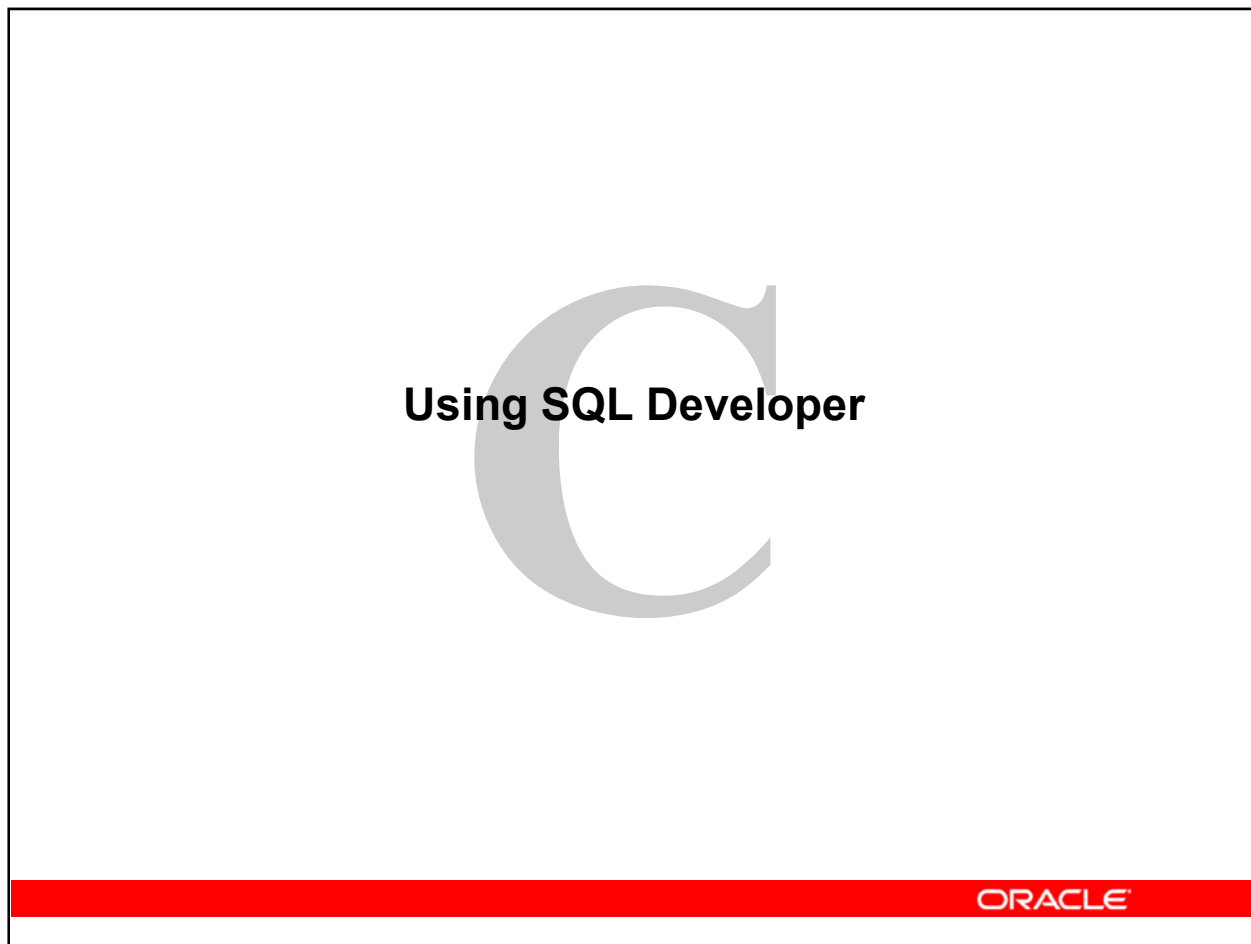


Using SQL Developer

Chapter 17

Using SQL Developer



Objectives

Objectives

After completing this appendix, you should be able to do the following:

- List the key features of Oracle SQL Developer
- Identify menu items of Oracle SQL Developer
- Create a database connection
- Manage database objects
- Use SQL Worksheet
- Save and run SQL scripts
- Create and save reports

ORACLE

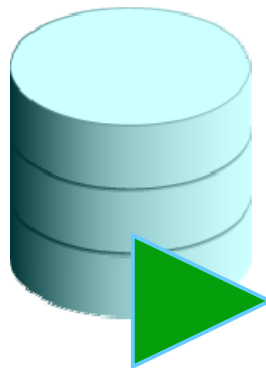
Objectives

In this appendix, you are introduced to the graphical tool called SQL Developer. You learn how to use SQL Developer for your database development tasks. You learn how to use SQL Worksheet to execute SQL statements and SQL scripts.

What Is Oracle SQL Developer?

What Is Oracle SQL Developer?

- Oracle SQL Developer is a graphical tool that enhances productivity and simplifies database development tasks.
- You can connect to any target Oracle database schema by using standard Oracle database authentication.



SQL Developer

ORACLE

What Is Oracle SQL Developer?

Oracle SQL Developer is a free graphical tool designed to improve your productivity and simplify the development of everyday database tasks. With just a few clicks, you can easily create and debug stored procedures, test SQL statements, and view optimizer plans.

SQL Developer, the visual tool for database development, simplifies the following tasks:

- Browsing and managing database objects
- Executing SQL statements and scripts
- Editing and debugging PL/SQL statements
- Creating reports

You can connect to any target Oracle database schema by using standard Oracle database authentication. When connected, you can perform operations on objects in the database.

The SQL Developer tightly integrates with *Developer Migration Workbench* that provides users with a single point to browse database objects and data in third-party databases, and to migrate from these databases to an Oracle database. You can also connect to schemas for selected third-party (non-Oracle) databases such as MySQL, Microsoft SQL Server, and Microsoft Access, and you can view metadata and data in these databases.

Additionally, SQL Developer includes support for Oracle Application Express 3.0.1 (Oracle APEX).

Specifications of SQL Developer

Specifications of SQL Developer

- Shipped along with Oracle Database 11g Release 2
- Developed in Java
- Supports Windows, Linux, and Mac OS X platforms
- Default connectivity by using the Java Database Connectivity (JDBC) thin driver
- Connects to Oracle Database version 9.2.0.1 and later
- Freely downloadable from the following link:
 - http://www.oracle.com/technology/products/database/sql_developer/index.html

ORACLE

Specifications of SQL Developer

Oracle SQL Developer 1.5 is shipped along with Oracle Database 11g Release 2. SQL Developer is developed in Java leveraging the Oracle JDeveloper integrated development environment (IDE). Therefore, it is a cross-platform tool. The tool runs on Windows, Linux, and Mac operating system (OS) X platforms.

Default connectivity to the database is through the JDBC thin driver, and therefore, no Oracle Home is required. SQL Developer does not require an installer and you need to simply unzip the downloaded file. With SQL Developer, users can connect to Oracle Databases 9.2.0.1 and later, and all Oracle database editions including Express Edition.

Note

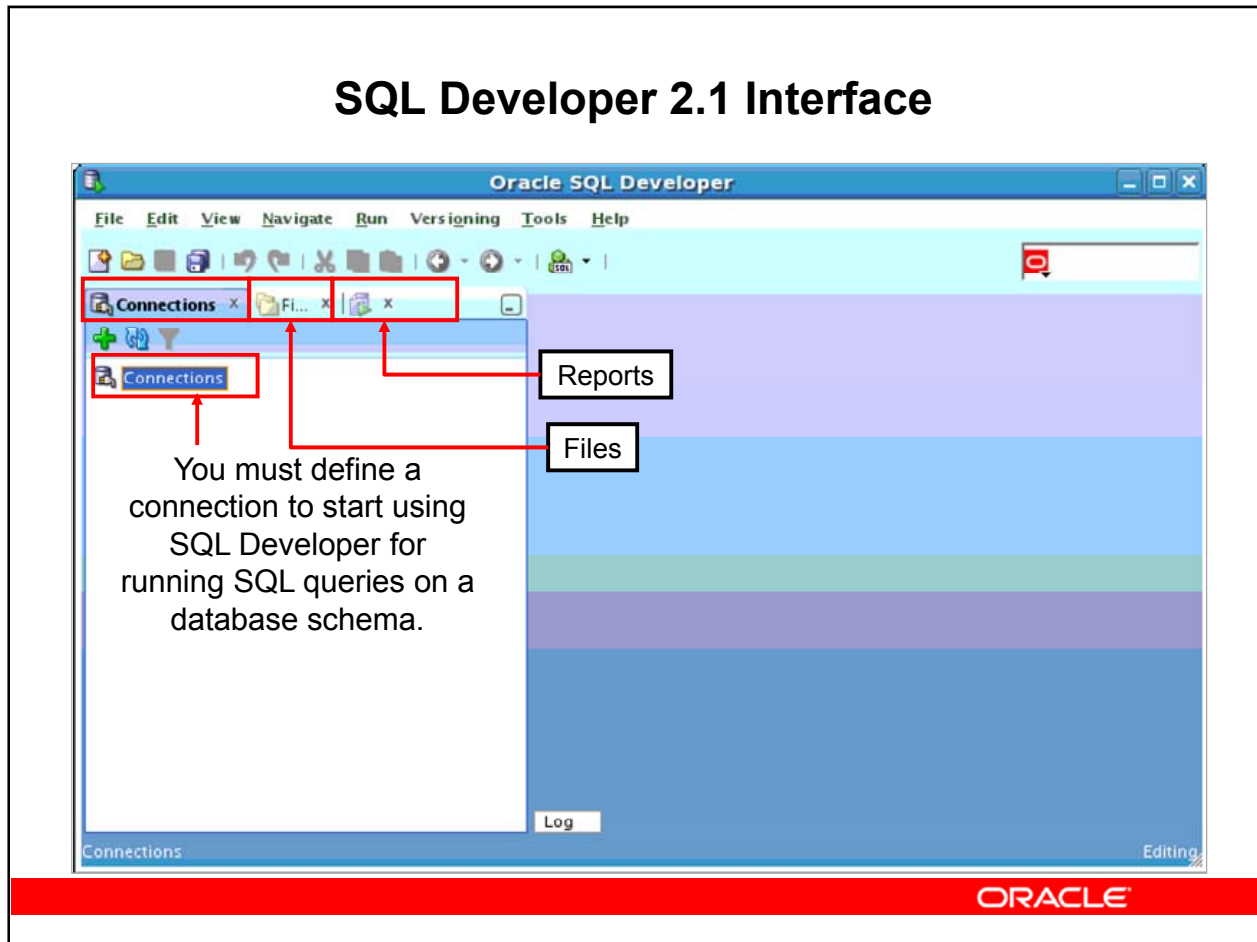
For Oracle Database versions earlier than Oracle Database 11g Release 2, you will have to download and install SQL Developer. SQL Developer 2.1 is the current version and is freely downloadable from the following link:

http://www.oracle.com/technology/products/database/sql_developer/index.html.

For instructions on how to install SQL Developer 2.1, you can visit the following link:

http://download.oracle.com/docs/cd/E15846_01/index.htm

SQL Developer 2.1 Interface



SQL Developer 2.1 Interface

The SQL Developer 2.1 interface contains three main navigation tabs, from left to right:

- **Connections tab:** By using this tab, you can browse database objects and users to which you have access.
- **Files tab:** Identified by the Files folder icon, this tab enables you to access files from your local machine without having to use the File > Open menu. This tab does not appear by default, Use the View > Files menu to activate it.
- **Reports tab:** Identified by the Reports icon, this tab enables you to run predefined reports or create and add your own reports.

General Navigation and Use

SQL Developer uses the left side for navigation to find and select objects, and the right side to display information about selected objects. You can customize many aspects of the appearance and behavior of SQL Developer by setting preferences.

Note: You need to define at least one connection to be able to connect to a database schema and issue SQL queries or run procedures/functions.

Menus

The following menus contain standard entries, plus entries for features specific to SQL Developer:

- **View:** Contains options that affect what is displayed in the SQL Developer interface
- **Navigate:** Contains options for navigating to various panes and for executing subprograms
- **Run:** Contains the Run File and Execution Profile options that are relevant when a function or procedure is selected, and also debugging options
- **Edit:** Contains options for use when you edit functions and procedures
- **Versioning:** Provides integrated support for the following versioning and source control systems: Concurrent Versions System (CVS) and Subversion
- **Migration:** Contains options related to migrating third-party databases to an Oracle database
- **Tools:** Invokes SQL Developer tools such as SQL*Plus, Preferences, and SQL Worksheet

Note: The Run menu also contains options that are relevant when a function or procedure is selected for debugging.

Creating a Database Connection

Creating a Database Connection

- You must have at least one database connection to use SQL Developer.
- You can create and test connections for multiple:
 - Databases
 - Schemas
- SQL Developer automatically imports any connections defined in the `tnsnames.ora` file on your system.
- You can export connections to an Extensible Markup Language (XML) file.
- Each additional database connection created is listed in the Connections Navigator hierarchy.

ORACLE

Creating a Database Connection

A connection is a SQL Developer object that specifies the necessary information for connecting to a specific database as a specific user of that database. To use SQL Developer, you must have at least one database connection, which may be existing, created, or imported.

You can create and test connections for multiple databases and for multiple schemas.

By default, the `tnsnames.ora` file is located in the `$ORACLE_HOME/network/admin` directory, but it can also be in the directory specified by the `TNS_ADMIN` environment variable or registry value. You can use network service names defined in the `tnsnames.ora` file to specify service names for you connections.

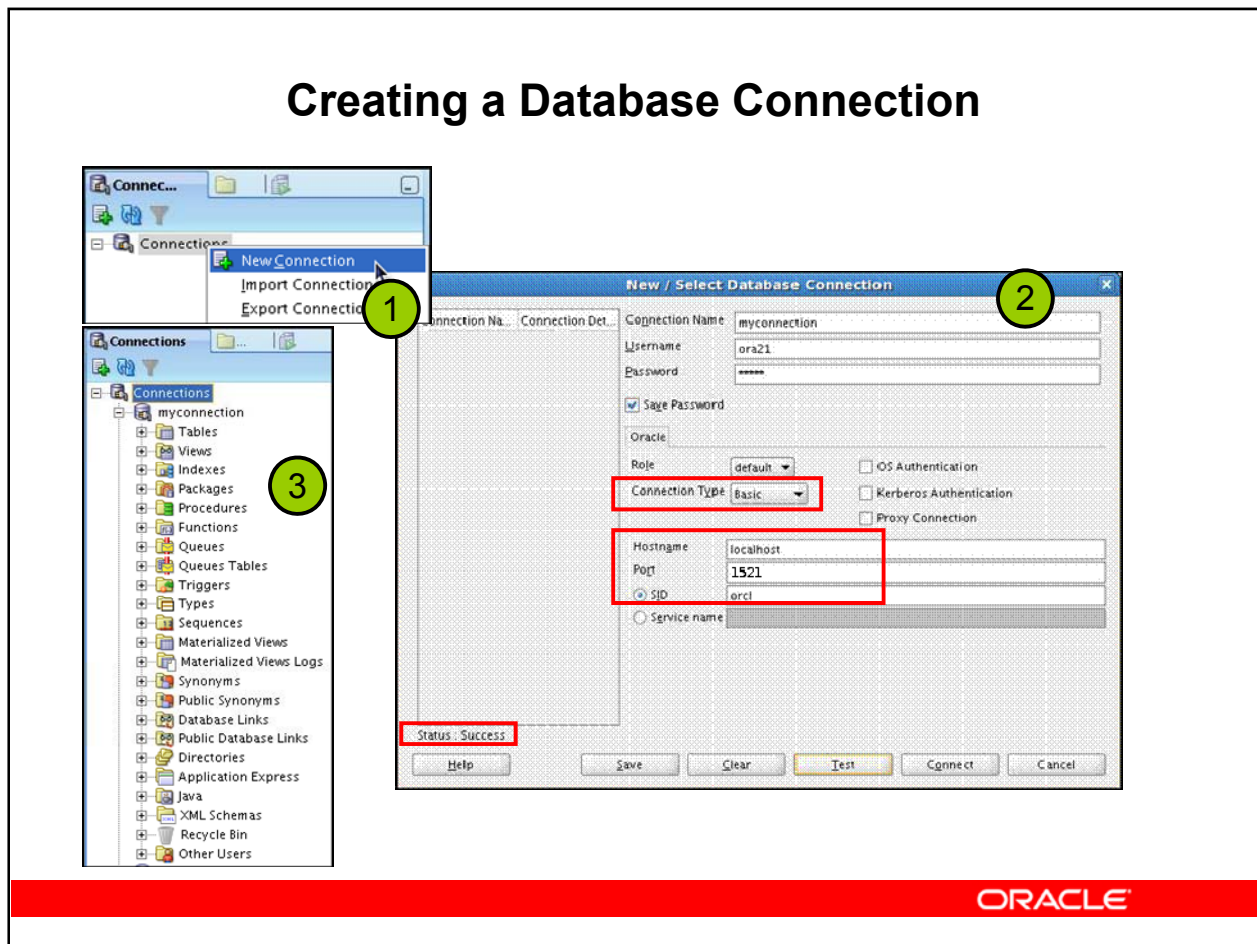
Note: On Windows, if the `tnsnames.ora` file exists but its connections are not being used by SQL Developer, define `TNS_ADMIN` as a system environment variable.

Using the Connection dialog menu you can select Create Local connections which will create a connection for every open account on the local database.

You can export connections to an XML file that you can reuse later.

You can create additional connections as different users to the same database or to connect to the different databases.

Creating a Database Connection



Creating a Database Connection (continued)

To create a database connection, perform the following steps:

1. On the Connections tabbed page, right-click **Connections** and select **New Connection**.
2. In the New/Select Database Connection window, enter the connection name. Enter the username and password of the schema that you want to connect to.
 - a) From the Role drop-down list, you can select either default or SYSDBA. (You choose SYSDBA for the `sys` user or any user with database administrator privileges.)
 - b) You can select the connection type as:
 - **Basic:** In this type, enter the host name and SID for the database you want to connect to. Port is already set to 1521. Or you can also choose to enter the Service name directly if you use a remote database connection.
 - **TNS:** You can select any one of the database aliases imported from the `tnsnames.ora` file.
 - **LDAP:** You can look up database services in Oracle Internet Directory which is a component of Oracle Identity Management.
 - **Advanced:** You can define a custom JDBC URL to connect to the database.
 - c) Click Test to ensure that the connection has been set correctly.

d) Click Connect

If you select the Save Password check box, the password is saved to an XML file. So, after you close the SQL Developer connection and open it again, you are not prompted for the password.

3. The connection gets added in the Connections Navigator. You can expand the connection to view the database objects and view object definitions—for example, dependencies, details, statistics, and so on.

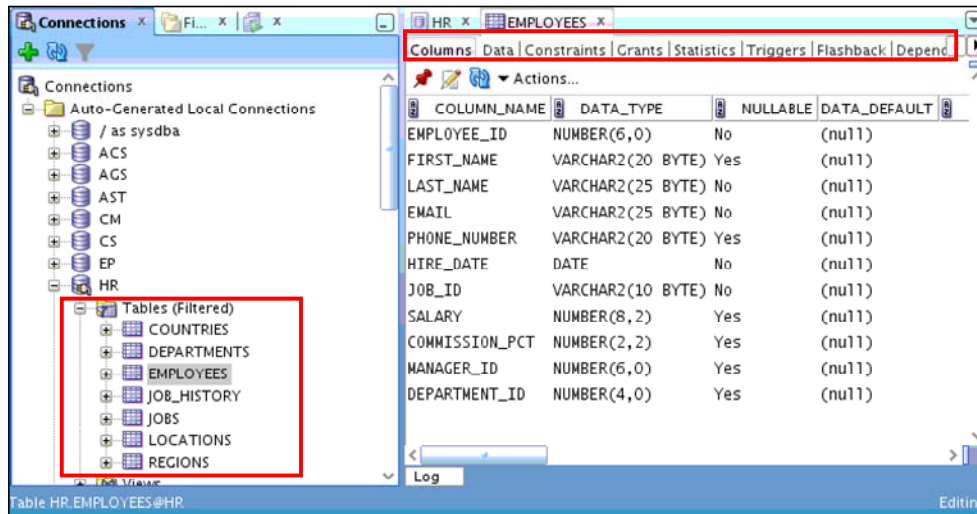
Note: From the same New/Select Database Connection window, You can also connect to schemas for selected third-party (non-Oracle) databases, such as MySQL, Microsoft SQL Server, Sybase Adaptive Server, Microsoft Access, and IBM DB2, and view metadata and data. However, these connections are read-only connections that enable you to browse objects and data in that data source.

Browsing Database Objects

Browsing Database Objects

Use the Connections Navigator to:

- Browse through many objects in a database schema
- Review the definitions of objects at a glance



Browsing Database Objects

After you create a database connection, you can use the Connections Navigator to browse through many objects in a database schema including Tables, Views, Indexes, Packages, Procedures, Triggers, and Types.

You can see the definition of the objects broken into tabs of information that is pulled out of the data dictionary. For example, if you select a table in the Navigator, the details about columns, constraints, grants, statistics, triggers, and so on are displayed on an easy-to-read tabbed page.

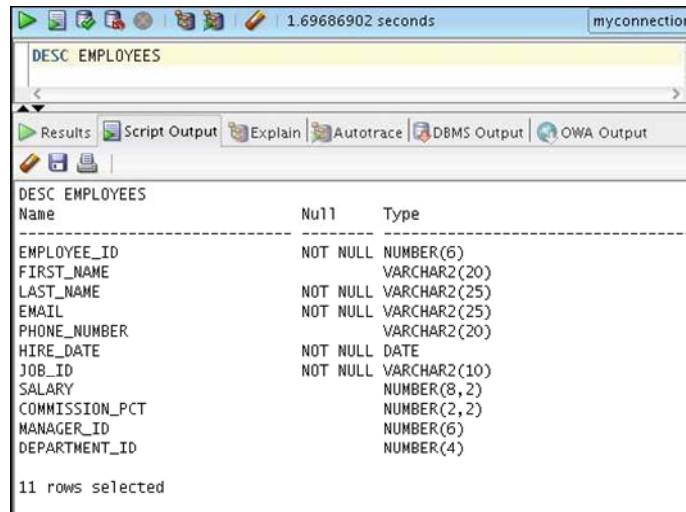
If you want to see the definition of the `EMPLOYEES` table as shown in the slide, perform the following steps:

1. Expand the Connections node in the Connections Navigator.
2. Expand Tables.
3. Click `EMPLOYEES`. By default, the Columns tab is selected. It shows the column description of the table. Using the Data tab, you can view the table data and also enter new rows, update data, and commit these changes to the database.

Displaying the Table Structure

Displaying the Table Structure

Use the `DESCRIBE` command to display the structure of a table:



ORACLE

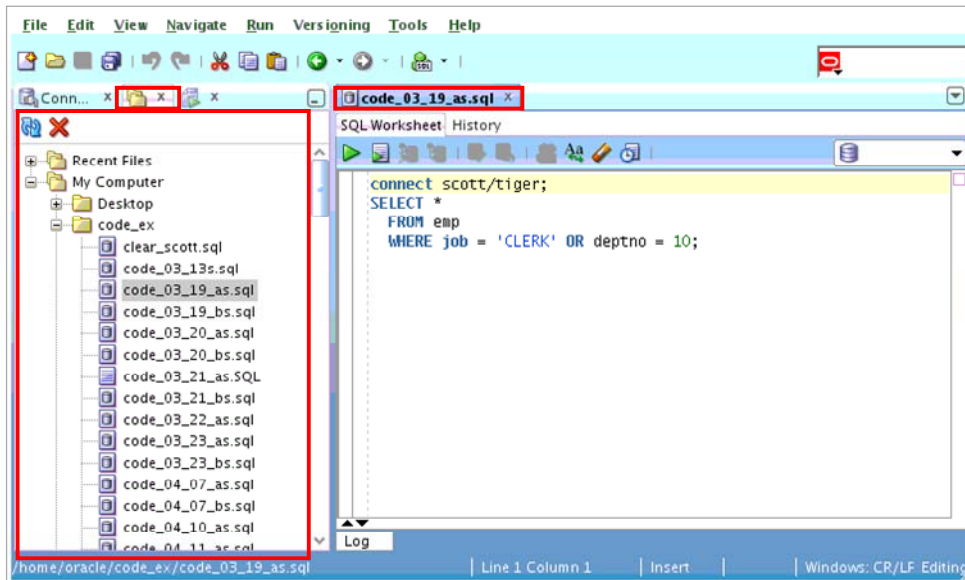
Displaying the Table Structure

In SQL Developer, you can also display the structure of a table using the `DESCRIBE` command. The result of the command is a display of column names and data types as well as an indication if a column must contain data.

Browsing Files

Browsing Files

Use the File Navigator to explore the file system and open system files.



Browsing Database Objects

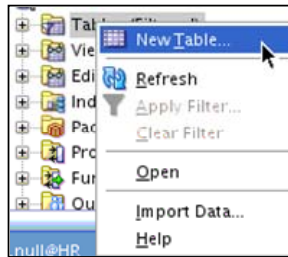
You can use the File Navigator to browse and open system files.

- To view the files navigator, click the Files tab, or select View > Files.
- To view the contents of a file, double-click a file name to display its contents in the SQL worksheet area.

Creating a Schema Object

Creating a Schema Object

- SQL Developer supports the creation of any schema object by:
 - Executing a SQL statement in SQL Worksheet
 - Using the context menu
- Edit the objects by using an edit dialog box or one of the many context-sensitive menus.
- View the data definition language (DDL) for adjustments such as creating a new object or editing an existing schema object.



ORACLE

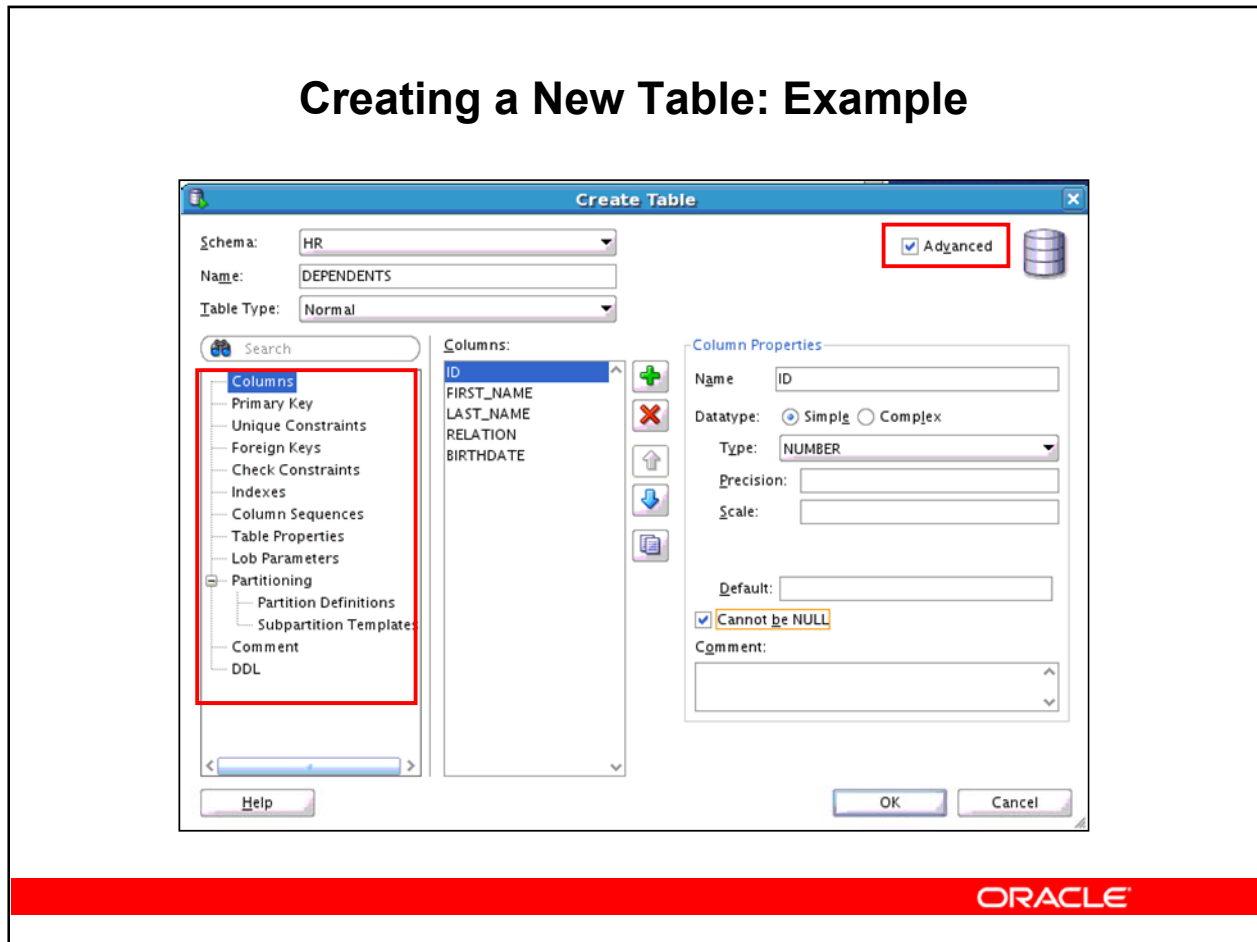
Creating a Schema Object

SQL Developer supports the creation of any schema object by executing a SQL statement in SQL Worksheet. Alternatively, you can create objects using the context menus. When created, you can edit the objects using an edit dialog box or one of the many context-sensitive menus.

As new objects are created or existing objects are edited, the DDL for those adjustments is available for review. An Export DDL option is available if you want to create the full DDL for one or more objects in the schema.

The slide shows how to create a table using the context menu. To open a dialog box for creating a new table, right-click Tables and select New Table. The dialog boxes to create and edit database objects have multiple tabs, each reflecting a logical grouping of properties for that type of object.

Creating a New Table: Example



Creating a New Table: Example

In the Create Table dialog box, if you do not select the Advanced check box, you can create a table quickly by specifying columns and some frequently used features.

If you select the Advanced check box, the Create Table dialog box changes to one with multiple options, in which you can specify an extended set of features while you create the table.

The example in the slide shows how to create the DEPENDENTS table by selecting the Advanced check box.

To create a new table, perform the following steps:

1. In the Connections Navigator, right-click Tables.
2. Select New TABLE.
3. In the Create Table dialog box, select Advanced.
4. Specify column information.
5. Click OK.

Although it is not required, you should also specify a primary key by using the Primary Key tab in the dialog box. Sometimes, you may want to edit the table that you have created; to do so, right-click the table in the Connections Navigator and select Edit.

Using the SQL Worksheet

Using the SQL Worksheet

- Use the SQL Worksheet to enter and execute SQL, PL/SQL, and SQL *Plus statements.
- Specify any actions that can be processed by the database connection associated with the worksheet.

ORACLE

Using the SQL Worksheet

When you connect to a database, a SQL Worksheet window for that connection automatically opens. You can use the SQL Worksheet to enter and execute SQL, PL/SQL, and SQL*Plus statements. The SQL Worksheet supports SQL*Plus statements to a certain extent. SQL*Plus statements that are not supported by the SQL Worksheet are ignored and not passed to the database.

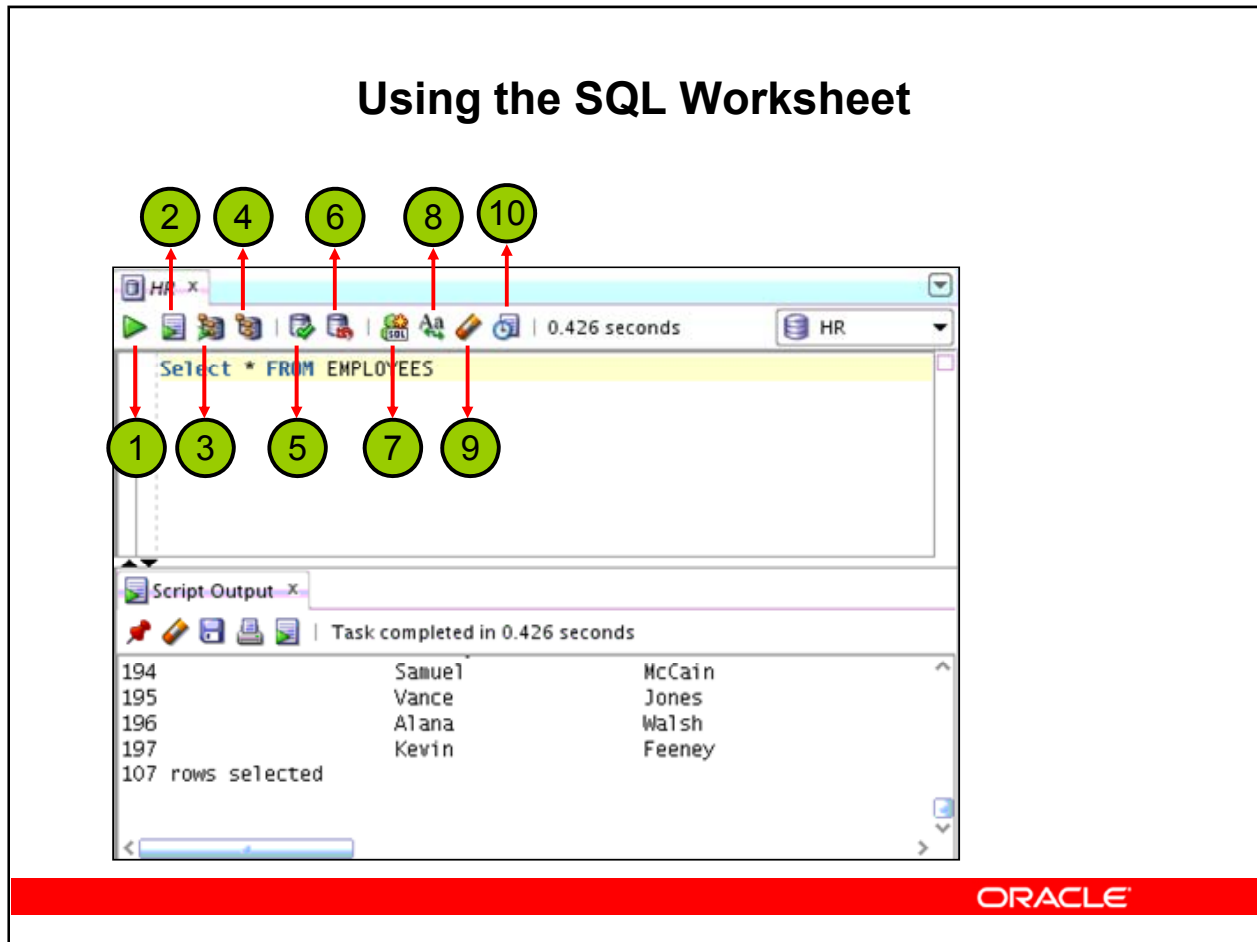
You can specify actions that can be processed by the database connection associated with the worksheet, such as:

- Creating a table
- Inserting data
- Creating and editing a trigger
- Selecting data from a table
- Saving the selected data to a file

You can display a SQL Worksheet by using one of the following:

- Select Tools > SQL Worksheet.
- Click the Open SQL Worksheet icon.

Using the SQL Worksheet



Using the SQL Worksheet (continued)

You may want to use the shortcut keys or icons to perform certain tasks such as executing a SQL statement, running a script, and viewing the history of SQL statements that you have executed. You can use the SQL Worksheet toolbar that contains icons to perform the following tasks:

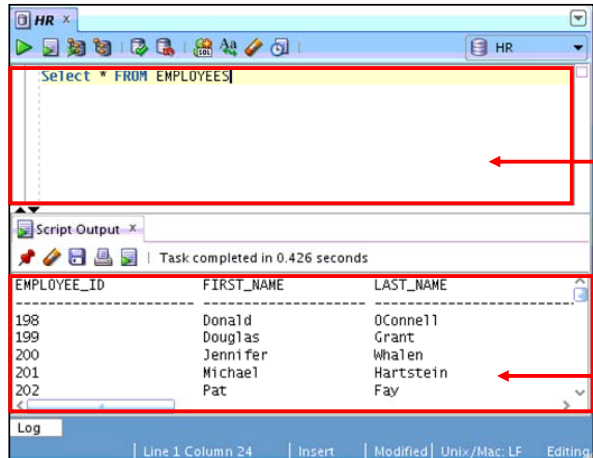
1. **Execute Statement:** Executes the statement where the cursor is located in the Enter SQL Statement box. You can use bind variables in the SQL statements, but not substitution variables.
2. **Run Script:** Executes all statements in the Enter SQL Statement box by using the Script Runner. You can use substitution variables in the SQL statements, but not bind variables.
3. **Autotrace:** Generates trace information for the statement
4. **Execute Explain Plan:** Generates the execution plan, which you can see by clicking the Explain tab
5. **Commit:** Writes any changes to the database and ends the transaction
6. **Rollback:** Discards any changes to the database, without writing them to the database, and ends the transaction
7. **Unshared SQL Worksheet:** Create a unshared SQL Worksheet

8. **Change Case:** Step through: To Uppercase, and Lower Case, and Initial Capitalization
9. **Clear:** Erases the statement or statements in the Enter SQL Statement box
10. **SQL History:** Displays a dialog box with information about SQL statements that you have executed

Using the SQL Worksheet

Using the SQL Worksheet

- Use the SQL Worksheet to enter and execute SQL, PL/SQL, and SQL*Plus statements.
- Specify any actions that can be processed by the database connection associated with the worksheet.



The screenshot shows the SQL Worksheet window with the following content:

```

Select * FROM EMPLOYEES
    
```

Below the editor, the Script Output window shows the execution results:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
198	Donald	O'Connell
199	Douglas	Grant
200	Jennifer	Whalen
201	Michael	Hartstein
202	Pat	Fay

Annotations in the image:

- A red box around the SQL editor with an arrow pointing to it and the text "Enter SQL statements."
- A red box around the Script Output window with an arrow pointing to it and the text "Results are shown here."

ORACLE

Using the SQL Worksheet (continued)

When you connect to a database, a SQL Worksheet window for that connection automatically opens. You can use the SQL Worksheet to enter and execute SQL, PL/SQL, and SQL*Plus statements. All SQL and PL/SQL commands are supported as they are passed directly from the SQL Worksheet to the Oracle database. SQL*Plus commands used in the SQL Developer have to be interpreted by the SQL Worksheet before being passed to the database.

The SQL Worksheet currently supports a number of SQL*Plus commands. Commands not supported by the SQL Worksheet are ignored and are not sent to the Oracle database. Through the SQL Worksheet, you can execute SQL statements and some of the SQL*Plus commands.

You can display a SQL Worksheet by using any of the following two options:

- Select Tools > SQL Worksheet.
- Click the Open SQL Worksheet icon.

Executing SQL Statements

Executing SQL Statements

Use the Enter SQL Statement box to enter single or multiple SQL statements.

Ctrl+Enter **F5**

SQL | Fetched 50 rows in 0.054 seconds

EMPLOYEE_ID	LAST_NAME
1	174 Abel
2	166 Ande
3	130 Atkinson
4	105 Austin

Task completed in 0.56 seconds

EMPLOYEE_ID	LAST_NAME
174	Abel
166	Ande
130	Atkinson
105	Austin
204	Baer
116	Baida

ORACLE

Executing SQL Statements

The example in the slide shows the difference in output for the same query when the Run Statement command (Ctrl+Enter key) is used versus the output when Run Script (F5 key) is used.

Saving SQL Scripts

Saving SQL Scripts

1 Click the Save icon to save your SQL statement to a file.

2 Identify a location, enter a file name, and click Save.

3 The contents of the saved file are visible and editable in your SQL Worksheet window.

ORACLE

Saving SQL Scripts

You can save your SQL statements from the SQL Worksheet into a text file. To save the contents of the Enter SQL Statement box, follow these steps:

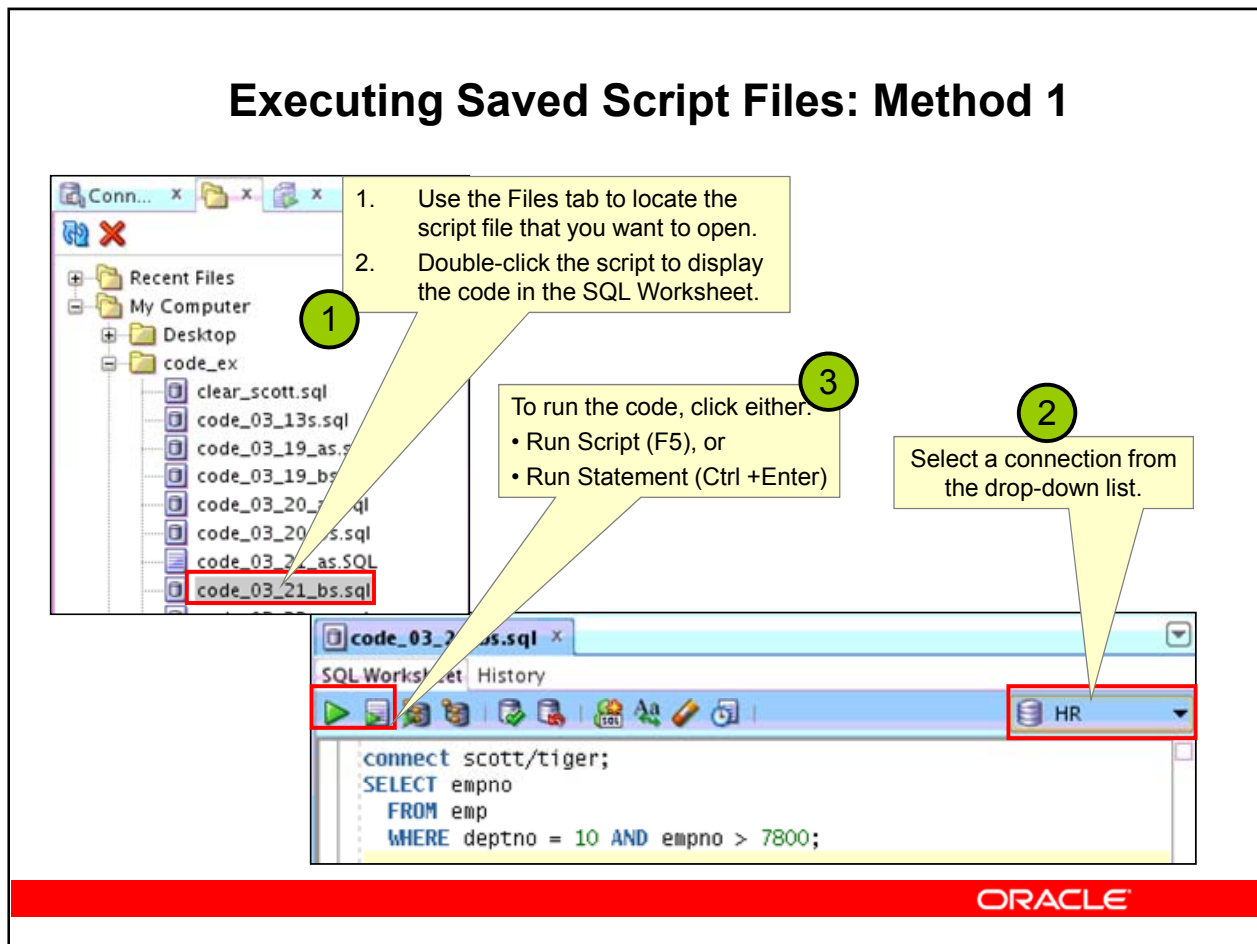
1. Click the Save icon or use the File > Save menu item.
2. In the Windows Save dialog box, enter a file name and the location where you want the file saved.
3. Click Save.

After you save the contents to a file, the Enter SQL Statement window displays a tabbed page of your file contents. You can have multiple files open at the same time. Each file displays as a tabbed page.

Script Pathing

You can select a default path to look for scripts and to save scripts. Under Tools > Preferences > Database > Worksheet Parameters, enter a value in the “Select default path to look for scripts” field.

Executing Saved Script Files: Method 1



Executing Saved Script Files: Method 1

To open a script file and display the code in the SQL Worksheet area, perform the following:

1. In the files navigator, select (or navigate to) the script file that you want to open.
2. Double-click to open. The code of the script file is displayed in the SQL Worksheet area.
3. Select a connection from the connection drop-down list.
4. To run the code, click the Run Script (F5) icon on the SQL Worksheet toolbar. If you have not selected a connection from the connection drop-down list, a connection dialog box will appear. Select the connection you want to use for the script execution.

Alternatively, you can also:

1. Select File > Open. The Open dialog box is displayed.
2. In the Open dialog box, select (or navigate to) the script file that you want to open.
3. Click Open. The code of the script file is displayed in the SQL Worksheet area.
4. Select a connection from the connection drop-down list.
5. To run the code, click the Run Script (F5) icon on the SQL Worksheet toolbar. If you have not selected a connection from the connection drop-down list, a connection dialog box will appear. Select the connection you want to use for the script execution.

Executing Saved Script Files: Method 2

Executing Saved Script Files: Method 2

Use the @ command followed by the location and name of the file you want to execute, and click the Run Script icon.

The output from the script is displayed on the Script Output tabbed page.

Executing Saved Script Files: Method 2

To run a saved SQL script, perform the following:

1. Use the @ command, followed by the location, and name of the file you want to run, in the Enter SQL Statement window.
2. Click the Run Script icon.

The results from running the file are displayed on the Script Output tabbed page. You can also save the script output by clicking the Save icon on the Script Output tabbed page. The Windows Save dialog box appears and you can identify a name and location for your file.

Formatting the SQL Code

Formatting the SQL Code

Before
formatting

```
select select employee_id, first_name, last_name from employees e, departments d
where e.department_id = d.department_id and e.salary > 3000;
```

After
formatting

```
SELECT
SELECT employee_id,
      first_name,
      last_name
FROM employees e,
      departments d
WHERE e.department_id = d.department_id
AND e.salary > 3000;
```

Run Statement	F9
Run Script	F5
Create Report...	
Save as Snippet...	
Autotrace...	F6
Explain Plan...	F10
Commit	F11
Rollback	F12
To Upper/Lower/InitCap	Ctrl+Quote
Clear	Ctrl-D
History	F8
Refactoring	
Query Builder	
Format	Ctrl-F7
Advanced Format...	Ctrl+Shift-F7
Cut	Ctrl-X
Copy	Ctrl-C
Paste	Ctrl-V
Select All	Ctrl-A

ORACLE

Formatting the SQL Code

You may want to beautify the indentation, spacing, capitalization, and line separation of the SQL code. SQL Developer has a feature for formatting SQL code.

To format the SQL code, right-click in the statement area and select Format SQL.

In the example in the slide, before formatting, the SQL code has the keywords not capitalized and the statement not properly indented. After formatting, the SQL code is beautified with the keywords capitalized and the statement properly indented.

Using Snippets

Using Snippets

Snippets are code fragments that may be just syntax or examples.

From the drop-down list, you can select the functions category that you want.

Using Snippets

You may want to use certain code fragments when you use the SQL Worksheet or create or edit a PL/SQL function or procedure. SQL Developer has the feature called Snippets. Snippets are code fragments such as SQL functions, Optimizer hints, and miscellaneous PL/SQL programming techniques. You can drag snippets into the Editor window.

To display Snippets, select View > Snippets.

The Snippets window is displayed at the right side. You can use the drop-down list to select a group.

ORACLE

Using Snippets: Example

Using Snippets: Example

The image contains two screenshots of the SQL Developer interface. The top screenshot shows a snippet being inserted into a SQL statement. A green box labeled 'Inserting a snippet' has an arrow pointing to the 'CONCAT(char1, char2)' snippet in the Snippets window. The SQL editor shows the snippet being pasted into a statement: `SELECT CONCAT(char1, char2) FROM table`. The bottom screenshot shows the snippet being edited. A green box labeled 'Editing the snippet' has an arrow pointing to the 'CONCAT(char1, char2)' snippet in the Snippets window. The SQL editor shows the snippet being edited to: `select concat(first_name, last_name) FROM EMPLOYEES;`. The Oracle logo is visible at the bottom of the slide.

Using Snippets: Example

To insert a Snippet into your code in a SQL Worksheet or in a PL/SQL function or procedure, drag the snippet from the Snippets window into the desired place in your code. Then you can edit the syntax so that the SQL function is valid in the current context. To see a brief description of a SQL function in a tool tip, place the cursor over the function name.

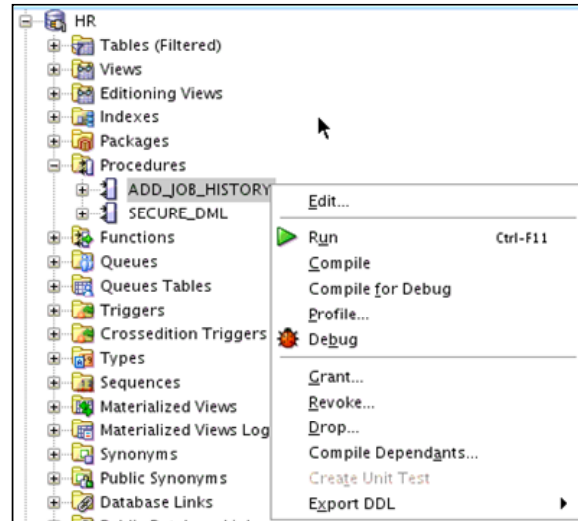
The example in the slide shows that `CONCAT(char1, char2)` is dragged from the Character Functions group in the Snippets window. Then the `CONCAT` function syntax is edited and the rest of the statement is added as in the following:

```
SELECT CONCAT(first_name, last_name)
FROM employees;
```

Debugging Procedures and Functions

Debugging Procedures and Functions

- Use SQL Developer to debug PL/SQL functions and procedures.
- Use the “Compile for Debug” option to perform a PL/SQL compilation so that the procedure can be debugged.
- Use the Debug menu options to set breakpoints, and to perform the step into and step over tasks.



ORACLE

Debugging Procedures and Functions

In SQL Developer, you can debug PL/SQL procedures and functions. Using the Debug menu options, you can perform the following debugging tasks:

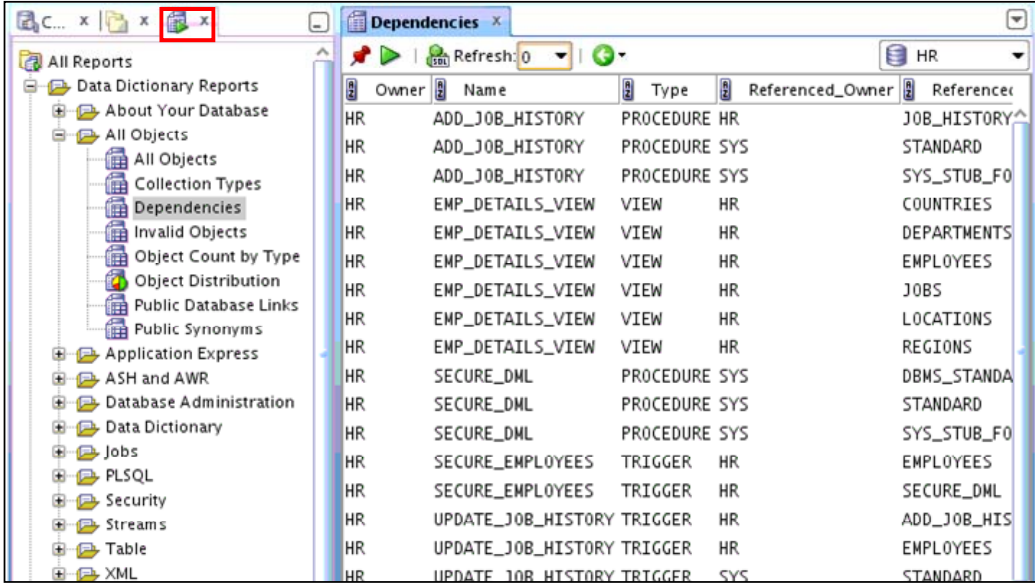
- **Find Execution Point** goes to the next execution point.
- **Resume** continues execution.
- **Step Over** bypasses the next method and goes to the next statement after the method.
- **Step Into** goes to the first statement in the next method.
- **Step Out** leaves the current method and goes to the next statement.
- **Step to End of Method** goes to the last statement of the current method.
- **Pause** halts execution but does not exit, thus allowing you to resume execution.
- **Terminate** halts and exits the execution. You cannot resume execution from this point; instead, to start running or debugging from the beginning of the function or procedure, click the Run or Debug icon on the Source tab toolbar.
- **Garbage Collection** removes invalid objects from the cache in favor of more frequently accessed and more valid objects.

These options are also available as icons on the debugging toolbar.

Database Reporting

Database Reporting

SQL Developer provides a number of predefined reports about the database and its objects.



Owner	Name	Type	Referenced_Owner	Referenced
HR	ADD_JOB_HISTORY	PROCEDURE	HR	JOB_HISTORY
HR	ADD_JOB_HISTORY	PROCEDURE	SYS	STANDARD
HR	ADD_JOB_HISTORY	PROCEDURE	SYS	SYS_STUB_F0
HR	EMP_DETAILS_VIEW	VIEW	HR	COUNTRIES
HR	EMP_DETAILS_VIEW	VIEW	HR	DEPARTMENTS
HR	EMP_DETAILS_VIEW	VIEW	HR	EMPLOYEES
HR	EMP_DETAILS_VIEW	VIEW	HR	JOBS
HR	EMP_DETAILS_VIEW	VIEW	HR	LOCATIONS
HR	EMP_DETAILS_VIEW	VIEW	HR	REGIONS
HR	SECURE_DML	PROCEDURE	SYS	DBMS_STANDA
HR	SECURE_DML	PROCEDURE	SYS	STANDARD
HR	SECURE_DML	PROCEDURE	SYS	SYS_STUB_F0
HR	SECURE_EMPLOYEES	TRIGGER	HR	EMPLOYEES
HR	SECURE_EMPLOYEES	TRIGGER	HR	SECURE_DML
HR	UPDATE_JOB_HISTORY	TRIGGER	HR	ADD_JOB_HIS
HR	UPDATE_JOB_HISTORY	TRIGGER	HR	EMPLOYEES
HR	UPDATE_JOB_HISTORY	TRIGGER	SYS	STANDARD

ORACLE

Database Reporting

SQL Developer provides many reports about the database and its objects. These reports can be grouped into the following categories:

- About Your Database reports
- Database Administration reports
- Table reports
- PL/SQL reports
- Security reports
- XML reports
- Jobs reports
- Streams reports
- All Objects reports
- Data Dictionary reports
- User-Defined reports

To display reports, click the Reports tab at the left side of the window. Individual reports are displayed in tabbed panes at the right side of the window; and for each report, you can select (using a drop-down list) the database connection for which to display the report. For reports

about objects, the objects shown are only those visible to the database user associated with the selected database connection, and the rows are usually ordered by Owner. You can also create your own user-defined reports.

Creating a User-Defined Report

Creating a User-Defined Report

Create and save user-defined reports for repeated use.

Organize reports in folders.

ORACLE

Creating a User-Defined Report

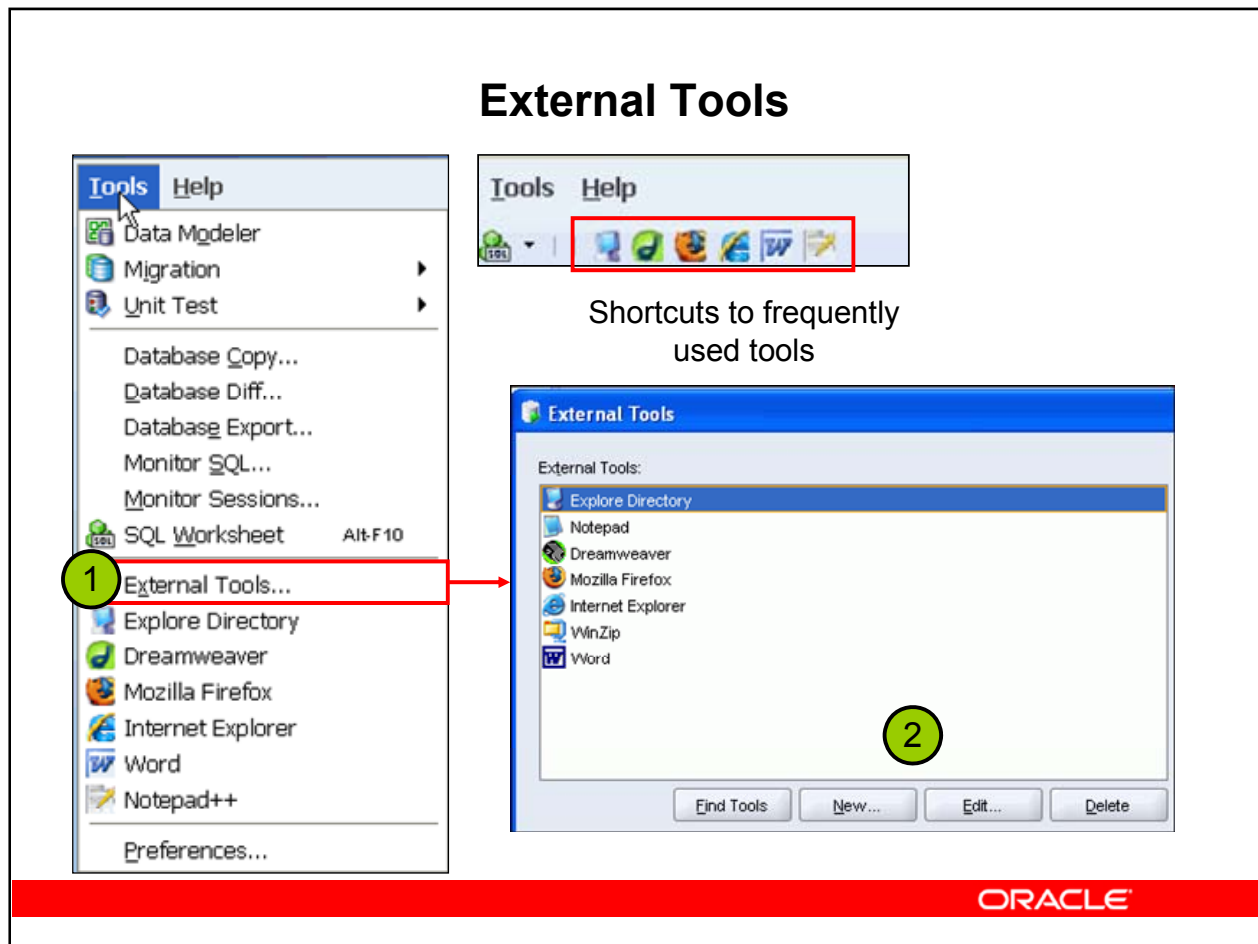
User-defined reports are reports created by SQL Developer users. To create a user-defined report, perform the following steps:

1. Right-click the User Defined Reports node under Reports, and select Add Report.
2. In the Create Report dialog box, specify the report name and the SQL query to retrieve information for the report. Then click Apply.

In the example in the slide, the report name is specified as `emp_sal`. An optional description is provided indicating that the report contains details of employees with `salary >= 10000`. The complete SQL statement for retrieving the information to be displayed in the user-defined report is specified in the SQL box. You can also include an optional tool tip to be displayed when the cursor stays briefly over the report name in the Reports navigator display.

You can organize user-defined reports in folders, and you can create a hierarchy of folders and subfolders. To create a folder for user-defined reports, right-click the User Defined Reports node or any folder name under that node and select Add Folder. Information about user-defined reports, including any folders for these reports, is stored in a file named `UserReports.xml` under the directory for user-specific information.

External Tools



Search Engines and External Tools

To enhance productivity of the SQL developers, SQL Developer allows you to add shortcut icons to some of the frequently used tools such as Notepad, Microsoft Word, and Dreamweaver, available to you.

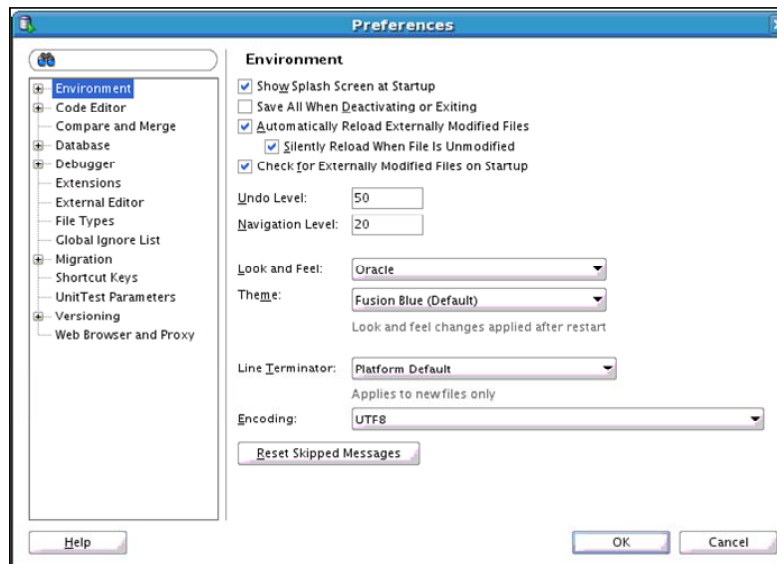
You can add external tools to the existing list or even delete shortcuts to tools that you do not use frequently. To do so, perform the following:

1. From the Tools menu, select External Tools.
2. In the External Tools dialog box, perform the following:
 - A. Click New to invoke the wizard to add new tools.
 - B. Click Delete to remove any tool from the list.
 - C. Click Edit to invoke the wizard to modify the availability and parameters of the selected tool.

Setting Preferences

Setting Preferences

- Customize the SQL Developer interface and environment.
- In the Tools menu, select Preferences.



ORACLE

Setting Preferences

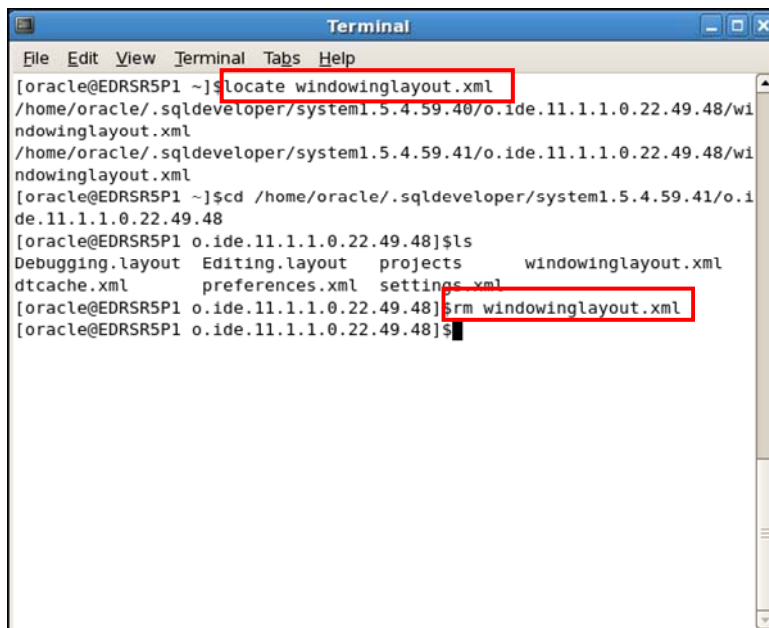
You can customize many aspects of the SQL Developer interface and environment by modifying SQL Developer preferences according to your preferences and needs. To modify SQL Developer preferences, select Tools, then Preferences.

The preferences are grouped into the following categories:

- Environment
- Accelerators (keyboard shortcuts)
- Code Editors
- Database
- Debugger
- Documentation
- Extensions
- File Types
- Migration
- PL/SQL Compilers
- PL/SQL Debugger, and so on

Resetting the SQL Developer Layout

Resetting the SQL Developer Layout



```
Terminal
File Edit View Terminal Tabs Help
[oracle@EDRSR5P1 ~]$ locate windowinglayout.xml
/home/oracle/.sqldeveloper/system1.5.4.59.40/o.ide.11.1.1.0.22.49.48/wi
ndowinglayout.xml
/home/oracle/.sqldeveloper/system1.5.4.59.41/o.ide.11.1.1.0.22.49.48/wi
ndowinglayout.xml
[oracle@EDRSR5P1 ~]$ cd /home/oracle/.sqldeveloper/system1.5.4.59.41/o.i
de.11.1.1.0.22.49.48
[oracle@EDRSR5P1 o.ide.11.1.1.0.22.49.48]$ ls
Debugging.layout  Editing.layout  projects        windowinglayout.xml
dtcache.xml      preferences.xml settings.xml
[oracle@EDRSR5P1 o.ide.11.1.1.0.22.49.48]$ rm windowinglayout.xml
[oracle@EDRSR5P1 o.ide.11.1.1.0.22.49.48]$
```

ORACLE

Resetting the SQL Developer Layout

While working with SQL Developer, if the Connections Navigator disappears or if you cannot dock the Log window in its original place, perform the following steps to fix the problem:

1. Exit from SQL Developer.
2. Open a terminal window and use the `locate` command to find the location of `windowinglayout.xml`.
3. Go to the directory which has `windowinglayout.xml` and delete it.
4. Restart SQL Developer.

Summary

Summary

In this appendix, you should have learned how to use SQL Developer to do the following:

- Browse, create, and edit database objects
- Execute SQL statements and scripts in SQL Worksheet
- Create and save custom reports

ORACLE

Summary

SQL Developer is a free graphical tool to simplify database development tasks. Using SQL Developer, you can browse, create, and edit database objects. You can use SQL Worksheet to run SQL statements and scripts. SQL Developer enables you to create and save your own special set of reports for repeated use.