

Using SQL Developer

Copyright © 2009, Oracle. All rights reserved.

ORACLE

PATITAPABAN PARIDA (pppparida9@gmail.com) has a
non-transferable license to use this Student Guide.

Objectives

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

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

ORACLE

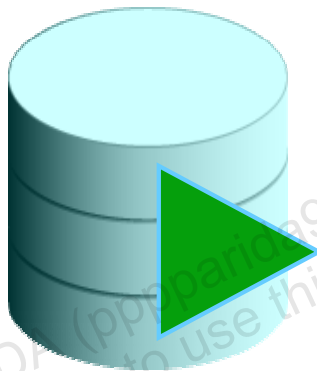
Copyright © 2009, Oracle. All rights reserved.

Objectives

In this appendix, you are introduced to the graphical tool 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?

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

**ORACLE**

Copyright © 2009, Oracle. All rights reserved.

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 using standard Oracle Database authentication. When connected, you can perform operations on objects in the database.

Key Features

- Developed in Java
- Supports Windows, Linux, and Mac OS X platforms
- Default connectivity by using the JDBC Thin driver
- Does not require an installer
- Connects to any Oracle Database version 9.2.0.1 and later
- Bundled with JRE 1.5

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2009, Oracle. All rights reserved.

Key Features of SQL Developer

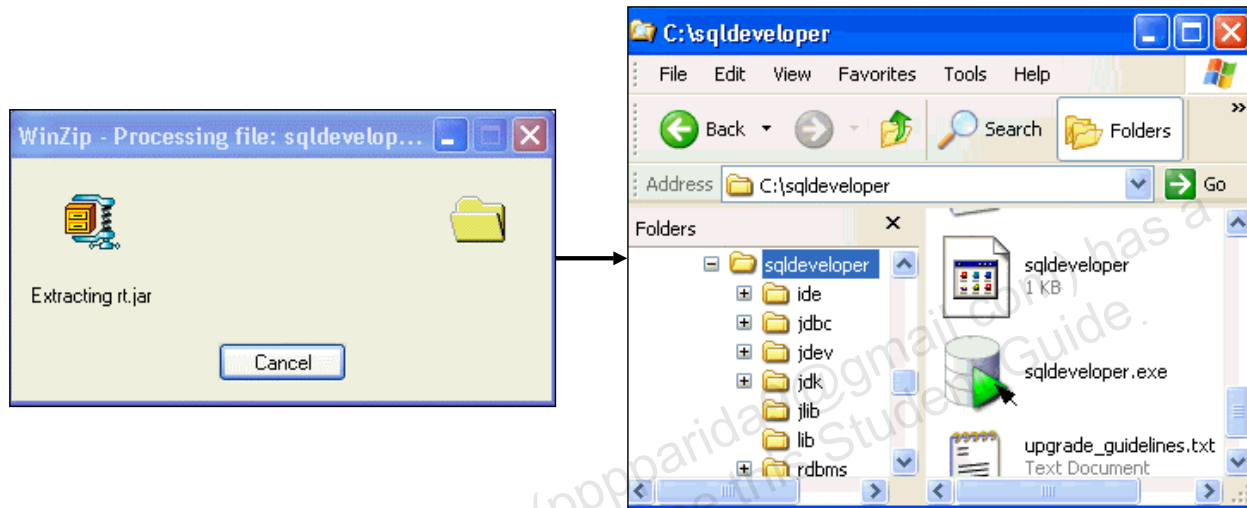
Oracle SQL Developer is developed in Java, leveraging the Oracle JDeveloper IDE. The tool runs on the Windows, Linux, and Mac OS X platforms. You can install SQL Developer on the Database Server and connect remotely from your desktop, thus avoiding client server network traffic.

Default connectivity to the database is through the JDBC Thin driver; so, no Oracle Home is required. SQL Developer does not require an installer and you need to just unzip the downloaded file.

With SQL Developer, users can connect to Oracle Database 9.2.0.1 and later versions, and all Oracle Database editions including Express Edition. SQL Developer is bundled with JRE 1.5, with an additional `tools.jar` to support Windows clients. Non-Windows clients need only JDK 1.5.

Installing SQL Developer

Download the Oracle SQL Developer kit and unzip it into any directory on your machine.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Installing SQL Developer

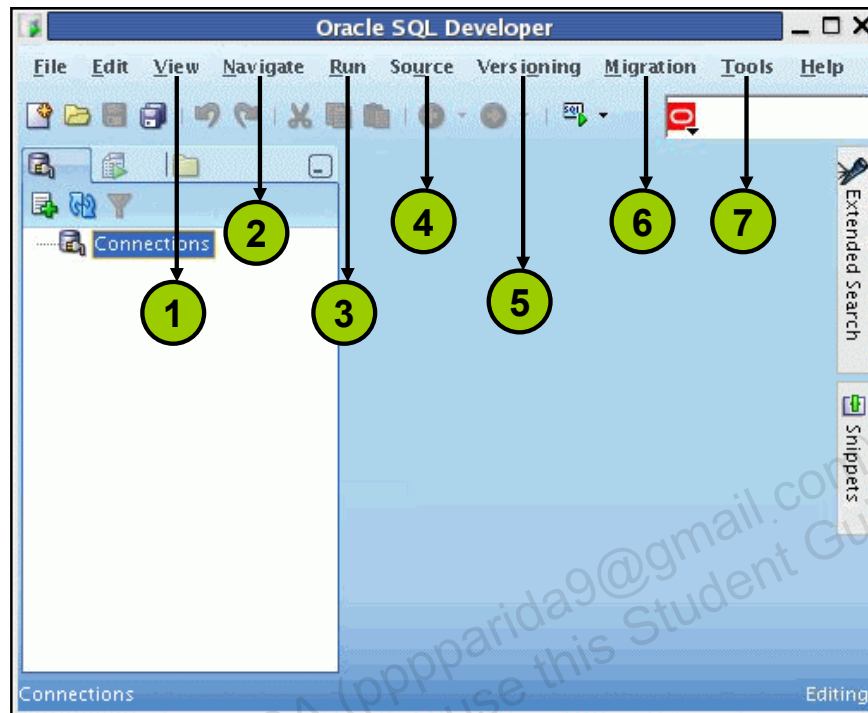
Oracle SQL Developer does not require an installer. To install SQL Developer, you need an unzip tool.

To install SQL Developer, perform the following steps:

1. Create a folder as <local drive>:\SQL Developer.
2. Download the SQL Developer kit from <http://www.oracle.com/technology/software/products/sql/index.html>.
3. Unzip the downloaded SQL Developer kit into the folder created in step 1.

To start SQL Developer, go to <local drive>:\SQL Developer, and double-click `sqldeveloper.exe`.

Menus for SQL Developer



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Menus for SQL Developer

SQL Developer has two main navigation tabs.

- **Connections Navigator:** By using this tab, you can browse database objects and users to which you have access.
- **Reporting Tab:** By using this tab, you can run predefined reports or create and add your own reports.

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. The menus at the top contain standard entries, plus entries for features specific to SQL Developer.

1. **View:** Contains options that affect what is displayed in the SQL Developer interface
2. **Navigate:** Contains options for navigating to panes and for the execution of subprograms
3. **Run:** Contains the Run File and Execution Profile options that are relevant when a function or procedure is selected
4. **Source:** Contains options for use when editing functions and procedures
5. **Versioning:** Enables you to work with the files placed under source code control
6. **Migration:** Enables you to migrate from another database, such as Microsoft SQL Server and Microsoft Access, to an Oracle Database
7. **Tools:** Invokes SQL Worksheet, Preferences, and any added External Tools

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
 - For multiple schemas
- SQL Developer automatically imports any connections defined in the `tnsnames.ora` file on your system.
- You can export connections to an XML file.
- Each additional database connection created is listed in the connections navigator hierarchy.

ORACLE

Copyright © 2009, Oracle. All rights reserved.

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.

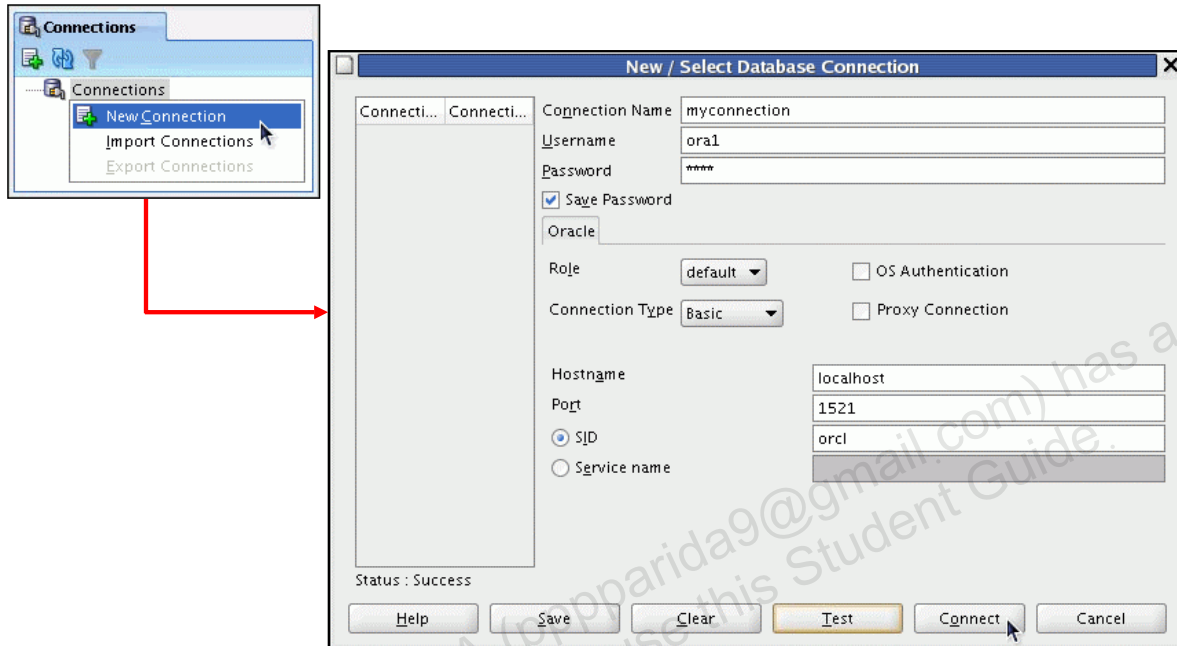
By default, the `tnsnames.ora` file is located in the `$ORACLE_HOME/network/admin` directory. But, it can also be in the directory specified by the `TNS_ADMIN` environment variable or registry value. When you start SQL Developer and display the database connections dialog box, SQL Developer automatically imports any connections defined in the `tnsnames.ora` file on your system.

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

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

You can create additional connections to connect to the same database but as different users, or to connect to different databases. Each database connection is listed in the Connections navigator hierarchy.

Creating a Database Connection



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a Database Connection (continued)

To create a database connection, perform the following steps:

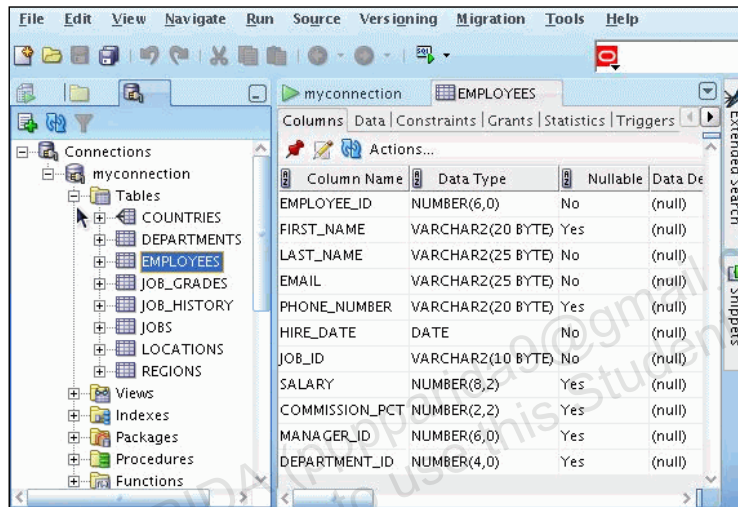
1. Double-click `<your_path>\sqldeveloper\sqldeveloper.exe`.
2. On the **Connections** tabbed page, right-click **Connections** and select **New Connection**.
3. Enter the connection name, username, password, host name, port number, and SID for the database you want to connect.
4. Click **Test** to make sure that the connection has been set correctly.
5. Click **Connect**.

Note: 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 will not be prompted for the password.

Browsing Database Objects

Use the Database Navigator to:

- Browse through many objects in a database schema
- Do a quick review of the definitions of objects



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Browsing Database Objects

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

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

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 more are all displayed in an easy-to-read tabbed page.

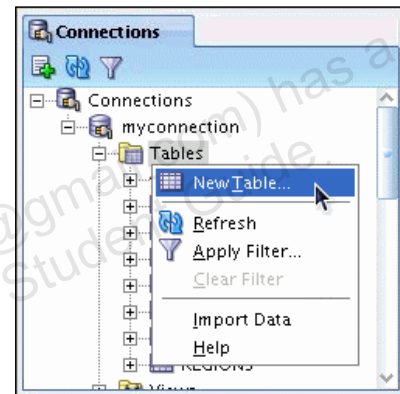
If you want to see the definition of the `EMPLOYEES` table as shown on the slide, perform the following steps:

1. Expand the connection node in the Connections Navigator.
2. Expand **Tables**.
3. Double-click **EMPLOYEES**.

Using the Data tab, you can enter new rows, update data, and commit these changes to the database.

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 many context-sensitive menus.
- View the DDL for adjustments, such as creating a new object or editing an existing schema object.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

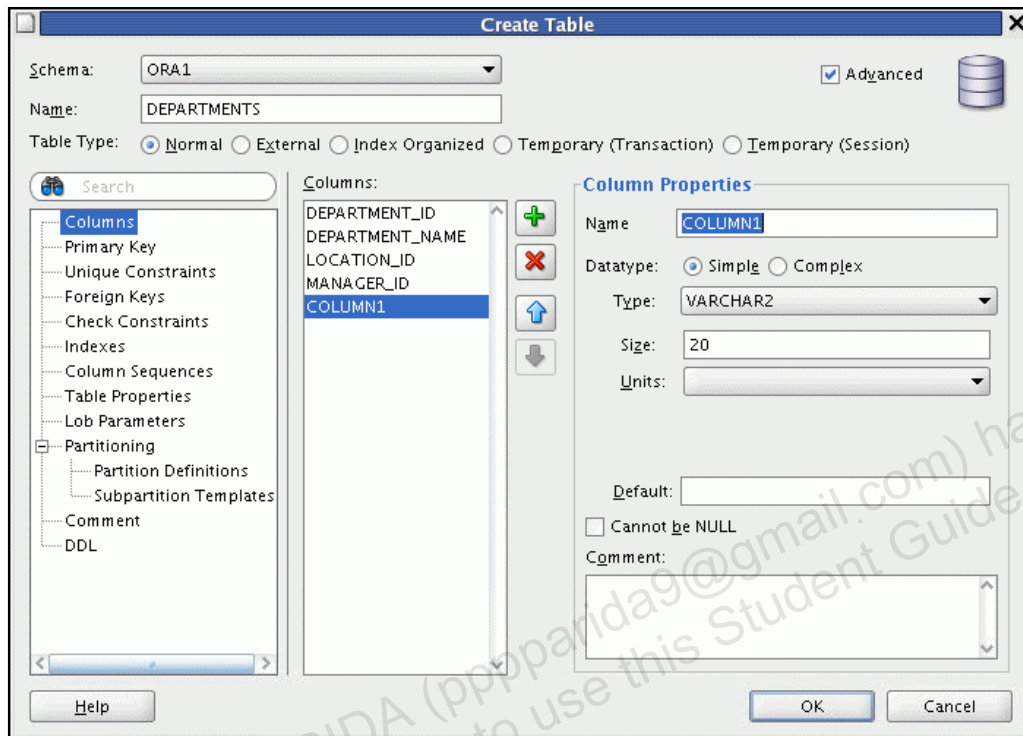
Creating a Schema Object

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. Once created, you can edit the objects using an edit dialog or one of 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 creating 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 for creating and editing database objects have multiple tabs, each reflecting a logical grouping of properties for that type of object.

Creating a New Table: Example



ORACLE

Copyright © 2009, Oracle. All rights reserved.

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 tabs, in which you can specify an extended set of features while creating the table.

The example in the slide shows creating the DEPENDENT table by selecting the **Advanced** check box.

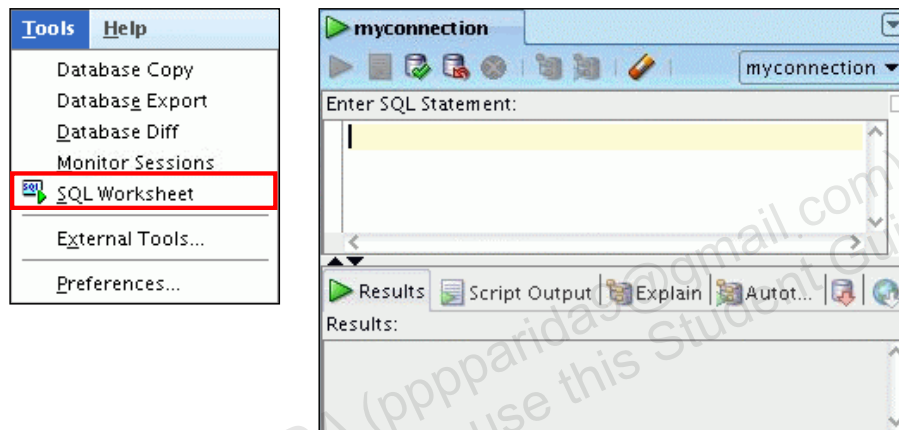
To create a new table, perform the following steps:

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 selecting Primary Key in the dialog box. Sometimes, you may want to edit the table that you have created. To edit a table, right-click the table in the Connections Navigator and select **Edit**.

Using SQL Worksheet

- Use 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

Copyright © 2009, Oracle. All rights reserved.

Using SQL Worksheet

When you connect to a database, a SQL Worksheet window for that connection is automatically opened. You can use 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.

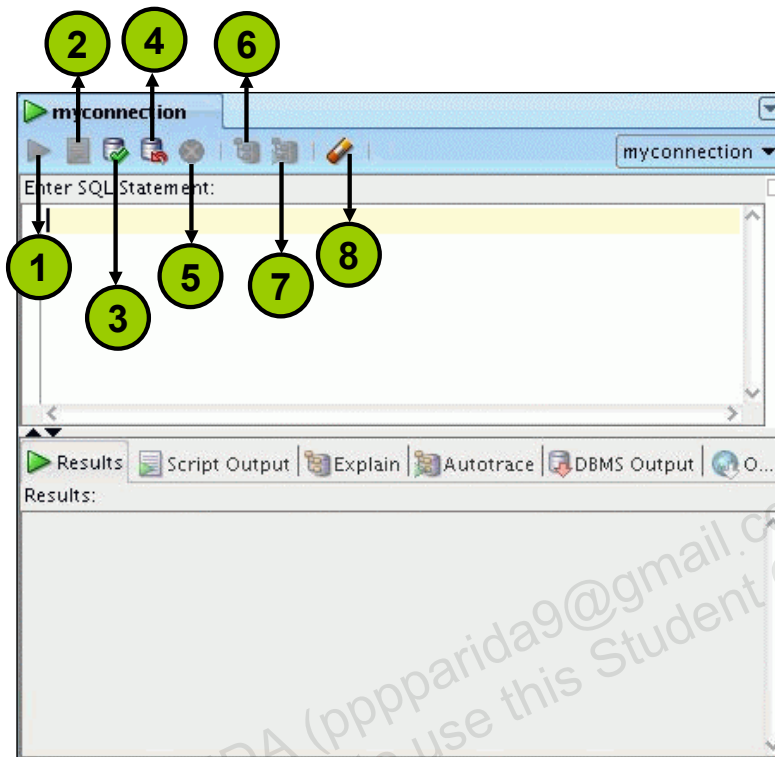
You can specify any actions that can be processed by the database connection associated with the worksheet, such as:

- 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 any of the following two options:

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

Using SQL Worksheet



ORACLE

Copyright © 2009, Oracle. All rights reserved.

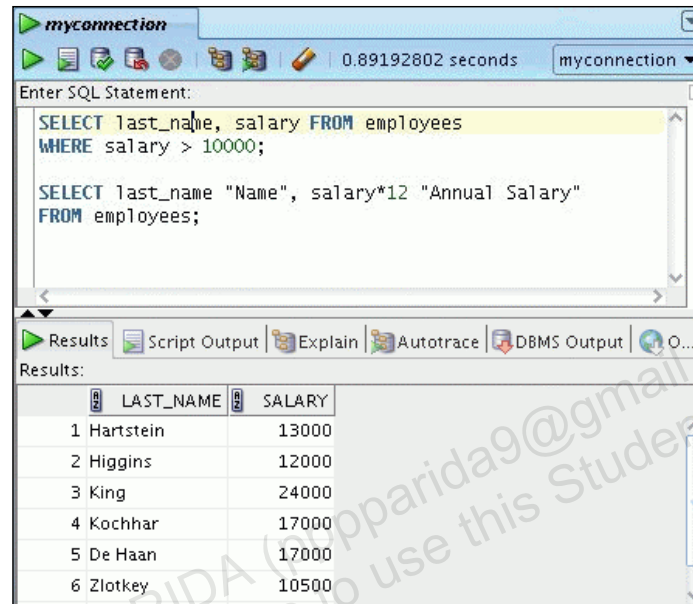
Using SQL Worksheet (continued)

You may want to use 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 at the cursor 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. **Commit:** Writes any changes to the database, and ends the transaction
4. **Rollback:** Discards any changes to the database, without writing them to the database, and ends the transaction
5. **Cancel:** Stops the execution of any statements currently being executed
6. **Execute Explain Plan:** Generates the execution plan, which you can see by clicking the Explain tab
7. **Autotrace:** Generates trace information for the statement, which you can see by clicking the Autotrace tab
8. **Clear:** Erases the statement or statements in the Enter SQL Statement box

Executing SQL Statements

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



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Executing SQL Statements

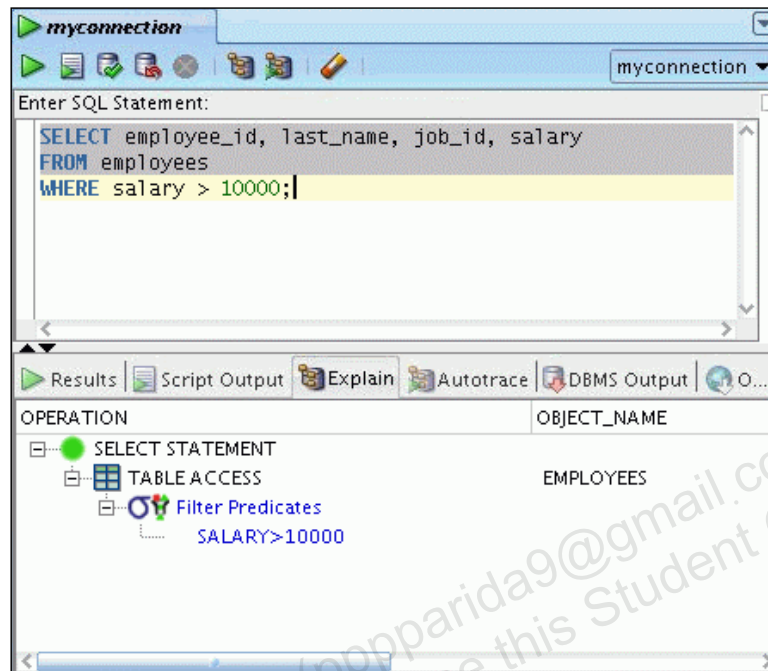
In SQL Worksheet, you can use the Enter SQL Statement box to enter a single or multiple SQL statements. For a single statement, the semicolon at the end is optional.

When you type in the statement, the SQL keywords are automatically highlighted. To execute a SQL statement, ensure that your cursor is within the statement and click the **Execute Statement** icon. Alternatively, you can press the **F9** key.

To execute multiple SQL statements and see the results, click the **Run Script** icon. Alternatively, you can press the **F5** key.

In the example in the slide, as there are multiple SQL statements, the first statement is terminated with a semicolon. The cursor is in the first statement and, therefore, when the statement is executed, results corresponding to the first statement are displayed in the Results box.

Viewing the Execution Plan



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Viewing the Execution Plan

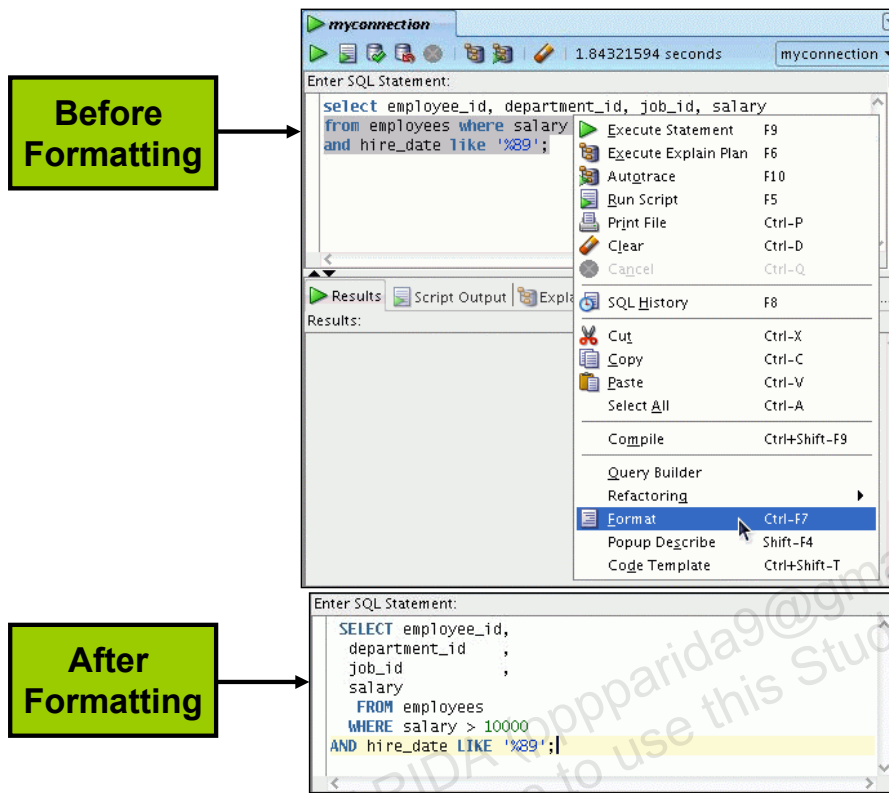
You can execute a SQL script and view the execution plan. To execute a SQL script file, perform the following steps:

1. From the **File** menu, Select **Open**.
2. In the Open dialog box, double-click the **.sql** file.
3. Click the **Run Script** icon.

When you double-click the **.sql** file, the SQL statements are loaded into the Enter SQL Statement box. You can execute the script or each line individually. The results are displayed in the Script Output area.

The example in the slide shows the execution plan. The Execute Explain Plan icon generates the execution plan. An execution plan is the sequence of operations that are performed to execute the statement. You can see the execution plan by clicking the **Explain** tab.

Formatting the SQL Code



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Formatting the SQL Code

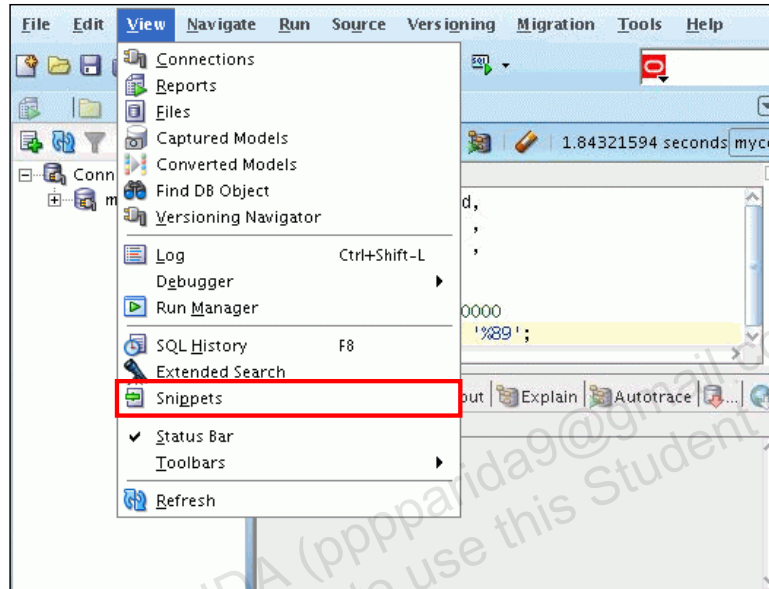
You may want to beautify the indentation, spacing, capitalization, and line separation of the SQL code. SQL Developer has the feature of formatting the SQL code.

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

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

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



ORACLE

Copyright © 2009, Oracle. All rights reserved.

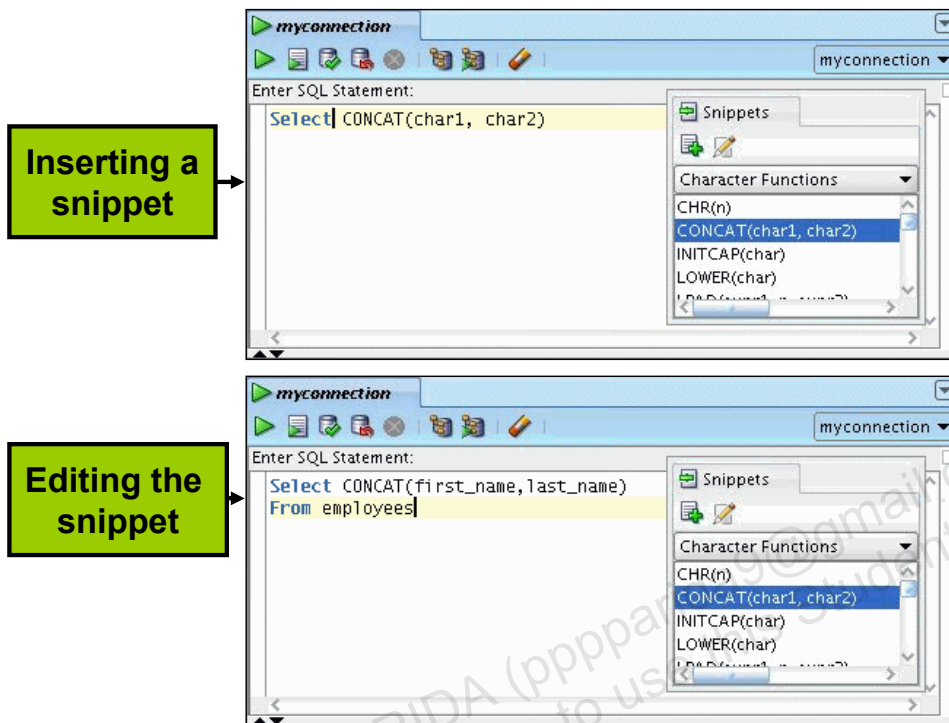
Using Snippets

You may want to use certain code fragments when you are using the SQL Worksheet or creating or editing 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 and drop snippets into the editor window.

To display Snippets, select **View > Snippets**.

The Snippets window is displayed on the right. You can use the drop-down list to select a group. A Snippets button is placed in the right window margin, so that you can display the Snippets window if it becomes hidden.

Using Snippets: Example



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Using Snippets: Example

To insert a snippet into your code in SQL Worksheet or in a PL/SQL function or procedure, drag the snippet from the Snippets window to 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 such as in the following:

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

Using SQL*Plus

- SQL Worksheet does not support all SQL*Plus statements.
- SQL*Plus statements that are not supported by SQL Worksheet are:
 - append
 - archive
 - attribute
 - break
 - change
 - clear

ORACLE

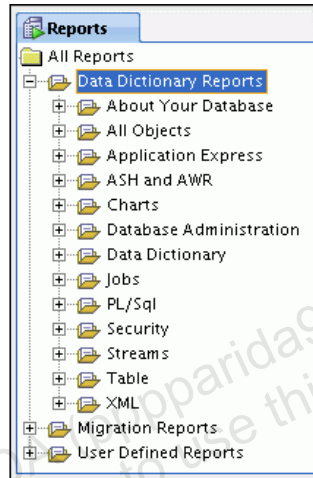
Copyright © 2009, Oracle. All rights reserved.

Using SQL*Plus

SQL Worksheet supports some SQL*Plus statements. SQL*Plus statements must be interpreted by the SQL Worksheet before being passed to the database; any SQL*Plus statements that are not supported by the SQL Worksheet are ignored and not passed to the database. For example, some of the SQL*Plus statements that are not supported by SQL Worksheet are listed in the slide. For the complete list of SQL*Plus statements that are supported and not supported by SQL Worksheet, refer to SQL Developer online Help.

Database Reporting

- SQL Developer provides you with a number of predefined reports about your database and objects.
- The Reports are organized into categories.
- You can create your own customized reports too.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Database Reporting

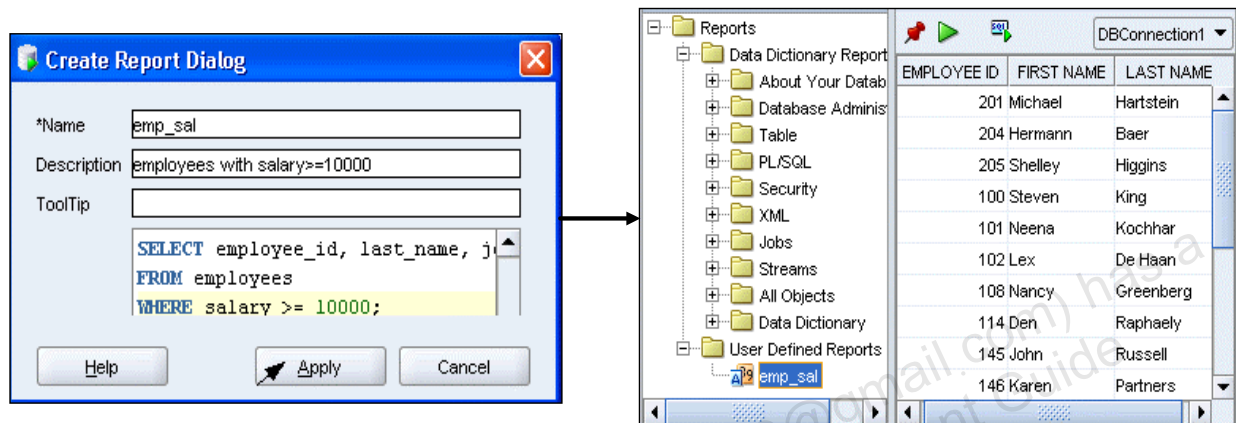
SQL Developer provides many reports about the database and its objects. These reports are grouped into the following categories:

- About Your Database reports
- Object reports
- Application Express reports
- Charts
- Database Administration reports
- Data Dictionary reports
- Jobs reports
- PL/SQL reports
- Security reports
- Streams reports
- Table reports
- XML reports

To display a report, click the Reports tabbed page and then select the report type. You can also create your own user-defined reports.

Creating a User-Defined Report

Create and save user-defined reports for repeated use.



ORACLE

Copyright © 2009, Oracle. All rights reserved.

Creating a User-Defined Report

User-defined reports are any reports that are 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.

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

Copyright © 2009, Oracle. All rights reserved.

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.