

Using SQL Developer

ORACLE

Copyright © 2006, Oracle. All rights reserved.

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 the SQL Worksheet**
- **Execute SQL statements and SQL scripts**
- **Edit and debug PL/SQL statements**
- **Create and save reports**

ORACLE

Copyright © 2006, Oracle. All rights reserved.

Objectives

This appendix introduces the graphical tool SQL Developer that simplifies your database development tasks. You learn how to use SQL Worksheet to execute SQL statements and SQL scripts. You also learn how to edit and debug PL/SQL.

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.**

**ORACLE**

Copyright © 2006, 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 by 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**

ORACLE

Copyright © 2006, Oracle. All rights reserved.

Key Features of SQL Developer

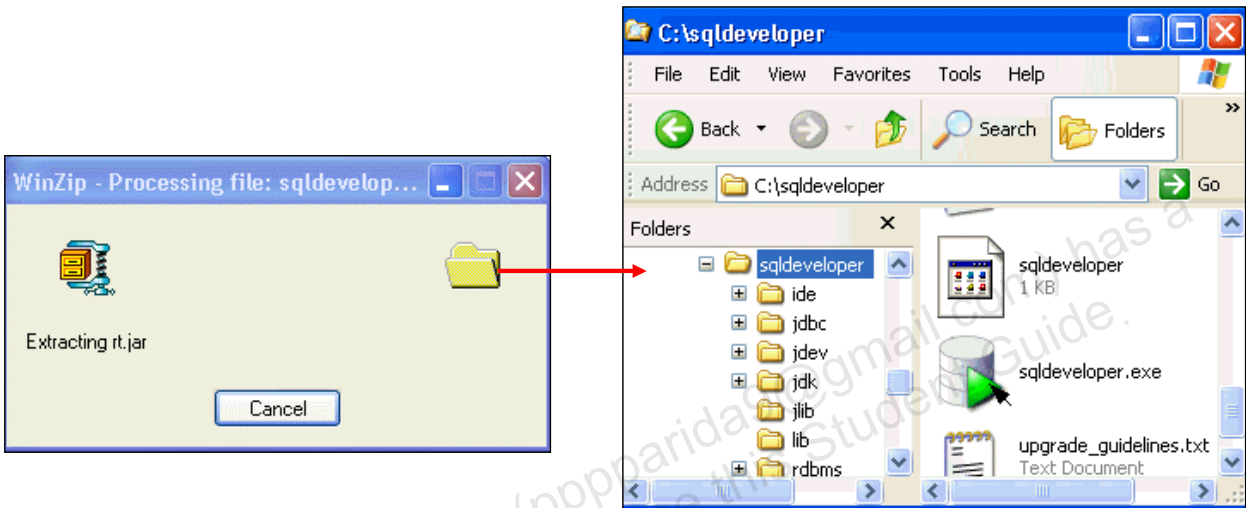
Oracle SQL Developer is developed in Java leveraging the Oracle JDeveloper integrated development environment (IDE). The tool runs on Windows, Linux, and Mac operating system (OS) X platforms. You can install SQL Developer on the Database Server and connect remotely from your desktop, thus avoiding client/server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

Default connectivity to the database is through the Java Database Connectivity (JDBC) Thin driver, so 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. SQL Developer is bundled with Java Runtime Environment (JRE) 1.5, with an additional `tools.jar` to support Windows clients. Non-Windows clients need only Java Development Kit (JDK) 1.5.

Installing SQL Developer

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



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Installing SQL Developer

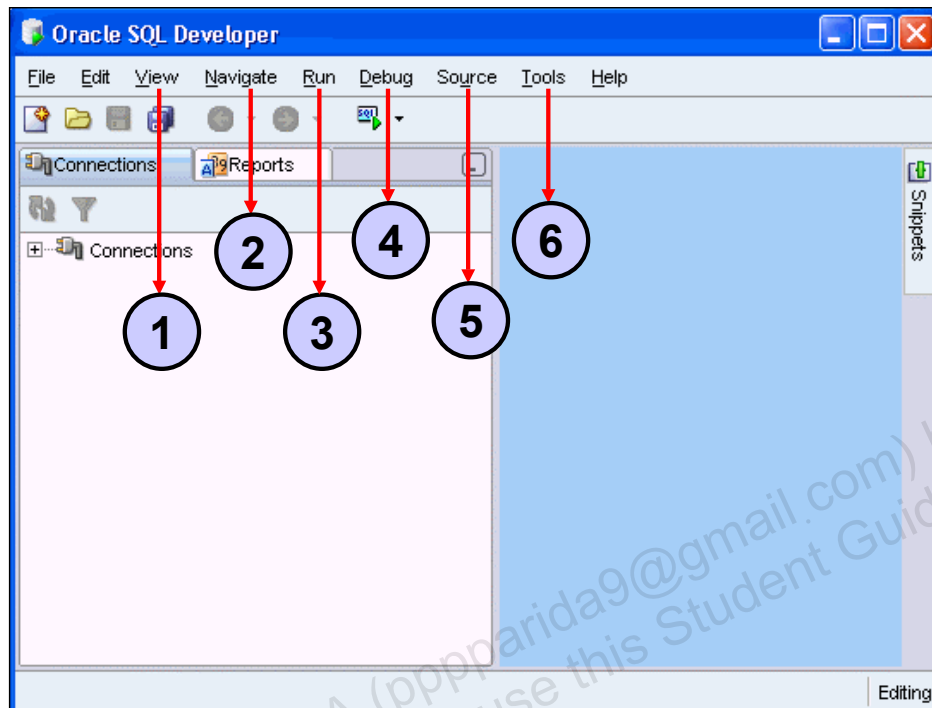
Oracle SQL Developer does not require an installer. To install SQL Developer, you need an unzip tool.

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 © 2006, 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 in 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. **Debug:** Contains options relevant when a function or procedure is selected
5. **Source:** Contains options for use when editing functions and procedures
6. **Tools:** Invokes SQL Developer tools such as SQL*Plus, Preferences, and SQL Worksheet

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 © 2006, 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 as different users to the same database or to connect to the different databases.

Creating a Database Connection

ORACLE

Copyright © 2006, 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 Database Connection**.
3. Enter the connection name, username, password, hostname, 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**.

On the basic tabbed page, at the bottom, enter the following options:

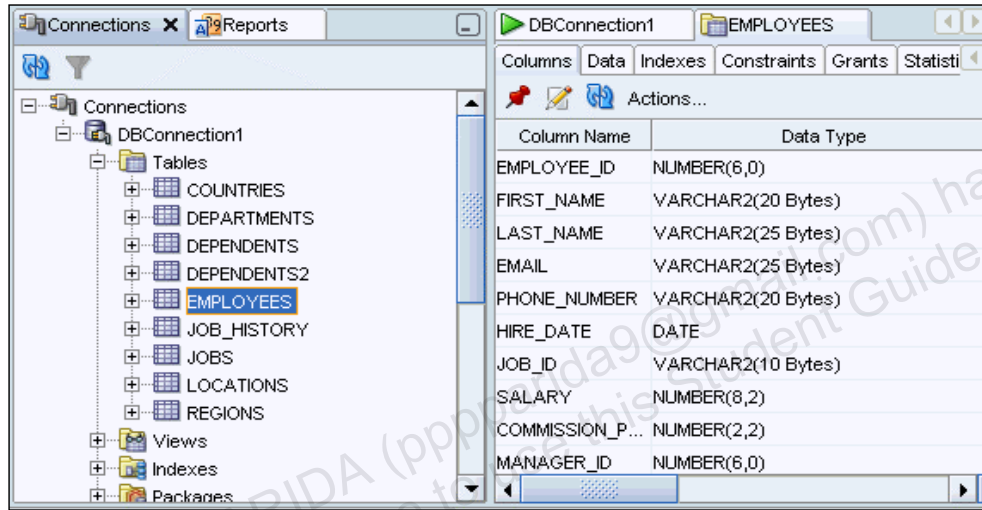
- **Hostname:** Host system for the Oracle database
- **Port:** Listener port
- **SID:** Database name
- **Service Name:** Network service name for a remote database connection

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
- Review the definitions of objects at a glance



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Browsing Database Objects

After you have created 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 so on are displayed in an easy-to-read tabbed page.

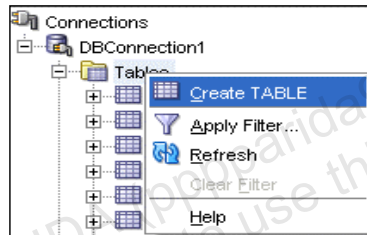
If you want to see the definition of the EMPLOYEES table as shown in the slide, perform the following steps:

1. Expand the Connections node in the Connections Navigator.
2. Expand **Tables**.
3. 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 the SQL Worksheet
 - Using the context menu
- **Edit the objects using an edit dialog box or one of the many context-sensitive menus.**
- **View the DDL for adjustments such as creating a new object or editing an existing schema object.**



ORACLE

Copyright © 2006, Oracle. All rights reserved.

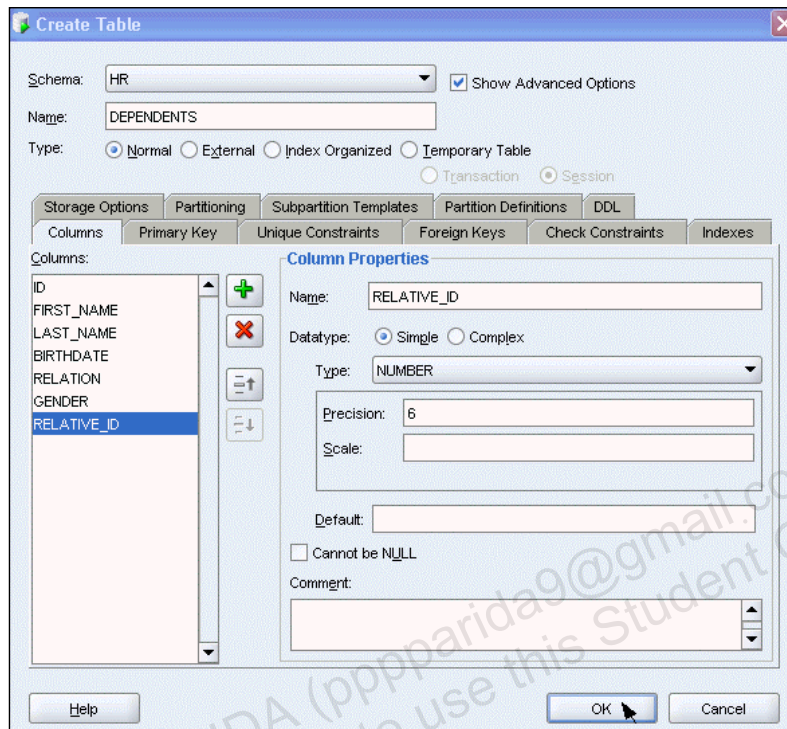
Creating a Schema Object

SQL Developer supports the creation of any schema object by executing a SQL statement in the SQL Worksheet. Alternatively, you can create objects using the context menus. After the objects are 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 data definition language (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 **Create 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 © 2006, Oracle. All rights reserved.

Creating a New Table: Example

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

If you select the **Show Advanced Options** 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 DEPENDENTS table by selecting the **Show Advanced Options** check box.

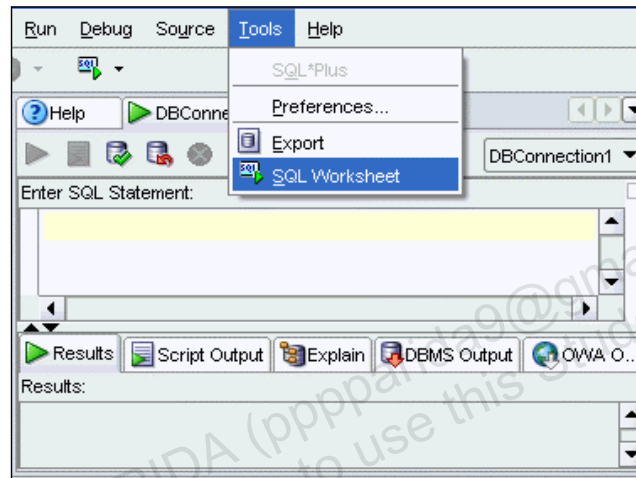
To create a new table, perform the following steps:

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

Although it is not required, you should also specify a primary key using the Primary Key tab 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 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

Copyright © 2006, Oracle. All rights reserved.

Using the SQL Worksheet

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

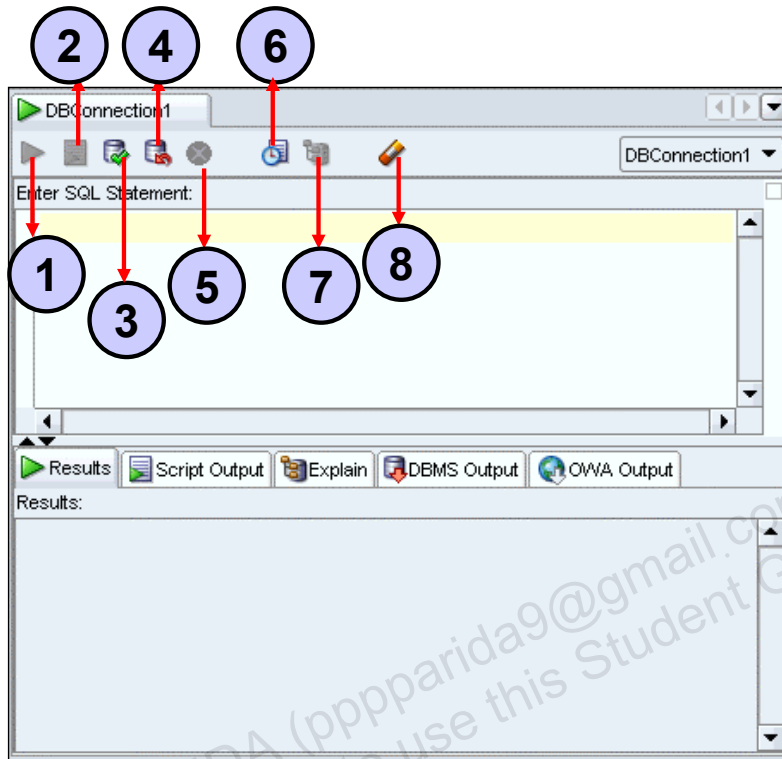
You can specify any actions that can be processed by the database connection associated with the worksheet, such as:

- 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 the SQL Worksheet



ORACLE

Copyright © 2006, Oracle. All rights reserved.

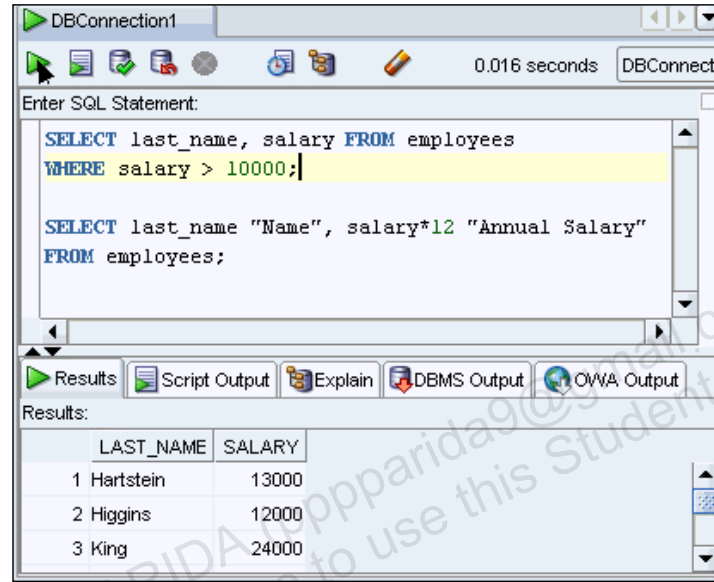
Using the SQL Worksheet (continued)

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 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. **SQL History:** Displays a dialog box with information about SQL statements that you have executed
7. **Execute Explain Plan:** Generates the execution plan, which you can see by clicking the Explain 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 © 2006, Oracle. All rights reserved.

Executing SQL Statements

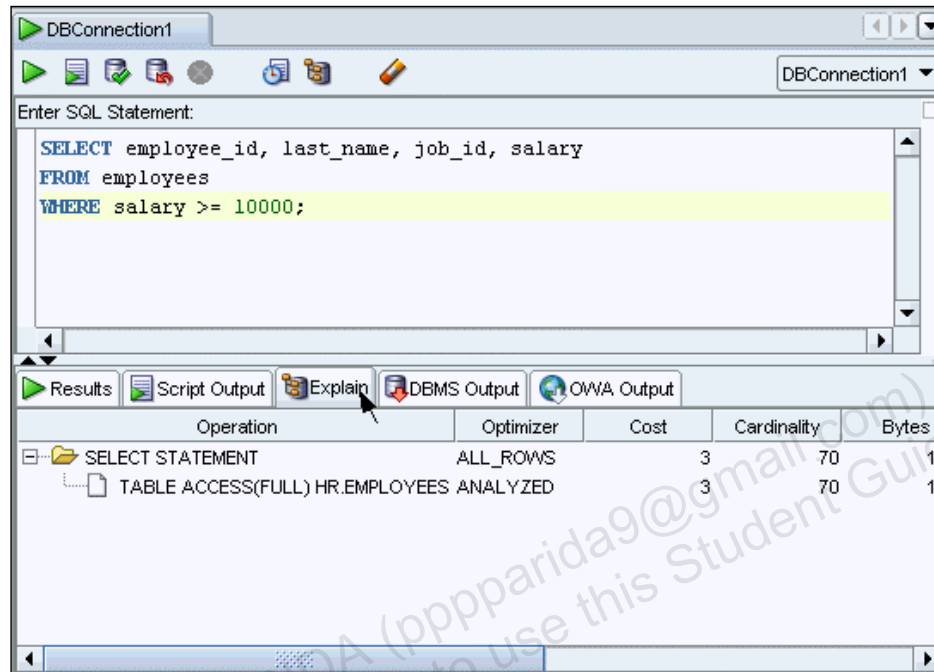
In the SQL Worksheet, you can use the Enter SQL Statement box to type a single or multiple SQL statements. For a single statement, the semicolon at the end is optional.

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, because there are multiple SQL statements, the first statement is terminated with a semicolon. The cursor is in the first statement and so when the statement is executed, results corresponding to the first statement are displayed in the Results box.

Viewing the Execution Plan



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Viewing the Execution Plan

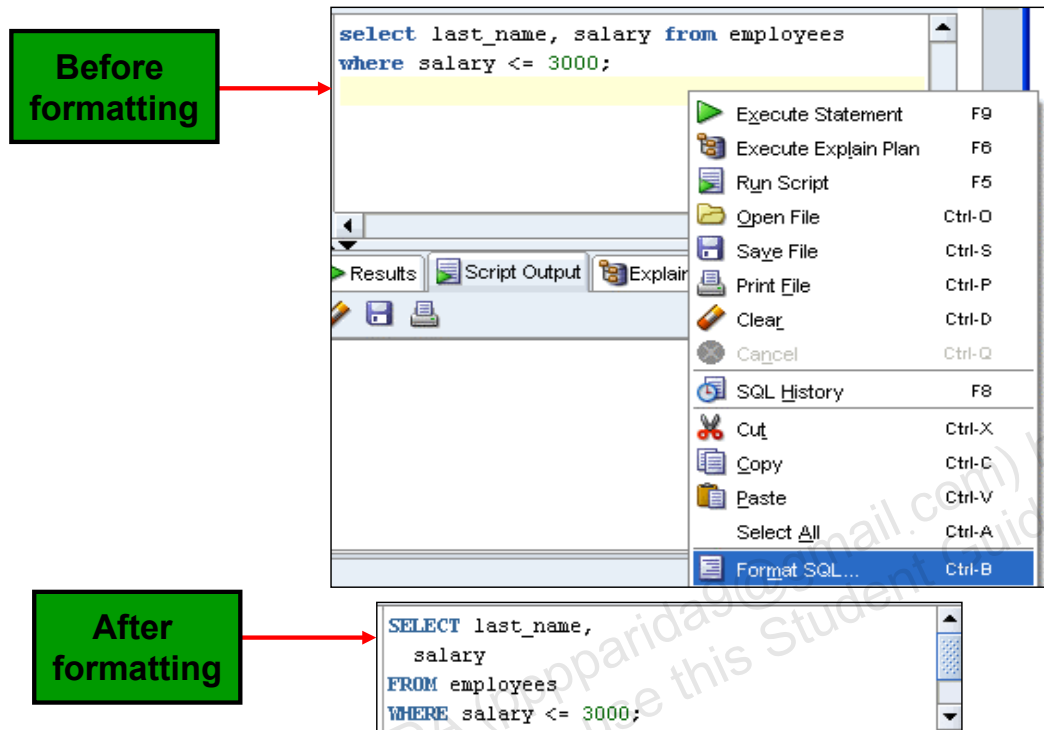
You can execute a SQL script, and view the execution plan. To execute a SQL script file, perform the following steps:

1. Right-click in the Enter SQL Statement box, and select **Open File** from the drop-down menu.
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 will be performed to execute the statement. You can see the execution plan by clicking the **Explain** tab.

Formatting the SQL Code



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Formatting the SQL Code

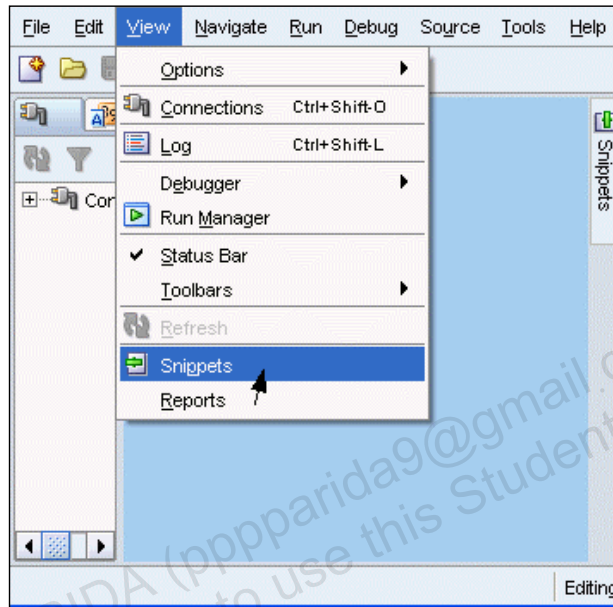
You may want to beautify the indentation, spacing, capitalization, and line separation of the SQL code. SQL Developer has the feature of formatting the SQL code.

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

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



ORACLE

Copyright © 2006, Oracle. All rights reserved.

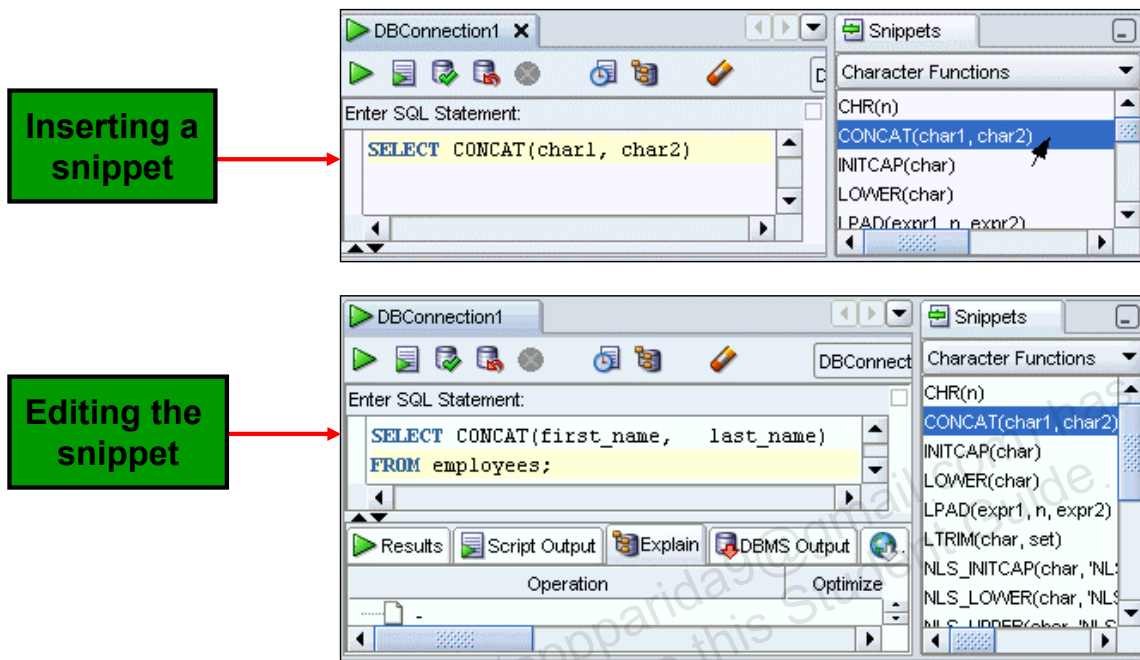
Using Snippets

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 snippets into the Editor window.

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

The Snippets window is displayed on the right side. 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 © 2006, Oracle. All rights reserved.

Using Snippets: Example

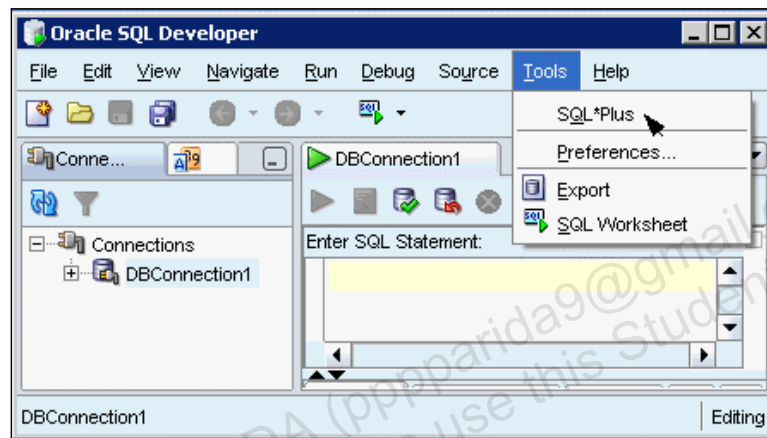
To insert a Snippet into your code in a SQL Worksheet or in a PL/SQL function or procedure, drag the snippet from the Snippets window into the desired place in your code. Then you can edit the syntax so that the SQL function is valid in the current context. To see a brief description of a SQL function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

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

Using SQL*Plus

- The SQL Worksheet does not support all SQL*Plus statements.
- You can invoke the SQL*Plus command-line interface from SQL Developer.



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Using SQL*Plus

The 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:

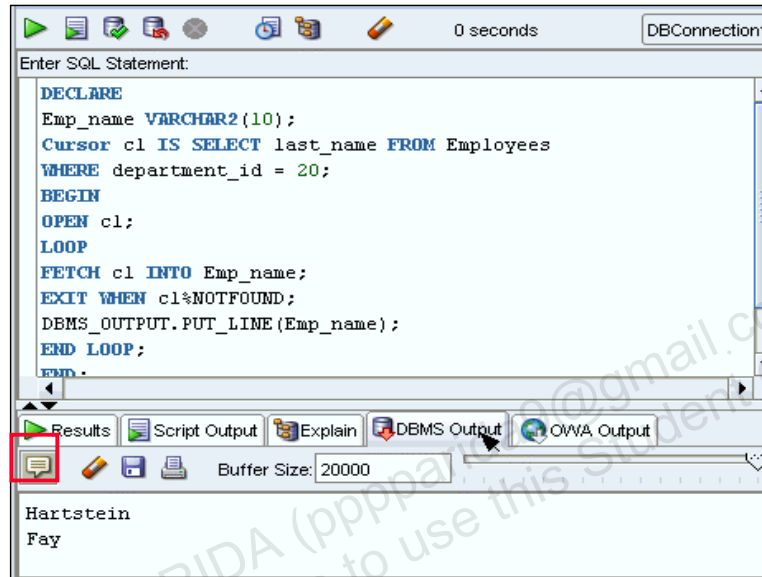
- append
- archive
- attribute
- break

For the complete list of SQL*Plus statements that are supported, and not supported by SQL Worksheet, refer to SQL Developer online Help.

To display the SQL*Plus command window, from the Tools menu, select **SQL*Plus**. To use this feature, the system on which you are using SQL Developer must have an Oracle Home directory or folder, with a SQL*Plus executable under that location. If the location of the SQL*Plus executable is not already stored in your SQL Developer preferences, you are asked to specify its location.

Creating an Anonymous Block

Create an anonymous block and display the output of the DBMS_OUTPUT package statements.



ORACLE

Copyright © 2006, Oracle. All rights reserved.

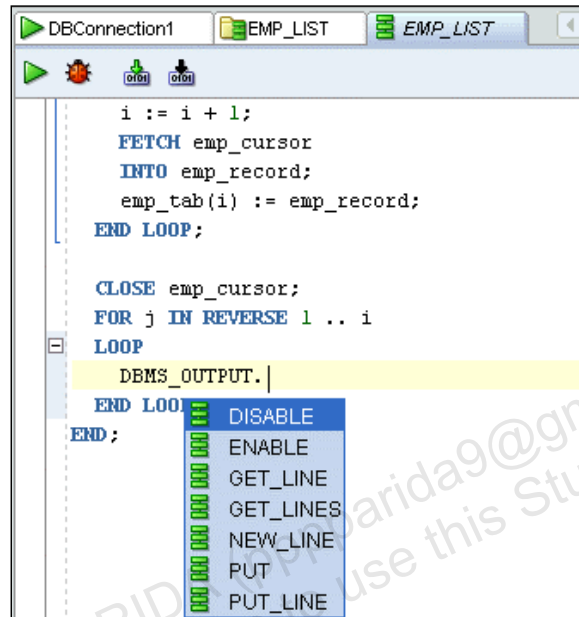
Creating an Anonymous Block

You can create an anonymous block and display the output of the DBMS_OUTPUT package statements. To create an anonymous block and view the results, perform the following steps:

1. Enter the PL/SQL code in the Enter SQL Statement box.
2. Click the **DBMS Output** pane. Then click the **Enable DBMS Output** icon to set the server output ON.
3. Click the **Execute Statement** icon above the Enter SQL Statement box. Then click the **DBMS Output** pane to see the results.

Editing the PL/SQL Code

Use the full-featured editor for PL/SQL program units.



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Editing the PL/SQL Code

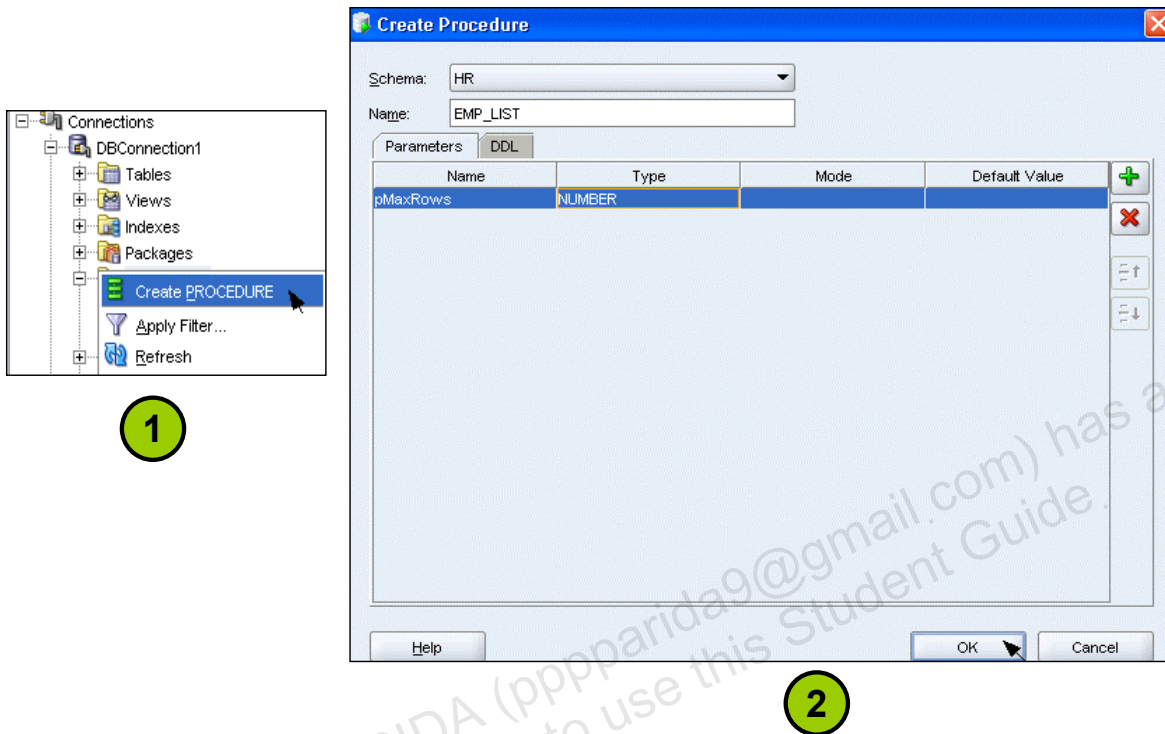
You may want to make changes to your PL/SQL code. SQL Developer includes a full-featured editor for PL/SQL program units. It includes customizable PL/SQL syntax highlighting in addition to common editor functions such as:

- Bookmarks
- Code Completion
- Code Folding
- Search and Replace

To edit the PL/SQL code, click the object name in the Connections Navigator, and then click the **Edit** icon. Optionally, double-click the object name to invoke the Object Definition page with its tabs and the Edit page. You can update only if you are on the Edit tabbed page. The Code Insight feature is shown in the slide. For example, if you type `DBMS_OUTPUT.` and then press [Ctrl] + [Spacebar], you can select from a list of members of that package. Note that, by default, Code Insight is invoked automatically if you pause after typing a period (".") for more than one second.

When using the Code Editor to edit PL/SQL code, you can Compile or Compile for Debug.

Creating a PL/SQL Procedure



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Creating a PL/SQL Procedure

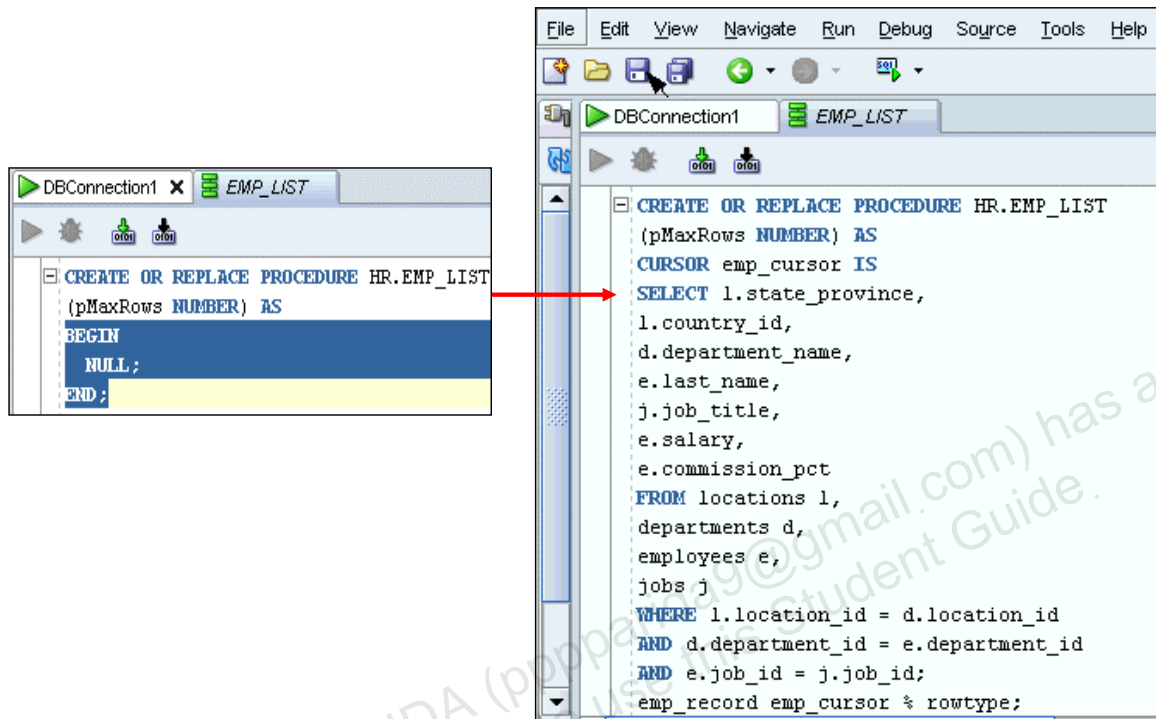
Using SQL Developer, you can create PL/SQL functions, procedures, and packages. To create a PL/SQL procedure, perform the following steps:

1. Right-click the Procedures node in the Connections Navigator to invoke the Context menu, and select **Create Procedure**.
2. In the Create Procedure dialog box, specify the procedure information and click **OK**.

Note: Ensure that you press Enter before you click OK.

In the example in the slide, the EMP_LIST procedure is created. The default values for parameter name and parameter type are replaced with pMaxRows and NUMBER, respectively.

Compiling a PL/SQL Procedure



ORACLE

Copyright © 2006, Oracle. All rights reserved.

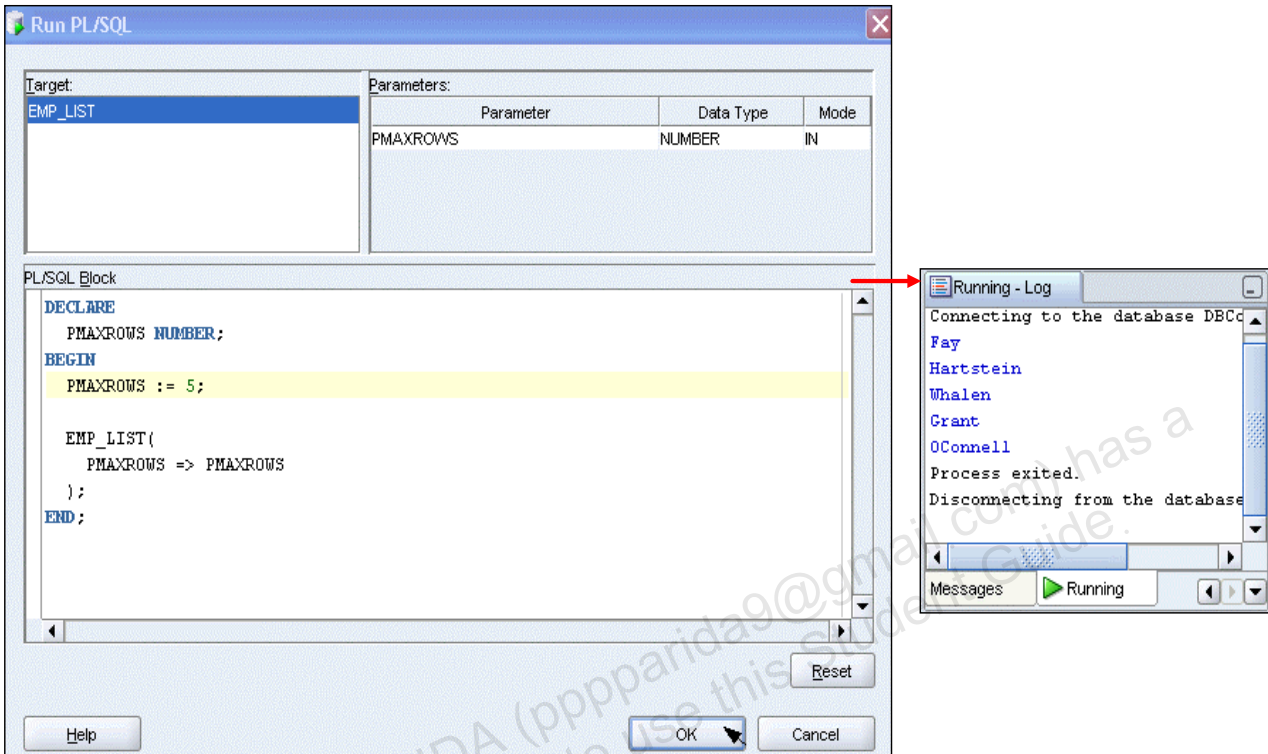
Compiling a PL/SQL Procedure

After you specify the parameter information in the Create Procedure dialog box and click OK, you see the Procedure tab added in the right window. You can then replace the Anonymous block with your PL/SQL code.

To compile the PL/SQL subprogram, click the Save button in the toolbar. If you expand Procedures in the Connections Navigator, you can see that the Procedure node is added.

When an invalid PL/SQL subprogram is detected by SQL Developer, the status is indicated with a red X over the icon for the subprogram in the Connections Navigator. Compilation errors are shown in the log window. You can navigate to the line reported in the error by simply double-clicking the error. SQL Developer also displays errors and hints in the right-hand gutter. If you place the cursor over each of the red bars in the gutter, the error message appears. For example, if the error messages indicate that there is a formatting error, modify the code accordingly and click the Compile icon.

Running a PL/SQL Procedure



ORACLE

Copyright © 2006, Oracle. All rights reserved.

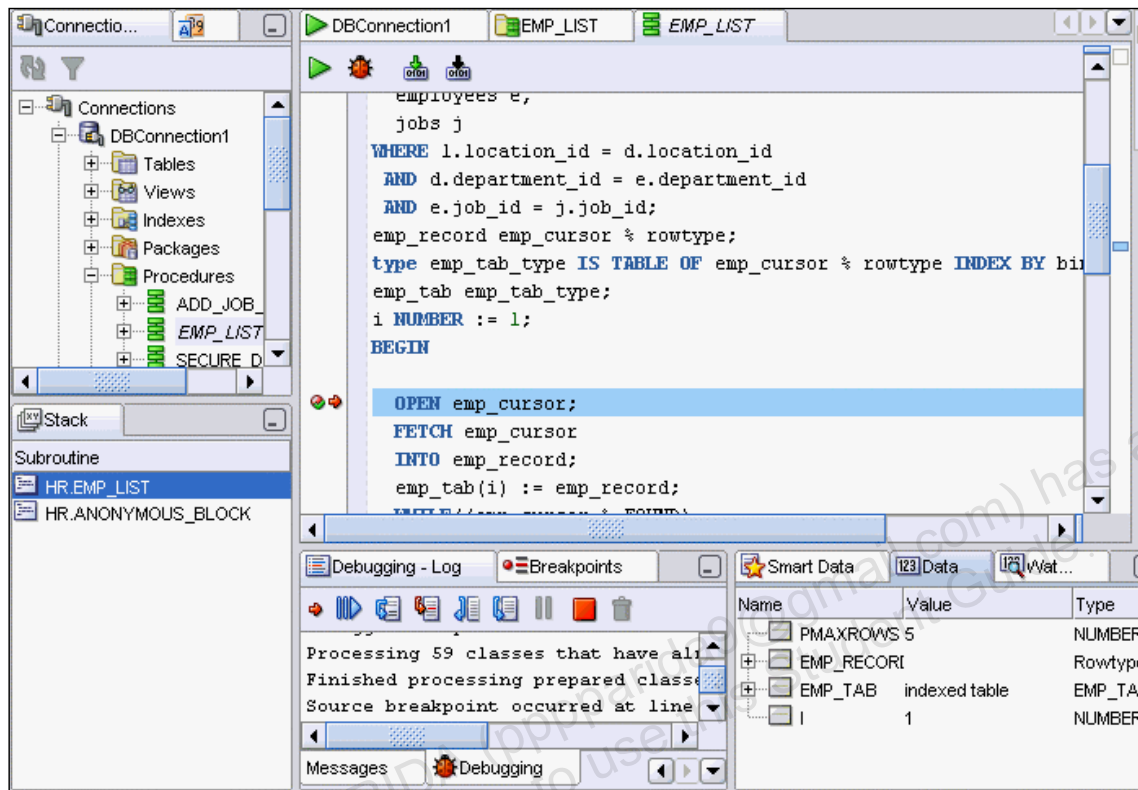
Running a PL/SQL Procedure

After you have created and compiled a PL/SQL procedure, you can run it using SQL Developer. To run a PL/SQL procedure, right-click the procedure name in the left navigator and select Run. Optionally, you can use the Run button in the right window. This invokes the Run PL/SQL dialog box. The Run PL/SQL dialog box enables you to select the target procedure or function to run and displays a list of parameters for the selected target.

You can use the PL/SQL block area to populate parameters to be passed to the program unit and to handle complex return types. After you make the necessary changes in the Run PL/SQL dialog box, click **OK**. You see the expected results in the Running-Log window.

In the example in the slide, `PMAXROWS := NULL;` is changed to `PMAXROWS := 5;`. The results of the five rows returned are displayed in the Running-Log window.

Debugging PL/SQL



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Debugging PL/SQL

You may want to debug a PL/SQL function, procedure, or package. SQL Developer provides full support for PL/SQL debugging. To debug a function or procedure, perform the following steps:

1. Click the object name in the Connections Navigator.
2. Right-click the object and select **Compile for debug**.
3. Click the **Edit** icon. Then click the **Debug** icon above its source listing.

If the toggle numbers before each line of code is not yet displayed, right-click in the Code Editor margin and select **Toggle Line Numbers**.

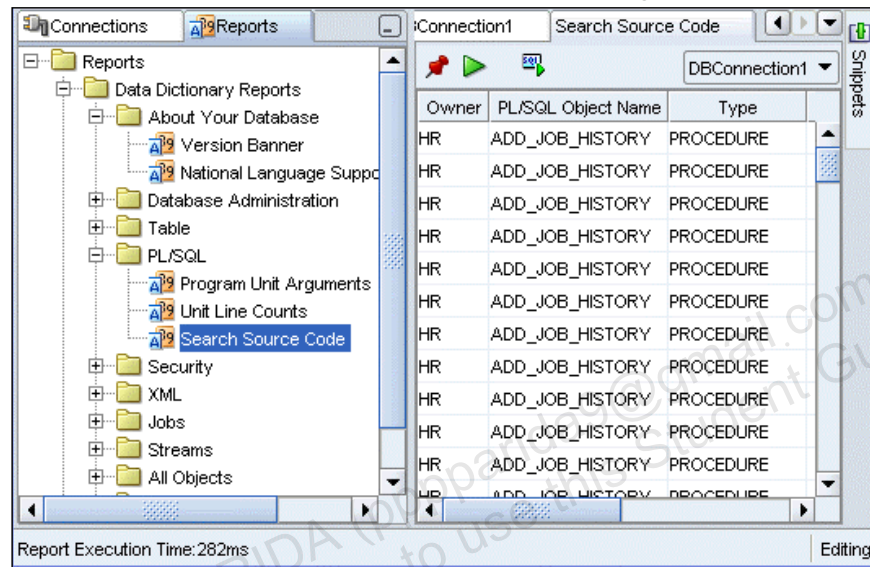
The PL/SQL debugger supplies many commands to control program execution including Step Into, Step Over, Step Out, Run to Cursor, and so on. While the debugger is paused, you can examine and modify the values of variables from the Smart Data, Watches, or Inspector windows.

The Breakpoints window lists the defined breakpoints. You can use this window to add new breakpoints, or customize the behavior of existing breakpoints.

Note: For PL/SQL debugging, you need the debug any procedure and debug connect session privileges.

Database Reporting

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



ORACLE

Copyright © 2006, Oracle. All rights reserved.

Database Reporting

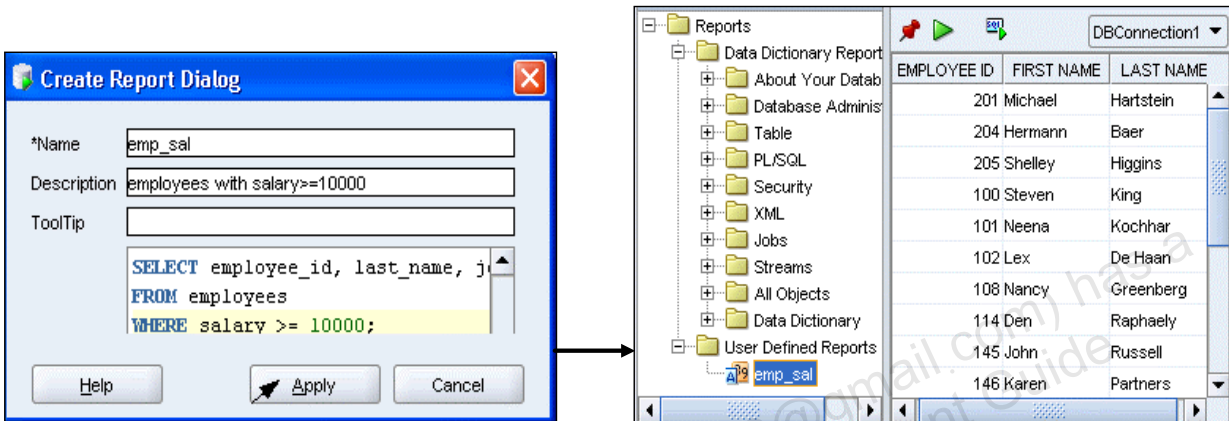
SQL Developer provides many reports about the database and its objects. These reports can be grouped into the following categories:

- 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 on the left side of the window. Individual reports are displayed in tabbed panes on 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.

Creating a User-Defined Report

Create and save user-defined reports for repeated use.



ORACLE

Copyright © 2006, 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 about employees with salary equal to or greater than 10,000. 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 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 the SQL Worksheet**
- **Edit and debug PL/SQL statements**
- **Create and save custom reports**

ORACLE

Copyright © 2006, 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 the SQL Worksheet to run SQL statements and scripts. Using SQL Developer, you can edit and debug PL/SQL. SQL Developer enables you to create and save your own special set of reports for repeated use.