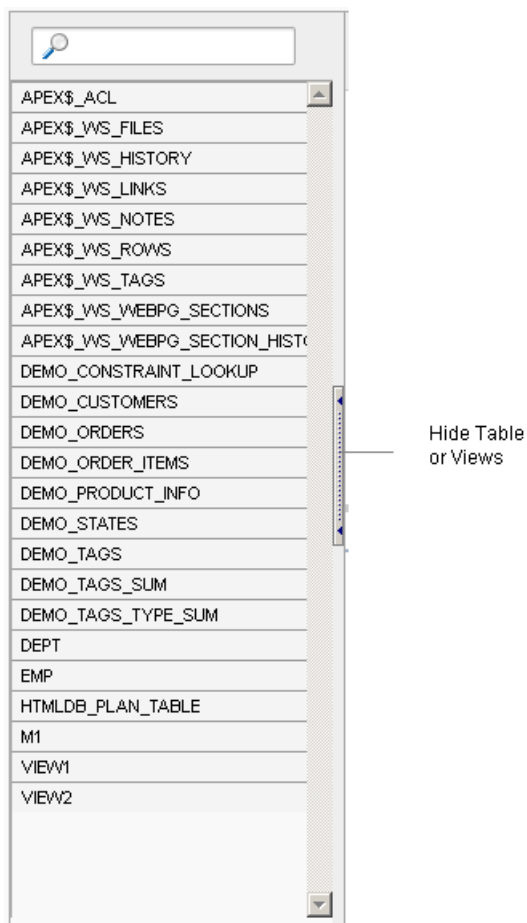
- [Searching and Filtering Objects](#)
- [Hiding the Object Selection Pane](#)

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 pane, enter a case insensitive query.
3. To view all tables or views within the currently selected schema, leave the search field blank.



Hiding the Object Selection Pane

You can hide the Object Selection pane by selecting the **Hide Table or Views** control. By hiding the Object Selection pane, you can increase the size of the Design and Result panes.

The Hide Table or Views control displays on the right side of the Object Selection pane. If the Object list appears, selecting this control hides it. Similarly, if the Object list is hidden, selecting this control causes the pane to reappear.

Selecting 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.

Topics:

- [About Supported Column Types](#)
- [Adding an Object to the Design Pane](#)
- [Removing or Hiding Objects in the Design Pane](#)

See Also: ["Creating Relationships Between Objects"](#) on page 4-9

About Supported Column Types

Columns of all types available in Oracle Database 10g 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

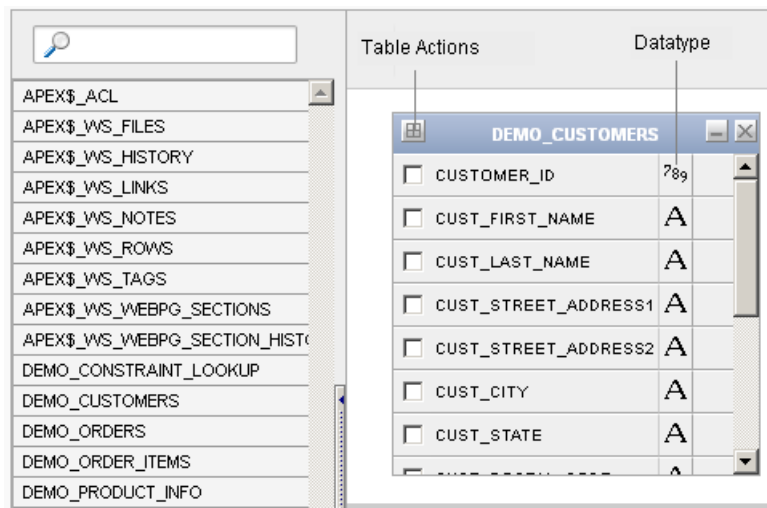
Adding an Object to the Design Pane

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. Note that a graphical representation of the datatype displays to the right of the column name.



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).

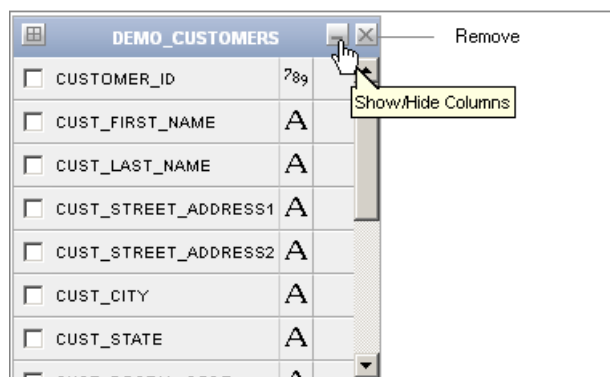
The Results pane displays the query results.

See Also: ["Specifying Query Conditions"](#) on page 4-7

Resizing the Design and Results Panes As you select objects, you can resize Design and Results panes by selecting the gray horizontal rule in the center of the page. Moving the rule up, shrinks the Design pane. Moving the rule down expands the Design pane.

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.



Specifying Query Conditions

Conditions enable you to filter and identify the data you want to work with. As you select columns within an object, you can specify conditions on the Conditions tab. 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 4-1](#) describes the attributes available on the Conditions tab.

Table 4-1 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. See Also: " Viewing Query Results " on page 4-13 |
| 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: <pre>>=10 ='VA' IN (SELECT dept_no FROM dept) BETWEEN SYSDATE AND SYSDATE + 15</pre> |
| Sort Type | Select a sort type. Options include: <ul style="list-style-type: none"> ■ 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 4–1 (Cont.) Conditions Tab

| Condition Attribute | Description |
|---------------------|---|
| Show | <p>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.</p> <p>For example, suppose you want to create following query:</p> <pre>SELECT ename FROM emp WHERE deptno = 10</pre> <p>To create this query in Query Builder:</p> <ol style="list-style-type: none"> 1. From the Object list, select emp. 2. In the Design Pane, select ename and deptno. 3. For the deptno column, in Condition enter =10 and uncheck the Show check box. |
| Function | <p>Select an argument function. Available functions are dependent on the column type and include:</p> <ul style="list-style-type: none"> ■ 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.

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.

Topics:

- [About Join Conditions](#)
- [Joining Objects Manually](#)
- [Joining Objects Automatically](#)

About Join Conditions

When you write a join query, you specify a condition that conveys a relationship between two objects. This condition is called a **join condition**. A join condition determines how the rows from one object combine with the rows from another object.

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.

See Also: *Oracle Database SQL Language Reference* for information about join conditions

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 help tip displays:

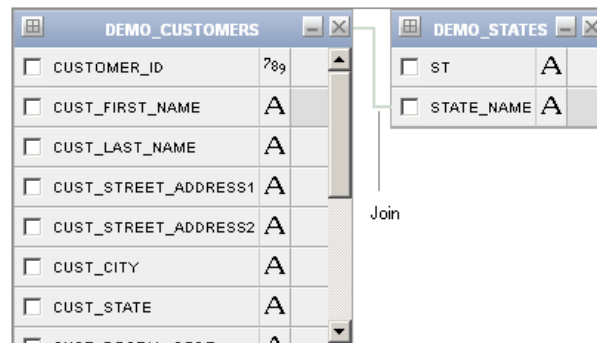
Click here to select column for join

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.

Joining 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.

Working with Saved Queries

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.

Topics:

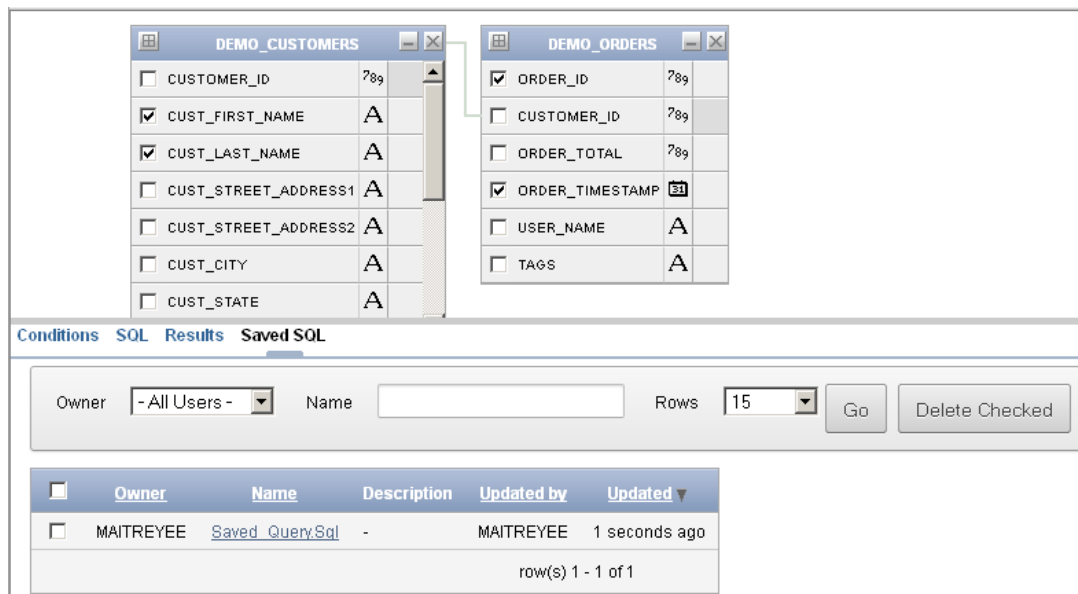
- [Saving a Query](#)
- [Editing a Saved Query](#)
- [Deleting a Saved Query](#)

Saving 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.

Editing a Saved Query

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

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.

Deleting a Saved 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**.

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 Application Builder.

See Also: ["Using SQL Commands"](#) on page 2-1

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.

Viewing Query Results

Once you select objects and determine what columns to include in your query, you 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.

| Conditions SQL Results Saved SQL | | | | | | | |
|----------------------------------|-----------|------|--------|------------|------|------|--------|
| EMPNO | JOB | MGR | ENAME | HIREDATE | SAL | COMM | DEPTNO |
| 7839 | PRESIDENT | - | KING | 11/17/1981 | 5000 | - | 10 |
| 7698 | MANAGER | 7839 | BLAKE | 5/1/1981 | 2850 | - | 30 |
| 7782 | MANAGER | 7839 | CLARK | 6/9/1981 | 2450 | - | 10 |
| 7566 | MANAGER | 7839 | JONES | 4/2/1981 | 2975 | - | 20 |
| 7788 | ANALYST | 7566 | SCOTT | 12/9/1982 | 3000 | - | 20 |
| 7902 | ANALYST | 7566 | FORD | 12/3/1981 | 3000 | - | 20 |
| 7369 | CLERK | 7902 | SMITH | 12/17/1980 | 800 | - | 20 |
| 7499 | SALESMAN | 7698 | ALLEN | 2/20/1981 | 1600 | 300 | 30 |
| 7521 | SALESMAN | 7698 | WARD | 2/22/1981 | 1250 | 500 | 30 |
| 7654 | SALESMAN | 7698 | MARTIN | 9/28/1981 | 1250 | 1400 | 30 |
| 7844 | SALESMAN | 7698 | TURNER | 9/8/1981 | 1500 | 0 | 30 |
| 7876 | CLERK | 7788 | ADAMS | 1/12/1983 | 1100 | - | 20 |
| 7900 | CLERK | 7698 | JAMES | 12/3/1981 | 950 | - | 30 |

Using the Data Workshop to Manage Data

The Data Load/Unload wizards in Oracle Application Express enable you to easily load and unload delimited text data to and from the database. The step-by-step wizards have the following features:

- You can load or unload XML files or delimited-field text files (such as comma-delimited (.csv) or tab-delimited files).

- You can load by copying and pasting from a spreadsheet.
- You can omit (skip) columns when loading or unloading.
- You can 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

Topics:

- [About Importing, Exporting, Loading, and Unloading Data](#)
- [Accessing the Data Load/Unload Page](#)
- [Loading Data](#)
- [Unloading Data](#)
- [Using Repository](#)

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). |

| Term | Definition |
|---------|--|
| 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.

Topics:

- [Choosing the Right Import/Export/Load/Unload Option](#)

Choosing the Right Import/Export/Load/Unload Option

The Oracle Database and Oracle Application Express provide several powerful options for importing, exporting, loading, and unloading data. [Table 4–2](#) provides a summary of these options.

Table 4–2 Summary of Oracle Application Express Import/Export Options

| Feature or Utility | Description |
|--|---|
| Data Load/Unload wizards in Oracle Application Express | <ul style="list-style-type: none"> ■ 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 | <ul style="list-style-type: none"> ■ Command-line interface, invoked with <code>sqlldr</code> 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 | <ul style="list-style-type: none"> ■ Command-line interface, invoked with <code>expdp</code> and <code>impdp</code> 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 |

Table 4–2 (Cont.) Summary of Oracle Application Express Import/Export Options

| Feature or Utility | Description |
|-----------------------------|--|
| Export and Import utilities | <ul style="list-style-type: none"> Command-line interface, invoked with <code>exp</code> and <code>imp</code> commands Exports and imports from one Oracle database to another (proprietary binary format) Supports XMLType data Does not support the <code>FLOAT</code> and <code>DOUBLE</code> data types Capabilities similar to Data Pump; Data Pump is preferred |

Table 4–3 provides several load, unload, import, and export scenarios and suggests the appropriate option to use for each.

Table 4–3 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 |
| 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. There is no XMLType data in any of the data. | 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 <code>FLOAT</code> or <code>DOUBLE</code> data types. | Import (<code>imp</code>) and Export (<code>exp</code>) |

See Also: *Oracle Database Utilities* for more information on Data Pump, the Import and Export utilities, and SQL*Loader

Accessing the Data Load/Unload Page

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.

Loading Data

You can load data into the Oracle Application Express database in the following ways:

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

Topics:

- [Loading a Text File or Spreadsheet Data](#)
- [Loading an XML Document](#)

See Also: ["Choosing the Right Import/Export/Load/Unload Option"](#) on page 15.

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 or enter the name of the Text File to load.
 - Separator - Enter the column separator character. See Item Help for more details.
 - Optionally Enclosed By - If your data contains a delimiter character, enter it here. See Item 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 Item 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. See Item Help for more details.
- **Group Separator** - If your data contains a character that separates integer groups, for example to show thousands and millions, enter it here. See item Help for more details.
- **Decimal Character** - If your data contains a character that separates the integer and decimal parts of a number, enter it here. See item Help for more details.

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.
- **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 an existing 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.

Loading an XML Document

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 to load the data into and click **Next**.
7. Follow the on-screen instructions.

Unloading Data

You can use the Unload page to export the contents of a table to a text file or XML document.

Topics:

- [Unloading a Text File](#)
- [Unloading to an XML Document](#)

See Also: ["Choosing the Right Import/Export/Load/Unload Option"](#) on page 4-15

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 and click **Next**.
6. Select the table to unload and click **Next**.
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. See item Help for more details.
 - Include Column Names - Check this box to include the name of each column in the text file. See item Help for more details.
 - File Format - Select the appropriate file format. See item Help for more details.
 - File Character Set - Select the character set in which your text file is encoded. See item Help for more details.
9. Click **Unload Data**.
The File Download window appears.
 10. Click **Save** to download the file.

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 and click **Next**.
5. Select the table to unload and click **Next**.
6. Highlight the columns to include.
7. Check **Export as File** to save the output directly to a file.
If you do not check Export as File, the XML document is displayed in your browser and can be saved to a file or cut and pasted as necessary.
8. Optionally enter a Where Clause to limit the records unloaded, and click **Unload Data**.

A browser displays the XML data.

Using Repository

Loaded text data files and spreadsheets are stored in the Repository.

To access the different repositories:

1. On the Workspace home page, click the **Utilities** icon.
2. Click **Data Workshop**.
3. Make one of the following selections:
 - **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.

By clicking the details magnifying glass you can review the failed rows.

- **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.

By clicking the details magnifying glass you can review the failed rows.

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**.

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**.

See Also:

- *Oracle Database SQL Language Reference* for information about DDL statements
- "The Data Dictionary" in *Oracle Database Concepts* for information about the data dictionary

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**.

Managing 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.

Topics:

- [Creating User Interface Defaults for a Table](#)
- [Modifying Table User Interface Defaults](#)
- [Creating User Interface Attributes](#)
- [Modifying User Interface Attributes](#)
- [About Exporting and Importing User Interface Defaults](#)

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. See "[Modifying Table User Interface Defaults](#)" on page 4-23.

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.

The current schema displays to the right of the breadcrumb menu.
4. Click **Manage 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 you want to create defaults for.

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.

Modifying 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.

The current schema displays to the right of the breadcrumb menu.
4. Click **Manage 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 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 Item 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.

Creating User Interface Attributes

Use the Attribute Dictionary Create wizard 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.

The current schema displays to the right of the breadcrumb menu.

4. Click **Manage 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 Item Help for a description of the individual settings on this page.

Modifying User Interface Attributes

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 **Manage 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 Item Help for a description of the individual settings on this page.

About 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. See "Exporting user Interface Defaults" in *Oracle Application Express Application Builder User's Guide*.
2. Import the exported file into the target Oracle Application Express instance. See "Importing User Interface Defaults" in *Oracle Application Express Application Builder User's Guide*.
3. Install the exported file from Export Repository. See "Installing Export Files" in *Oracle Application Express Application Builder User's Guide*.

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.

Viewing Database Details

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. Under Additional Utilities, 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

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. Under additional Utilities, click **Schema Comparison**.
4. On the Schema Comparison page, make the appropriate selections to run the comparison:
 - Schema 1 and Schema 2 - Select the schemas to compare.
 - Compare - Restrict the report to show one object type or select **All** to show all database objects.
 - Search - Enter a case insensitive query for the object name.
 - Rows - To change the number of rows that appear in the report, make a selection from the Display list.
 - Compare - Click **Go** to find the results matching your selections.
 - Show Differences Only or Show Details - Select the type of information you want to review.

Viewing Object Reports

Utilities includes a variety of object reports to help you better manage the objects in your database.

Topics:

- [Table Reports](#)
- [Security Reports](#)
- [PL/SQL Reports](#)
- [Exception Reports](#)
- [All Object Reports](#)

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. In the Table Reports section, select the report you want to view:
 - Table Columns
 - Table Comments
 - Table Constraints
 - Table Statistics
 - Table Storage Sizes
5. To filter a report, enter search criteria in the fields provided or make selections from the lists, and click **Go**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.
6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

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. In the Security Reports section, 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 fields provided or make selections from the lists, and click **Go**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.

6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

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.

Topics:

- [Program Unit Arguments](#)
- [Unit Line Counts](#)
- [Search PL/SQL Source Code](#)

Program Unit Arguments

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. In the PL/SQL Reports section, click **Program Unit Arguments**.
5. To filter a report, enter search criteria in the fields provided or make selections from the lists, and click **Go**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.
6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

Unit Line Counts

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. In the PL/SQL Reports section, click **Unit Line Counts**.
5. To filter the report, enter an object name and click **Go**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.
6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

Search 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. In the PL/SQL Reports section, click **Search PL/SQL Source Code**.
4. To filter the report:
 - a. In From/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**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details
5. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

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**. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.
6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

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. In the All Object Reports section, select the report you want 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. For All reports, you can filter the report:
 - a. Select an object type.
 - b. Enter an object name.
 - c. Click **Go**. See *Customizing Interactive Reports in the Oracle Application Express Application Builder User's Guide* for further details
6. To view reports for a different schema, select the schema from the Schema list on the upper right side of the page.

See Also: *Oracle Database Concepts* for information about the data dictionary

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.

To access any of the icons on the Database Monitor page, you must have an account that has been granted a database administrator role.

Topics:

- [Sessions](#)
- [About System Statistics](#)
- [About Top SQL](#)
- [About Long Operations](#)

Sessions

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.

You must have database administrator privileges to access the Sessions page.

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

The sections that follow describe each report.

Sessions Report

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**.

Locks Report

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**.

Waits Report

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**.

I/O Report

The I/O 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**.

SQL Report

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**.

Open Cursors

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**.

About System 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.

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*

About Top SQL

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.

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.

About Long Operations

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.

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*

Using the Recycle Bin to View and Restore Dropped Objects

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.

Topics:

- [Managing Objects in the Recycle Bin](#)
- [Emptying the Recycle Bin Without Viewing the Objects](#)

Note: The Recycle Bin feature is only available if you run with an Oracle 10g or later database.

See Also: "Performing Backup and Recovery" in *Oracle Database 2 Day DBA*

Managing Objects in the Recycle Bin

You can view objects in the Recycle Bin on the Dropped Objects page. 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.

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. See *Customizing Interactive Reports* in the *Oracle Application Express Application Builder User's Guide* for further details.
6. To filter the report, select an object type, enter the object name in the Original Name field, and click **Go**.
7. To view object details, click the object name.

The Object Details page appears.

8. To restore the current object, click **Restore Object**.
9. To permanently delete the current object, click **Purge Object**.

Emptying the Recycle Bin Without Viewing the Objects

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.

Using RESTful Services

RESTful Services enable the declarative specification of RESTful services used to access the database. These services work with the Oracle Application Express Listener to enable the consumption of these services. This section describes how to create, edit and delete RESTful Services.

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 "Configuring SQL Workshop" in *Oracle Application Express Administration Guide*.

Topics:

- [What Is REST?](#)
- [About RESTful Services](#)
- [RESTful Service Requirements](#)
- [Accessing RESTful Services](#)
- [About the Example RESTful Service Module \(oracle.example.hr\)](#)
- [Managing RESTful Service Modules](#)
- [Managing Resource Templates](#)
- [Managing Resource Handlers](#)

What Is REST?

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. A service is described as RESTful when it conforms to the tenets of REST. Although a full discussion of REST is outside the scope of this document, a RESTful Service has the following characteristics:

- Data is modelled as a set of resources. Resources are identified by URLs.
- A small set of operations are used to manipulate resources (for example, PUT, POST, GET, DELETE).
- A resource can have multiple representations (for example, a blog might have a HTML representation and a RSS representation).

- Services are stateless and since it is likely that the client wants to access related resources, these should be identified in the representation returned, typically by providing hypertext links.

About RESTful Services

RESTful Services enable an application to publish Web services that are accessed by other applications to exchange data. For example, you can configure a RESTful Service to return all employee names for a particular department. The data exchange for a RESTful Service follows the Representational State Transfer (REST) architectural style.

For each Web service you want your application to provide, you must configure a RESTful Service Module. This module is a grouping of common templates, known as resource templates under a common Universal Resource Identifier (URI) prefix. For each method associated with the same resource template prefix that you want to expose through this Web service, 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.

When you create a RESTful Service, you supply the necessary information about the structure of the request, response and operation to be performed including:

- A URI Template identifying the set of resources the service provides
- One or more HTTP methods that the service supports along with the SQL query or PL/SQL block to be performed. There are four HTTP methods including GET, DELETE, POST and PUT. Only one handler per HTTP method is permitted. For example, the `empinfo` resource handler can only have one resource handler using GET, and only one using PUT.
- HTTP Header values that should be bound to parameters of the query block
- Parameters expected by the Web service
- Type of input expected by the Web service
- The format of the response

See Also: For an example RESTful Service Module, see "[About the Example RESTful Service Module \(oracle.example.hr\)](#)" on page 5-3

This section describes how to configure RESTful Services. If you want your application to use a Web service, you must configure RESTful Web References. See *Implementing Web Services in Oracle Application Express Application Builder User's Guide*.

RESTful Service Requirements

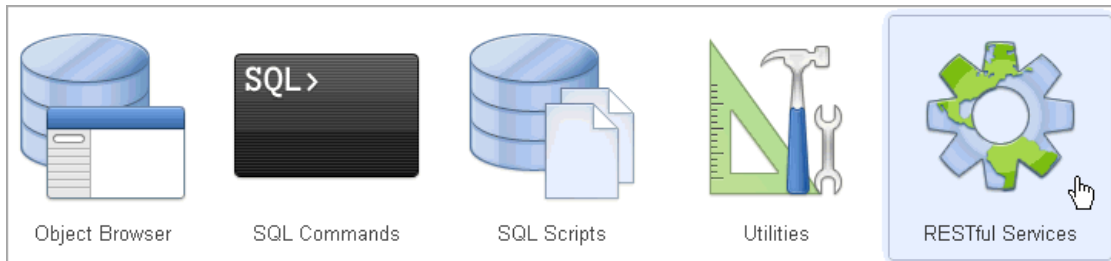
In order for the RESTful Service utility to be available and function properly, the following requirements must be met:

- The instance administrator must enable RESTful Services for this Application Express instance. See *Oracle Application Express Administration Guide*.
- The Oracle Application Express Listener version 2.0 or higher must be used. The Oracle XML DB HTTP Server with the embedded PL/SQL gateway and Oracle HTTP Server with `mod_plsql` do not support this functionality. See *Oracle Application Express Listener Installation and Developer Guide*.

Accessing RESTful Services

To access RESTful Services:

1. Log in to the Workspace home page.
2. Click **SQL Workshop**.
3. Click **RESTful Services**.



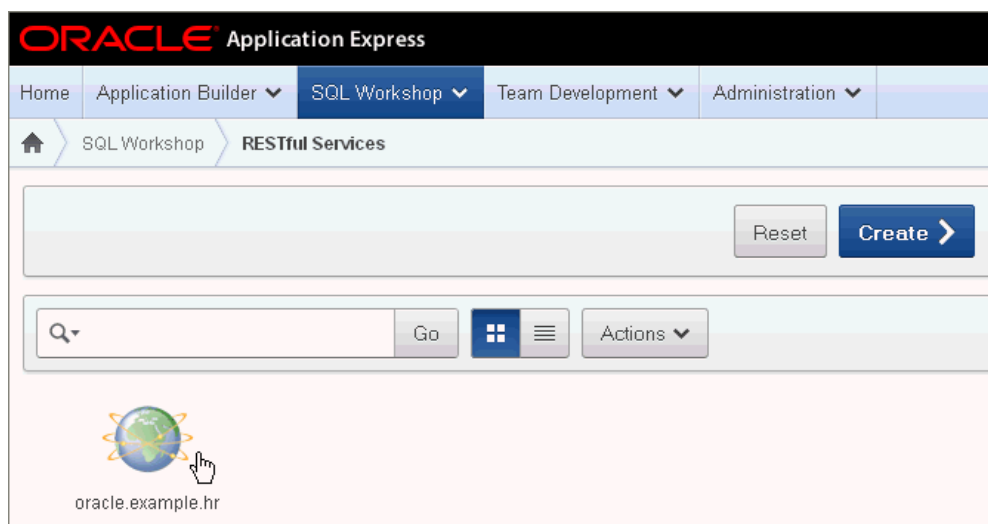
The RESTful Services home page displays.

About the Example RESTful Service Module (oracle.example.hr)

Your Application Express Workspace by default has an example RESTful Service Module, `oracle.example.hr`. 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](#)" on page 5-3.

See Also: For additional RESTful Service examples, go to the Learning Library. Search for free online training content, including Oracle by Example (OBE), demos, and tutorials. To access the Oracle Learning Library, go to:

<http://www.oracle.com/technetwork/tutorials/index.html>



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:

- `empinfo/` - Employee information from the `employees emp` table is returned in CSV format. See ["Returning the Result Set in CSV Format \(empinfo/\)"](#) on page 5-5.
- `employees/{id}` - Employee information from the `employees emp` table for a specified `id` is returned. The curly bracket identifies the parameter sent as part of the URI. See ["Retrieving Data Based on a Parameter \(employees/{id}\)"](#) on page 5-7.
- `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. See ["Returning the Result Set in JSON Format with Pagination Set \(employees/\)"](#) on page 5-9.
- `employeesfeed/` - Employee information from the `employees emp` table is returned in CVS format. See ["Returning the Result Set as a Feed \(employeesfeed/\)"](#) on page 5-12.
- `employeesfeed/{id}` - Employee information from the `employees emp` table for a specified `id` is displayed as a feed. The curly bracket identifies the parameter sent as part of the URI. See also ["Retrieving Data Based on a Parameter \(employees/{id}\)"](#) on page 5-7 and ["Returning the Result Set as a Feed \(employeesfeed/\)"](#) on page 5-12.
- `empsec/{empname}` - Employee information for a specified employee name, `empname`, is returned and displayed in JSON. The curly bracket identifies the parameter sent as part of the URI. See also ["Retrieving Data Based on a Parameter \(employees/{id}\)"](#) on page 5-7.

- `empsecformat/{empname}` - Employee information for a specified employee name, `empname`, is returned and formatted based on the execution of a PL/SQL block. The curly bracket identifies the parameter sent as part of the URI. See ["Returning the Result Set Based on a PL/SQL Block \(`empsecformat/{empname}`\)"](#) on page 5-13.

Returning the Result Set in CSV Format (`empinfo/`)

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 **Create**.
3. For Name, enter `my.example.demo`.
4. For URI Prefix, enter `demo/`. This is the base of the Uniform Resource Identifier (URI) that is used to access this RESTful Service.
5. Under Add a Resource Template, for URI Template enter `empinfo/`.
6. Under Add a Resource Handler, for Method select **GET**.
The Resource Handler settings appear.
7. Make the following selections and entries:
 - 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
```

RESTful Services Module

Name

my.example.demo

URI Prefix

demo/

Origins Allowed

Status

Published

Pagination Size

25

Required Privilege

- Assign Privilege -

Add a Resource Template

URI Template

empinfo/

Priority

0

Entity Tag

Secure HASH

Add a Resource Handler

Method

GET

Source Type

Query

Format

JSON

Source

select * from emp

8. Click Create Module.

The RESTful Service Module page displays.

9. Under empinfo/, click the GET resource handler.

10. Click Test.

An Opening dialog appears, allowing you to open or save the CSV file containing the results of executing the resource handler source SQL query.

11. Select Save File and click OK.

12. Open the saved file in Microsoft Excel or another editor to view the contents of the CSV file.

All data from the emp table is included in the CSV file.

| | A | B | C | D | E | F | G | H |
|----|------|--------|-----------|------|------------|------|------|----|
| 1 | 7839 | KING | PRESIDENT | | 1981-11-11 | 5000 | | 10 |
| 2 | 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | | 30 |
| 3 | 7782 | CLARK | MANAGER | 7839 | 1981-06-01 | 2450 | | 10 |
| 4 | 7566 | JONES | MANAGER | 7839 | 1981-04-01 | 2975 | | 20 |
| 5 | 7788 | SCOTT | ANALYST | 7566 | 1982-12-01 | 3000 | | 20 |
| 6 | 7902 | FORD | ANALYST | 7566 | 1981-12-01 | 3000 | | 20 |
| 7 | 7369 | SMITH | CLERK | 7902 | 1980-12-11 | 800 | | 20 |
| 8 | 7499 | ALLEN | SALESMAN | 7698 | 1981-02-21 | 1600 | 300 | 30 |
| 9 | 7521 | WARD | SALESMAN | 7698 | 1981-02-21 | 1250 | 500 | 30 |
| 10 | 7654 | MARTIN | SALESMAN | 7698 | 1981-09-21 | 1250 | 1400 | 30 |
| 11 | 7844 | TURNER | SALESMAN | 7698 | 1981-09-01 | 1500 | 0 | 30 |
| 12 | 7876 | ADAMS | CLERK | 7788 | 1983-01-11 | 1100 | | 20 |
| 13 | 7900 | JAMES | CLERK | 7698 | 1981-12-01 | 950 | | 30 |
| 14 | 7934 | MILLER | CLERK | 7782 | 1982-01-21 | 1300 | | 10 |

Note: If your results are not correct, compare your empinfo GET resource handler with the `oracle.example.hr empinfo GET` resource handler.

13. Click **Apply Changes**.

Retrieving Data Based on a Parameter (`employees/{id}`)

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 the RESTful Service Module you want to add the resource template to.
The RESTful Service Module page appears.
3. At the bottom of the left panel, click **Create Template**.
The Resource Template options appear on the right.
4. For URI Template, enter `employees/{id}`.
5. Click **Create**.
The `employees/{id}` resource template displays on the left panel.
6. Under the `employees/{id}` resource template, click **Create Handler**.
The Resource Handler options appear.
7. 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
- Requires Secure Access - select **No**.
- 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
```

Resource Handler

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: **demo/**

URI Template: **employees/{id}**

Method: **GET** ▼

Source Type: **Query One Row** ▼


Requires Secure Access: **No** ▼

Pagination Size:

Source

* Source

```
select * from emp
where empno = :id
```

 **Example**

8. Click Create.

The GET Handler appears under the `employees/{id}` resource template on the left panel.

9. Click Set Bind Variables.

10. For Bind Variable Value, enter 7876.

11. Click Test.

A dialog appears showing the results for the employee record with an employee id of 7876 displays.


```

{
  empno: 7876,
  ename: "ADAMS",
  job: "CLERK",
  mgr: 7788,
  hiredate: "1983-01-12T05:00:00Z",
  sal: 1100,
  deptno: 20
}

```

Note: If your results are not correct, compare your employees/{id} GET resource handler with the oracle.example.hr employees/{id} GET resource handler.

Returning the Result Set in JSON Format with Pagination Set (employees/)

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 the RESTful Service Module you want to add the resource template to.

The RESTful Service Module page appears.

3. At the bottom of the left panel, click **Create Template**.

The Resource Template options appear on the right.

4. For URI Template, enter employees/.

5. Click **Create**.

The employees/ resource template displays on the left panel.

6. Under the employees/ resource template, click **Create Handler**.

The Resource Handler options appear.

7. Make the following selections and entries:

- Method - Select **GET**.
- Source Type - Select **Query**.
- Format - Select **JSON**.
- Requires Secure Access - Select **No**.
- Pagination Size - Enter 7.
- Source - Enter the following:

```

select empno "$uri", empno, ename
from (

```

```

select emp.*
       , row_number() over (order by empno) rn
from emp
) tmp
where rn between :row_offset and :row_count

```

Resource Handler

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: **demo/**

URI Template: **employees/**

Method: **GET**

Source Type: **Query** Format: **JSON**

Requires Secure Access: **No**

Pagination Size: **7**

Source

```

* Source
select empno "$uri", empno, ename
from (
  select emp.*
       , row_number() over (order by empno) rn
  from emp
  ) tmp
where rn between :row_offset and :row_count |

```

 **Example**

8. Click **Create**.

The GET Handler appears under the `employees/` resource template on the left panel.

9. Click **Test**.

The JSON results display. Notice the `uri: {$ref}` for each item invokes the `employees/{id}` RESTful Service for the id specified in the link.



Note: If your results are not correct, compare your employees/GET resource handler with the `oracle.example.hr` employees GET resource handler.

10. Click the first `uri: {$ref}` link in the link for `empno 7369`.
The individual employee record for 7369 displays.
11. Click the Back button in your browser.
The original JSON results appear listing 7 employees.
12. Click the next `{$ref}` link to navigate to the next page of data.

The next 7 rows display with links to the first, previous and the next page.

Returning the Result Set as a Feed (employeesfeed/)

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 the RESTful Service Module you want to add the resource template to.

The RESTful Service Module page appears.

3. At the bottom of the left panel, click **Create Template**.

The Resource Template options appear on the right.

4. For URI Template, enter `employeesfeed/`.

5. Click **Create**.

The `employeesfeed/` resource template displays on the left panel.

6. Under the employees resource template, click **Create Handler**.

The Resource Handler options appear.

7. Make the following selections and entries:

- Method - select **GET**.
- Source Type - select **Feed**.
- Requires Secure Access - select **No**.
- Pagination Size - enter 25.
- Source - enter the following:

```
select empno, ename from emp
order by deptno, ename
```

8. Click **Create**.

The GET resource handler appears under the `employeesfeed/` resource template on the left panel.

9. Click **Test**.

The results display a feed for each employee in the `emp` table.

```

{
  - items: [
    - {
      rel: "edit http://apex.oracle.com/item",
      uri: http://database-june07trial.apex.c9lab.us.oracle.com/apex/employeesfeed/7782,
      content-type: "application/json",
      - item: {
        empno: 7782,
        ename: "CLARK"
      }
    },
    - {
      rel: "edit http://apex.oracle.com/item",
      uri: http://database-june07trial.apex.c9lab.us.oracle.com/apex/employeesfeed/7839,
      content-type: "application/json",
      - item: {
        empno: 7839,
        ename: "KING"
      }
    },
    - {
      rel: "edit http://apex.oracle.com/item",
      uri: http://database-june07trial.apex.c9lab.us.oracle.com/apex/employeesfeed/7934,
      content-type: "application/json",
      - item: {
        empno: 7934,
        ename: "MILLER"
      }
    },
    - {
      rel: "edit http://apex.oracle.com/item",
      uri: http://database-june07trial.apex.c9lab.us.oracle.com/apex/employeesfeed/7876,
      content-type: "application/json",
      - item: {
        empno: 7876,
        ename: "ADAMS"
      }
    },
    - {
      rel: "edit http://apex.oracle.com/item",
      uri: http://database-june07trial.apex.c9lab.us.oracle.com/apex/employeesfeed/7902,
      content-type: "application/json",
      - item: {
        empno: 7902,
        ename: "FORD"
      }
    }
  ],
}

```

Note: If your results are not correct, compare your employeesfeed/ GET resource handler with the oracle.example.hr employeesfeed GET resource handler.

Returning the Result Set Based on a PL/SQL Block (empsecformat / {empname})

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 the RESTful Service Module you want to add the resource template to.
The RESTful Service Module page appears.
3. At the bottom of the left panel, click **Create Template**.
The Resource Template options appear on the right.
4. For URI Template, enter empsecformat / {empname}.

5. Click **Create**.

The empsecformat/{empname} resource template displays on the left panel.

6. Under the empsecformat/{empname} resource template, click **Create Handler**.

The Resource Handler options appear.

7. Make the following selections and entries:

- Method - Select **GET**.
- Source Type - Select **PL/SQL**.
- Requires Secure Access - Select **No**.
- Pagination Size - Leave blank.
- Source - enter the following:

```
DECLARE
    prevdeptno    number;
    deptloc       varchar2(30);
    deptname      varchar2(30);
    CURSOR getemps IS select * from emp
                        where ((select job from emp where ename = :empname)
IN ('PRESIDENT', 'MANAGER'))
                        or deptno = (select deptno from emp where ename =
:empname)
                        order by deptno, ename;
BEGIN
    sys.http.htmlopen;
    sys.http.headopen;
    sys.http.title('Departments');
    sys.http.headclose;
    sys.http.bodyopen;

    for empregs in getemps
    loop

        if empregs.deptno != prevdeptno or prevdeptno is null then
            select dname, loc into deptname, deptloc
            from dept where deptno = (select deptno from emp where ename =
empregs.ename);
            if prevdeptno is not null then
                sys.http.print('</ul>');
            end if;
            sys.http.print('Department ' || deptname || ' located in ' ||
deptloc || '<p/>');
            sys.http.print('<ul>');
            end if;

            sys.http.print('<li>' || empregs.ename || ', ' || empregs.job || ', '
|| empregs.sal || '</li>');

            prevdeptno := empregs.deptno;

        end loop;
    sys.http.print('</ul>');
    sys.http.bodyclose;
    sys.http.htmlclose;
END;
```

Resource Handler

A resource handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted.

RESTful Service Module: **demo/**

URI Template: **empsecformat/{empname}**

Method: **GET**

Source Type: **PL/SQL**

Requires Secure Access: **No**

Pagination Size:

Source

* Source

```

DECLARE
  prevdeptno  number;
  deptloc     varchar2(30);
  deptname    varchar2(30);
  CURSOR getemps IS select * from emp
                    where ((select job from emp where ename
= :empname) IN ('PRESIDENT', 'MANAGER'))
                    or deptno = (select deptno from emp where
ename = :empname)
                    order by deptno, ename;

```

[Example](#)

8. Click Create.

The GET resource handler appears under the `empsecformat/{empname}` resource template on the left panel.

9. Click Set Bind Variables.

10. For the empname parameter, enter ADAMS.

11. Click Test.

The results display formatted results for Adams.

database-meyorade31trial.db.
/apex/hr/empsecformat/ADAMS

Department RESEARCH located in DALLAS

- ADAMS, CLERK, 1100
- FORD, ANALYST, 3000
- JONES, MANAGER, 2975
- SCOTT, ANALYST, 3000
- SMITH, CLERK, 800

Note: If your results are not correct, compare your `empsecformat/{empname}` GET resource handler with the `oracle.example.hr empsecformat/{empname}` GET resource handler.

Managing RESTful Service Modules

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.

Topics:

- [Creating a RESTful Service Module](#)
- [Editing a RESTful Service Module](#)
- [Deleting a RESTful Service Module](#)
- [Managing RESTful Service Privileges](#)
- [Exporting a RESTful Service Module](#)
- [Importing a RESTful Service Module](#)

Creating a RESTful Service Module

To create a RESTful Service Module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.
The RESTful Services home page appears.
2. Click **Create**.
3. 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 item Help for that option.
4. Specify the following RESTful Service Module settings:
 - **Name** (required) - Specify the RESTful Service Module name.
 - **URI Prefix** - Specify the URI prefix.
 - **Origins Allowed** - Specify a comma-separated list of origins that are permitted to access the resource template.
 - **Status** - Select **Published** to make this service available for use, otherwise select **Not Published**.
 - **Pagination Size** - Specify the size of the pagination window. For database queries, this is the number of rows to return.
 - **Required Privilege** - Select the user privilege required to access this service module.
5. To add a resource template to this module, specify the following under Add a Resource Template:
 - **URI Template** - Enter the URI template to identify your Uniform Resource Identifier. A URI template is simple syntax for describing URIs, for example:
`example/{id}`
 - **Priority** - Specify the resource template's evaluation order. Higher numbers are evaluated first.

- **Entity Tag** - Specify how the ETag HTTP Header for the resource is generated. See item Help for further details.
 - **Entity Tag Query** - Specify the SQL SELECT statement that returns one column and one row that is to be used to generate a custom entity tag. This option is only visible when the user selects Query for the Entity Tag.
6. If you want to add a resource handler to the resource template, specify the following under Add a Resource Handler:
- **Method** - Specify the HTTP method to be used for the resource handler. Options include: GET, DELETE, POST, PUT. See item Help for further details.
 - **Source Type** - Select the source implementation for the selected HTTP method. Options include: Query, Query One Row, PL/SQL, Feed, Media Resource.
 - **Format** - Define how results are returned. Options include: JSON, CSV.
 - **Source** - Specify the SQL query or PL/SQL block responsible for handling the selected HTTP method.

Note: To define additional resource handlers, complete the Resource Handler Creation wizard and then edit the created resource handler.

7. Click **Create**.

The RESTful Service Module page appears.

Editing a RESTful Service Module

A 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 a 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 RESTful Service Module edit page appears.

The screenshot shows the Oracle Application Express SQL Workshop interface. The top navigation bar includes 'Home', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration'. The breadcrumb trail is 'SQL Workshop > RESTful Services > RESTful Service Module'. On the left, the 'RESTful Service' tree shows a module named 'my.example.demo' with three resource templates: 'empinfo/' (GET), 'employees/' (GET), and 'employees/{id}' (GET). Each template has a '+ Create Handler' link. At the bottom of the tree is a '+ Create Template' link. The right pane displays the configuration for the selected module. At the top, it says 'RESTful Service Module: my.example.demo' with 'Cancel', 'Delete', and 'Apply Changes' buttons. Below this is a section titled 'RESTful Service Module' with a description: 'A RESTful Service Module is a grouping of common templates under a common URI prefix. This prefix is prepended to all templates.' The configuration fields are: 'Name' (my.example.demo), 'URI Prefix' (demo/), 'Origins Allowed' (empty), 'Required Privilege' (- Assign Privilege -), 'Pagination Size' (25), and 'Status' (Published). An 'Example' link is at the bottom.

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 RESTful Services Module. See ["Creating a RESTful Service Module"](#) on page 5-16 for further details.
4. 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. See ["Adding a Resource Template"](#) on page 5-22 for further details.
 - c. Click **Apply Changes**.
5. To edit a resource handler:
 - a. Select the resource handler you want to edit on the RESTful Service tree.
The resource handler options appear on the right panel.
 - b. Make your changes. See ["Adding a Resource Handler"](#) on page 5-24 for further details.
 - c. Click **Apply Changes**.
6. Click **Apply Changes**

Deleting a RESTful Service Module

Deleting a RESTful service module removes the module along with all resource templates and resource handlers associated with this module.

To delete a 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 RESTful Service Module page appears.
3. Click **Delete**.
4. Click **OK** to confirm.
The RESTful Services home page appears.

Managing RESTful Service Privileges

Privileges to access RESTful Service Modules can be restricted to Application Express users that belong to an Application Express user group.

To protect RESTful Service Modules by assigning RESTful Service Privileges:

1. Define an Application Express user group using Application Express Administration. See *Using Groups to Manage Application Express Users in Oracle Application Express Administration Guide*.
2. Create a RESTful Service Privilege based on one or more user groups. See ["Creating a RESTful Service Privilege"](#) on page 5-20.
3. Specify which RESTful Service Modules are protected by this privilege. See ["Assigning Privileges to RESTful Service Modules"](#) on page 5-19.

Topics:

- [Assigning Privileges to RESTful Service Modules](#)
- [Creating a RESTful Service Privilege](#)
- [Editing a RESTful Service Privilege](#)
- [Deleting a RESTful Service Privilege](#)

Assigning Privileges to RESTful Service Modules

There are a couple of ways to assign privileges to RESTful Service Modules:

- During the creation or editing of a RESTful Service Module, you can specify the privilege for the module. See ["Creating a RESTful Service Module"](#) on page 5-16 and ["Editing a RESTful Service Module"](#) on page 5-17.
- During the creation or editing of a RESTful Service Privilege, you can specify the module or modules protected by this privilege. See ["Creating a RESTful Service Privilege"](#) on page 5-20 and ["Editing a RESTful Service Privilege"](#) on page 5-20.

Note: After a RESTful Service Privilege is deleted, all RESTful Service Modules protected by that privilege are unprotected. See ["Deleting a RESTful Service Privilege"](#) on page 5-21.

Creating a RESTful Service Privilege

You can create a 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 a RESTful Service Privilege:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.
The RESTful Services home page appears.
2. Under Tasks, click the **RESTful Services Privileges** link.
The RESTful Service Privileges page appears.
3. Click **Create**.
The RESTful Service Privileges Details page appears.
4. For Name, enter a name to identify the privilege.
5. For Label, enter a label value to identify the privilege.
6. For Assigned Groups, select the user groups this privilege is assigned to.
7. For Description, enter a description of this privilege.
8. For Protected Modules, use the arrows to select the modules this privilege is assigned to.
9. Click **Create**.
The RESTful Service Privileges page appears.

To verify the privilege has been assigned to the appropriate modules:

1. Click **RESTful Services** in the breadcrumb.
The RESTful Services page appears.
2. Click the protected RESTful Service Module.
The RESTful Service Module page appears.
3. Make sure the Required Privilege displays the created RESTful Service Privilege.
4. Repeat these verification steps to ensure all modules assigned to this privilege are protected.

Editing a RESTful Service Privilege

You can modify a 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 RESTful Service Privilege:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.
The RESTful Services home page appears.
2. Under Tasks, click the **RESTful Services Privileges** link.
The RESTful Service Privileges page appears.
3. Click a RESTful Service Privilege.
The RESTful Service Privileges Details page appears.
4. Make modifications.

5. Click **Apply Changes**.

The RESTful Service Privileges page appears.

Deleting a RESTful Service Privilege

You can remove a 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 a RESTful Service Privilege:

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

The RESTful Services home page appears.

2. Under Tasks, click the **RESTful Services Privileges** link.

The RESTful Service Privileges page appears.

3. Click a RESTful Service Privilege.

The RESTful Service Privileges Details page appears.

4. Click **Delete**.

The RESTful Service Privileges page appears.

Exporting a RESTful Service Module

RESTful Service Module meta data 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. See ["Importing a RESTful Service Module"](#) on page 5-21.

To export a RESTful Service module:

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

The RESTful Services home page appears.

2. Under Tasks, click the **Export** link.

The Export RESTful Services page appears.

3. For RESTful Service, perform one of the following:

- To export one module, select the module name.
- To export all modules, select **All Services**.

4. Click **Export**.

The Opening dialog appears.

5. Select export options.

6. Click **OK**.

Importing a RESTful Service Module

RESTful Service Module meta data 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. See ["Exporting a RESTful Service Module"](#) on page 5-21.

Note: If the imported module contains a RESTful Service name that exists in this workspace, the existing RESTful Service is deleted and replaced with the definition from the imported module.

To import a RESTful Service module:

1. On the Workspace home page, click **SQL Workshop** and then **RESTful Services**.
The RESTful Services home page appears.
2. Under Tasks, click the **Import** link.
The Import RESTful Services page appears.
3. For Import File, browse to and select the import file.
4. For File Character Set, select the character set used by the import file.
5. Click **Import**.
The RESTful Services page appears.

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.

Topics:

- [Adding a Resource Template](#)
- [Editing a Resource Template](#)
- [Deleting a Resource Template](#)

Adding a Resource Template

To add a resource template to a 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 RESTful Service for which you want to create a resource template.
3. On the RESTful Service tree, click **Create Template**.
The resource template settings appear on the right panel.
4. 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 item Help for that option.
5. 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.
 - **Entity Tag** - Specify how the ETag HTTP Header for the resource is generated. See item Help for further details.
 - **Entity Tag Query** - Specify the SQL SELECT statement that returns one column and one row that is to be used to generate a custom entity tag. This option is only visible when the user selects Query for the Entity Tag.
6. Click **Create**.

Editing 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 the name of the module the resource template you want to edit belongs to.
The RESTful Service Module page appears.
3. On the RESTful Service tree, click the resource template you want to edit.
The selected resource template is highlighted on the RESTful Service tree and the corresponding resource template settings appear on the right panel.
4. Make the changes you want. See "[Adding a Resource Template](#)" on page 5-22 for further details.
5. Click **Apply Changes**.

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 the name of the module you want to update.
The RESTful Service Module page appears.
3. On the RESTful Service tree, click the resource template you want to delete.
The resource template settings appear on the right panel.
4. Click **Delete**.
5. Click **OK** to confirm.
The RESTful Service Module appears.

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.

Topics:

- [Adding a Resource Handler](#)
- [Editing a Resource Handler](#)
- [Deleting a Resource Handler](#)

Adding a Resource Handler

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 the RESTful Service you want to add a resource handler to.
The RESTful Service Module page appears.
3. On the left panel, under the associated resource template, click **Create Handler**.
The resource handler settings appear on the right panel.
4. 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 item Help for that option.
5. Under Resource Handler, specify the following:
 - **Method** - Specify the HTTP method to be used for the resource handler. See item Help for further details.
 - **Source Type** - Identify the strategy type used to generate the resource. See item help for further information.
 - **Requires Secure Access** - Identify whether the resource should be accessed over a secure channel, for example, HTTPS. Set to Yes if secure access is required
 - **Pagination Size** - Identify the size of the pagination window. For database queries, this is the number of rows to return.
6. Under Source, enter the SQL query or PL/SQL block responsible for handling the selected HTTP method. Click **Example** at the bottom of the page for further details.
7. Click **Create**.
The RESTful Services Module page appears.
8. To add parameters to the resource handler:
 - a. Click **Create Parameter**.
The parameter settings appear on the Resource Handler Parameter panel.
 - b. For **Name**, enter the identifying parameter name.
 - c. For **Bind Variable Name**, enter the parameter bind variable name used within the SQL query or PL/SQL block.

- d. For **Access Method**, select the HTTP method. Options include IN, IN/OUT, or OUT. URI Template parameters can only be IN. A value of IN for a header parameter implies the header will be present in the HTTP request. A value of IN/OUT indicates the value will be present in both the HTTP request and response. A value of OUT indicates the value will only be present in the HTTP response.
- e. For **Source Type**, select the parameter source type.
- f. For **Parameter Type**, select the data type of the parameter.
- g. Click **Create**.
The resource handler page appears with the added parameter included under Parameters at the bottom of the right panel.
- h. For each parameter you want to add, repeat the above steps in this subsection.
9. To test the behavior of the resource handler:
 - a. Click **Apply Changes**.
 - b. If you have bind variables in your SQL query or PL/SQL block, click Set Bind Variables and define the bind variables before testing.
 - c. Click **Test**.

A new window appears displaying the JSON result of executing the RESTful Service. The URL used in the test is composed of the value of the URI prefix and URI Template, along with any parameters.

Note: The Test button and Set Bind Variables test options should only be used for the testing of GET methods, and not for the testing of the other three supported methods: PUT, POST, DELETE.

Note: To make the test results easier to read, the installation of a JSON Viewer in your browser is recommended.

Editing 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 the name of the Resource Service Module the resource handler belongs to.
The RESTful Service Module page appears.
3. On the RESTful Service tree, click the resource handler you want to edit.
The resource handler settings appears on the right panel.
4. Make the changes you want. See ["Adding a Resource Handler"](#) on page 5-24 for further details.
5. Click **Apply Changes**.
The RESTful Service Module page appears.

Deleting 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 the name of the module the resource handler belongs to.
The RESTful Service Module page appears.
3. On the RESTful Service tree, click the resource handler you want to delete.
The resource handler settings appear on the right panel.
4. Click **Delete**.
5. Click **OK** to confirm.
The RESTful Service Module appears.

Index

C

command termination, in SQL Commands, 2-6
Create Lookup Table, 1-10

D

data
 exporting from database, 4-19
 loading, 4-14
 loading into database, 4-17
 unloading, 4-14, 4-19
data dictionary, browsing, 4-30
Data Workshop, 4-14
database
 loading data into, 4-14
 monitoring, 4-30
 unloading data from, 4-14
database definition language, generating DDL
 statements, 4-21
database link
 browsing, 1-30
 creating, 1-30
 testing, 1-31
Database Monitor
 I/O, 4-32
 Locks, 4-32
 Long Operations, 4-34
 Open Cursors, 4-32
 Sessions, 4-31
 SQL, 4-32
 Top SQL, 4-33
 Waits, 4-32
database monitoring, system statistics, 4-33
database objects
 purging, 4-34
 restoring, 4-34
database security
 Column Privileges report, 4-28
 Role Privileges report, 4-28
 System Privileges report, 4-28
DDL
 about, 4-21
 generating, 4-21

E

Explain Plan
 about, 2-10
 using, 2-10, 2-11
Export XML Wizard, 4-20
exporting data
 summary of available methods, 4-15

F

function
 browsing, 1-25
 compiling, 1-26
 creating, 1-24
 downloading, 1-26
 dropping, 1-26
 editing, 1-26

I

importing data, summary of methods, 4-15
index
 browsing, 1-14
 creating, 1-13
 disabling, 1-14
 dropping, 1-14
 rebuilding, 1-14
 statistics about, 1-14
I/O report, 4-32

L

Load Data Wizard, 4-17
Load XML Wizard, 4-19
Locks report, 4-32

M

materialized view
 browsing, 1-32
 creating, 1-31
 dependent objects, 1-33
 dropping, 1-33
 list of grants, 1-33
 report of data, 1-33
 sort by column, 1-33

viewing object details, 1-33

O

Object Browser

- about, 1-2
- accessing, 1-1
- creating objects, 1-5
- Detail pane, 1-2
- Hide Objects control, 1-4
- hiding Object Selection pane, 1-4
- managing database links, 1-30
- managing functions, 1-24
- managing indexes, 1-13
- managing materialized views, 1-31
- managing packages, 1-18
- managing procedures, 1-21
- managing sequences, 1-15
- managing tables, 1-5
- managing triggers, 1-27
- managing types, 1-16
- managing views, 1-10
- Object Selection pane, 1-2
- searching for objects, 1-3
- synonyms, 1-33

Object Reports

- All Objects, 4-30
- All Objects report, 4-30
- Column Privileges report, 4-28
- Data Dictionary, 4-30
- Exception Reports, 4-29
- Invalid Objects, 4-30
- Object Calendar, 4-30
- Object Counts by Type, 4-30
- PL/SQL Source Code report, 4-29
- Program Unit Arguments report, 4-28
- Role Privileges report, 4-28
- System Privileges report, 4-28
- Unit Line Counts report, 4-28

objects

- purging, 4-34
- restoring, 4-34

Open Cursors report, 4-32

Oracle Optimizer

- Explain Plan, 2-11

Oracle Optimizer, Explain Plan, 2-10

P

package

- compiling, 1-20
- creating, 1-18
- downloading, 1-19, 1-21
- dropping, 1-21
- editing, 1-19, 1-20
- finding, 1-19
- grants for, 1-20
- object dependencies, 1-20
- package body, 1-20
- related errors, 1-20
- viewing, 1-19

procedure

- browsing, 1-22
- compiling, 1-23
- creating, 1-22
- downloading, 1-24
- dropping, 1-24
- editing, 1-23
- editing manually, 1-23
- finding, 1-23

Q

Query Builder

- about, 4-2
- accessing, 4-3
- creating joins automatically, 4-10
- creating joins manually, 4-9
- filtering objects, 4-4
- hiding objects in Design Pane, 4-7
- Object Selection pane, 4-5
- removing objects in Design Pane, 4-7
- resizing panes, 4-7
- Saved SQL tab, 4-12
- searching for objects, 4-4
- supported column types, 4-6
- Table Actions, 4-10
- View Actions, 4-10

R

Recycle Bin

- purging, 4-34
- searching, 4-34
- using, 4-34
- viewing objects, 4-34

reports

- Data Dictionary, 4-30
- database objects, 4-30
- exception reports, 4-29
- PL/SQL Source Code, 4-29
- Program Unit Arguments, 4-28
- Unit Line Counts report, 4-28

Representational State Transfer (REST), 5-1

resource handler

- adding, 5-24
- deleting, 5-26
- editing, 5-25

resource template

- adding, 5-22
- deleting, 5-23
- editing, 5-23
- managing, 5-22

RESTful services

- about, 5-2
- about oracle.example.hr, 5-3
- accessing, 5-3
- creating, 5-16
- deleting, 5-19
- editing, 5-17
- example, 5-3
- exporting, 5-21

- importing, 5-21
- managing, 5-16
- managing privileges, 5-19
- managing resource handlers, 5-23
- managing resource templates, 5-22
- requirements, 5-2
- using, 5-1

results, 3-4

S

scripts

- creating in editor, 3-4
- deleting from editor, 3-8
- deleting from SQL Scripts page, 3-4
- executing, 3-9
- executing from SQL Scripts page, 3-9
- executing in editor, 3-9
- export, 3-4
- exporting, 3-12
- importing, 3-12
- results, 3-4
- SQL Script editor, 3-5
- uploading to another account, 3-4
- uploading to SQL Scripts, 3-5
- viewing repository quotas, 3-15

Security reports

- Column Privileges report, 4-28
- Object Grants, 4-27
- Role Privileges, 4-28
- System Privileges, 4-28

sequence

- altering, 1-16
- browsing, 1-16
- creating, 1-15
- dependent objects, 1-16
- dropping, 1-16
- grants associated with, 1-16

Session report, 4-31

session, killing, 4-31

SQL Commands

- command termination, 2-6
- Explain Plan, running, 2-11
- home page, 2-2
- running Explain Plan, 2-10
- transaction support, 2-5

SQL report, 4-32

SQL Scripts

- accessing, 3-1
- creating script in editor, 3-4
- deleting script from editor, 3-8
- deleting script from SQL Scripts home page, 3-8
- executing script from SQL Scripts page, 3-9
- executing script in editor, 3-9
- exporting scripts, 3-12
- importing scripts, 3-12
- script editor, 3-5
- SQL Scripts page, 3-2
- uploading script, 3-5
- viewing quotas, 3-15

- viewing script results, 3-10

SQL Scripts page

- accessing Manage Script Results, 3-4
- Show Quotas, 3-4
- Task list, 3-4

SQL Scripts page Task list

- export scripts, 3-4
- uploading scripts, 3-4

SQL*Plus command support, 2-6

SQL*Plus commands, unsupported, 2-6

synonym

- creating, 1-34
- dropping, 1-34
- viewing, 1-34

T

tab-delimited file

- loading, 4-17

tables

- add column, 1-10
- browsing, 1-7
- constraints, 1-9
- copy, 1-10
- creating, 1-5
- drop column, 1-10
- dropping, 1-10
- editing, 1-9
- exporting user interface defaults, 4-25
- exports data to spreadsheet, 1-8
- grants on, 1-9
- indexes associated, 1-8
- insert row, 1-8
- modify column, 1-10
- remove all rows, 1-10
- rename, 1-10
- rename column, 1-10
- report of number of rows, 1-8
- sort by column, 1-8
- statistics about, 1-9
- user interface defaults, 1-9
- viewing data, 1-8

text file, loading, 4-17

transaction support, 2-5

trigger

- browsing, 1-27
- compiling, 1-29
- creating, 1-27
- downloading, 1-29
- dropping, 1-29
- editing, 1-28

type

- browsing, 1-17
- creating, 1-17
- dropping, 1-17
- grants associated with, 1-17
- list of synonyms, 1-17

U

- Unload to Text Wizard, 4-19

- Unload to XML Wizard, 4-20
- 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, 4-30

V

- view
 - browsing, 1-11
 - compiling, 1-13
 - creating, 1-11
 - dropping, 1-13
 - editing, 1-12
 - grants associated with, 1-12
 - objects referenced by, 1-12
 - report of data, 1-12
 - user interface defaults, 1-12

W

- Waits report, 4-32
- wizards
 - copying and pasting tab-delimited data, 4-17
 - exporting contents of table, 4-19
 - exporting text file, 4-19
 - exporting XML, 4-20
 - importing text, 4-17
 - importing XML, 4-19

X

- XML document
 - exporting to, 4-20
 - loading, 4-19