



# **Using SQL Developer**

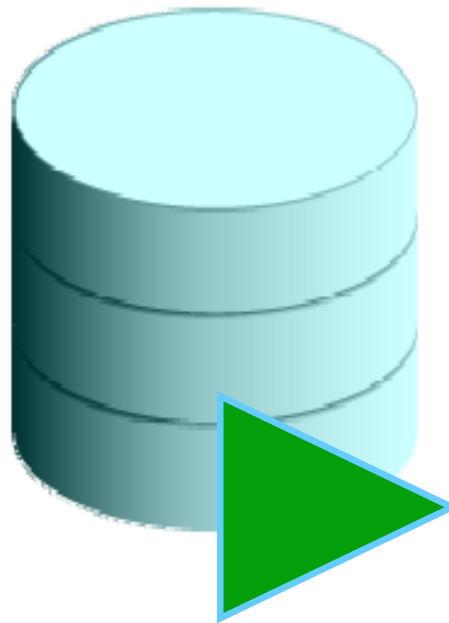
# Objectives

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

# 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.

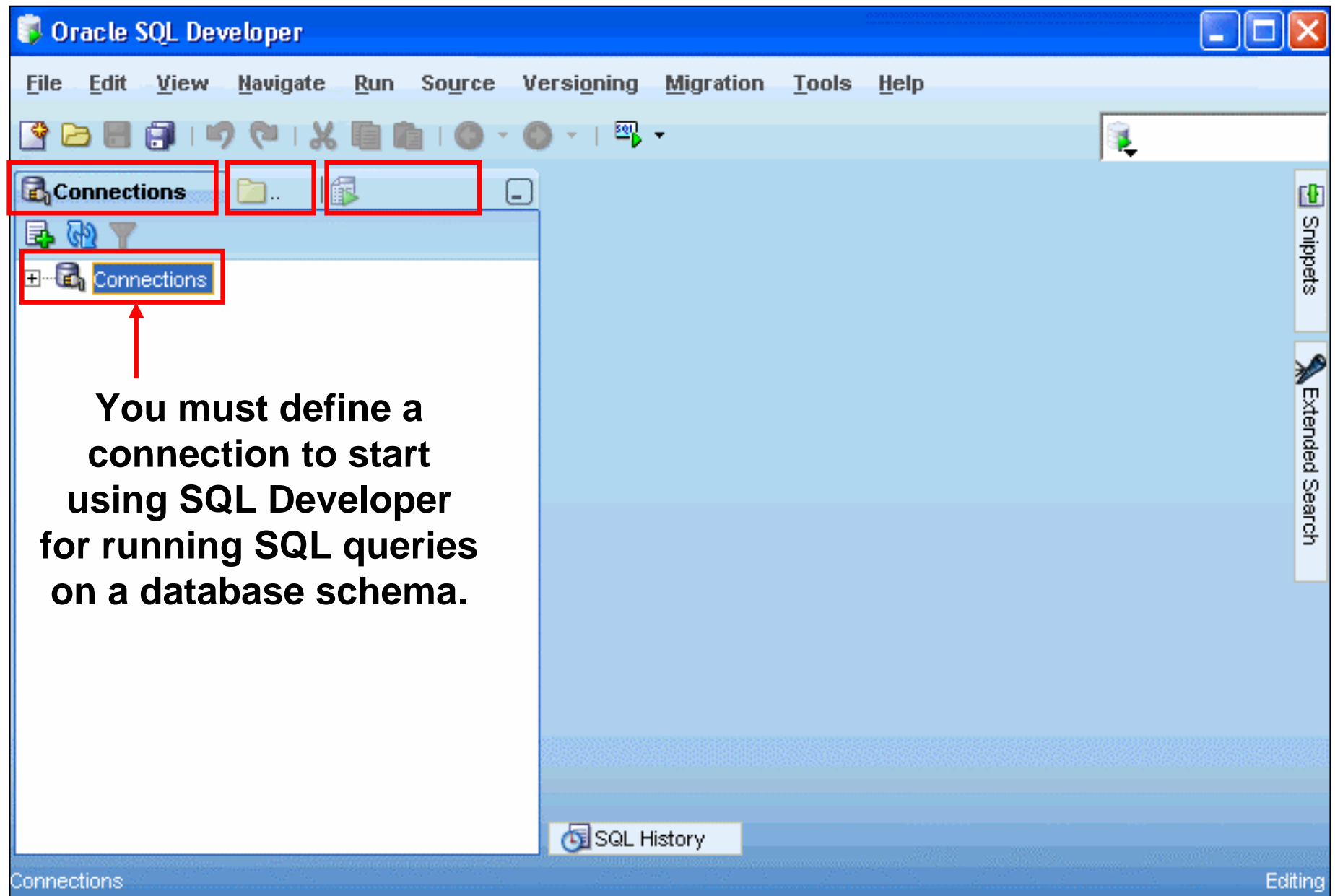


**SQL Developer**

# 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](http://www.oracle.com/technology/products/database/sql_developer/index.html)

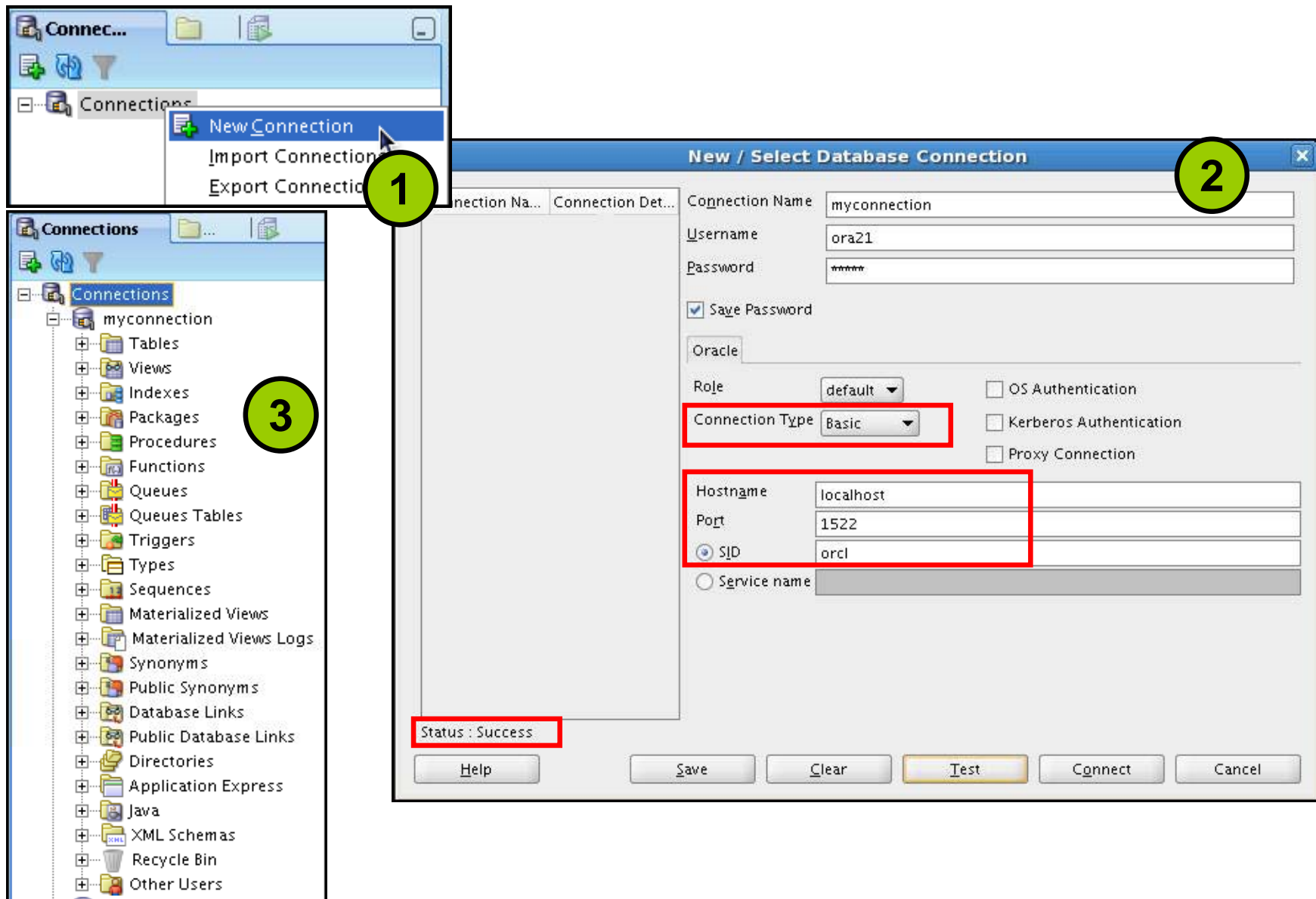
# SQL Developer 1.5 Interface



# 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.

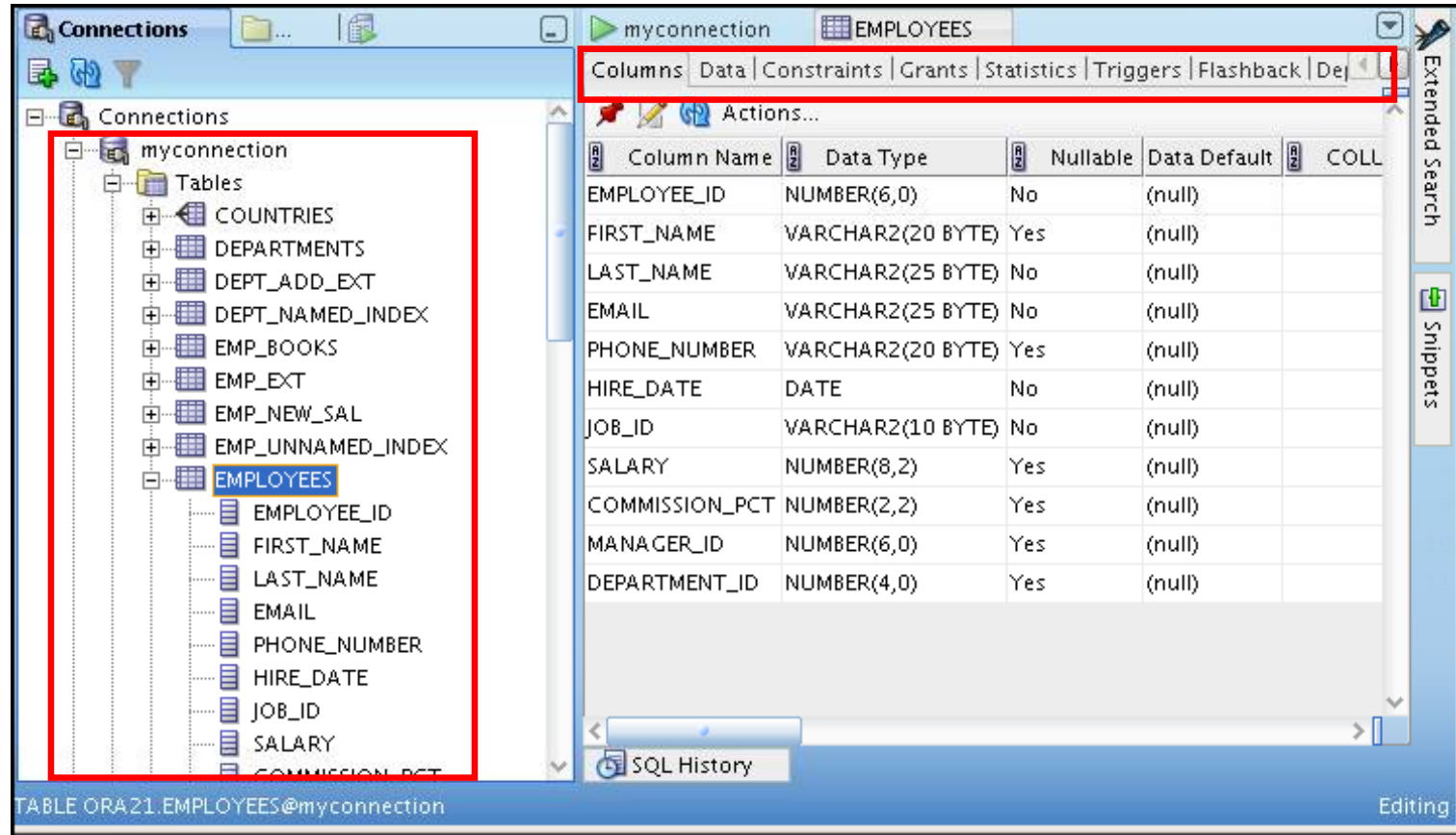
# Creating a Database Connection



# Browsing Database Objects

Use the Connections Navigator to to:

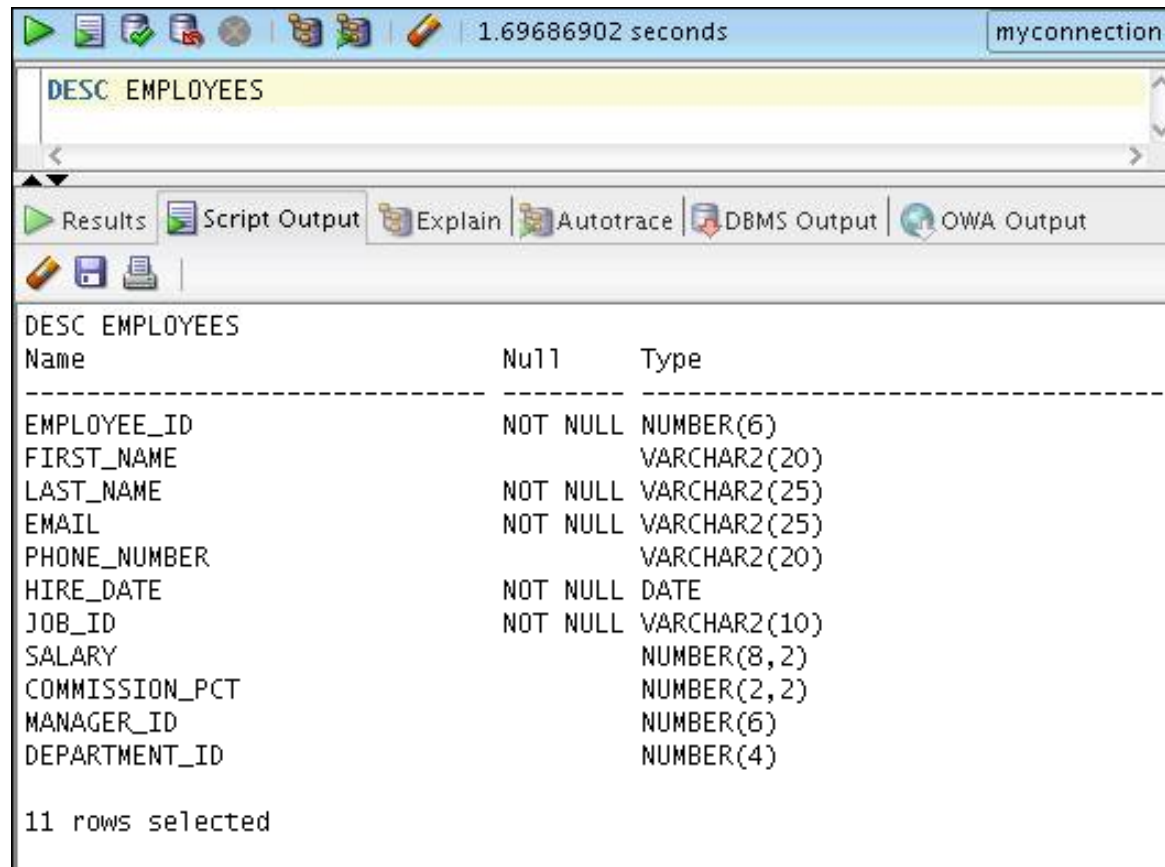
- Browse through many objects in a database schema
- Review the definitions of objects at a glance





# Displaying the Table Structure

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

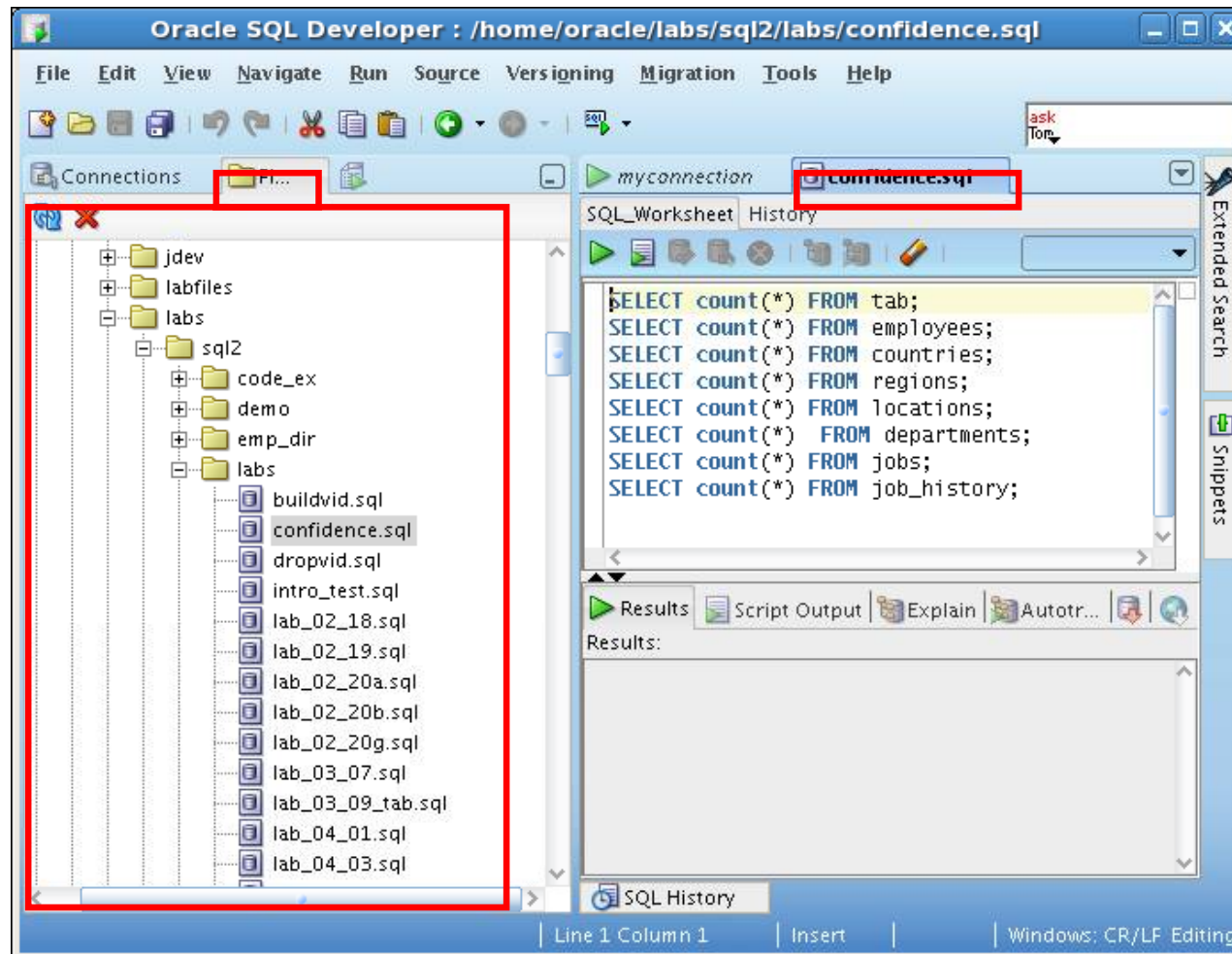


DESC EMPLOYEES	Null	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

11 rows selected

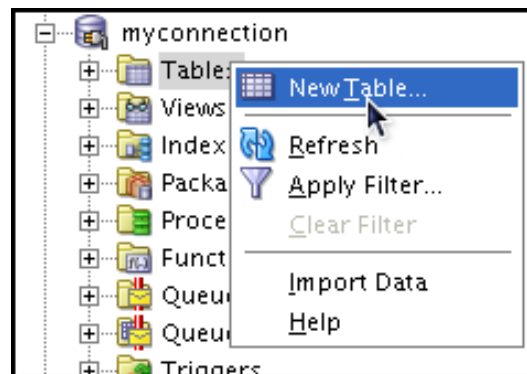
# Browsing Files

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



# 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.



# Creating a New Table: Example

**Create Table**

Schema: ORAZ1

Name: Dependents

Table Type: ☒ Normal ☐ External ☐ Index Organized ☐ Temporary (Transaction) ☐ Temporary (Session)

**Columns:**

- ID
- FIRST\_NAME
- LAST\_NAME
- RELATION
- BIRTHDATE

**Column Properties**

Name: ID

Datatype: ☒ Simple ☐ Complex

Type: NUMBER

Precision:

Scale:

Default:

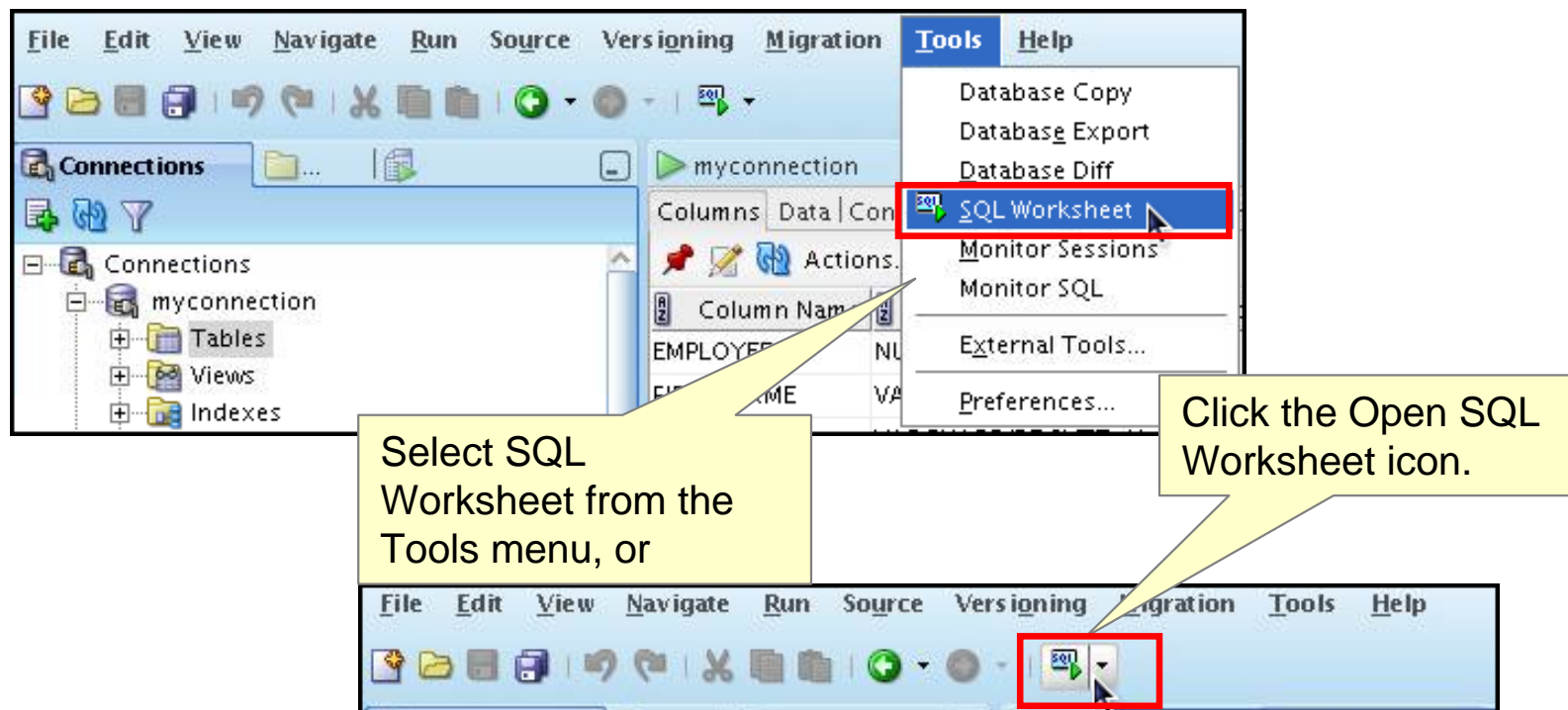
☒ Cannot be NULL

Comment:

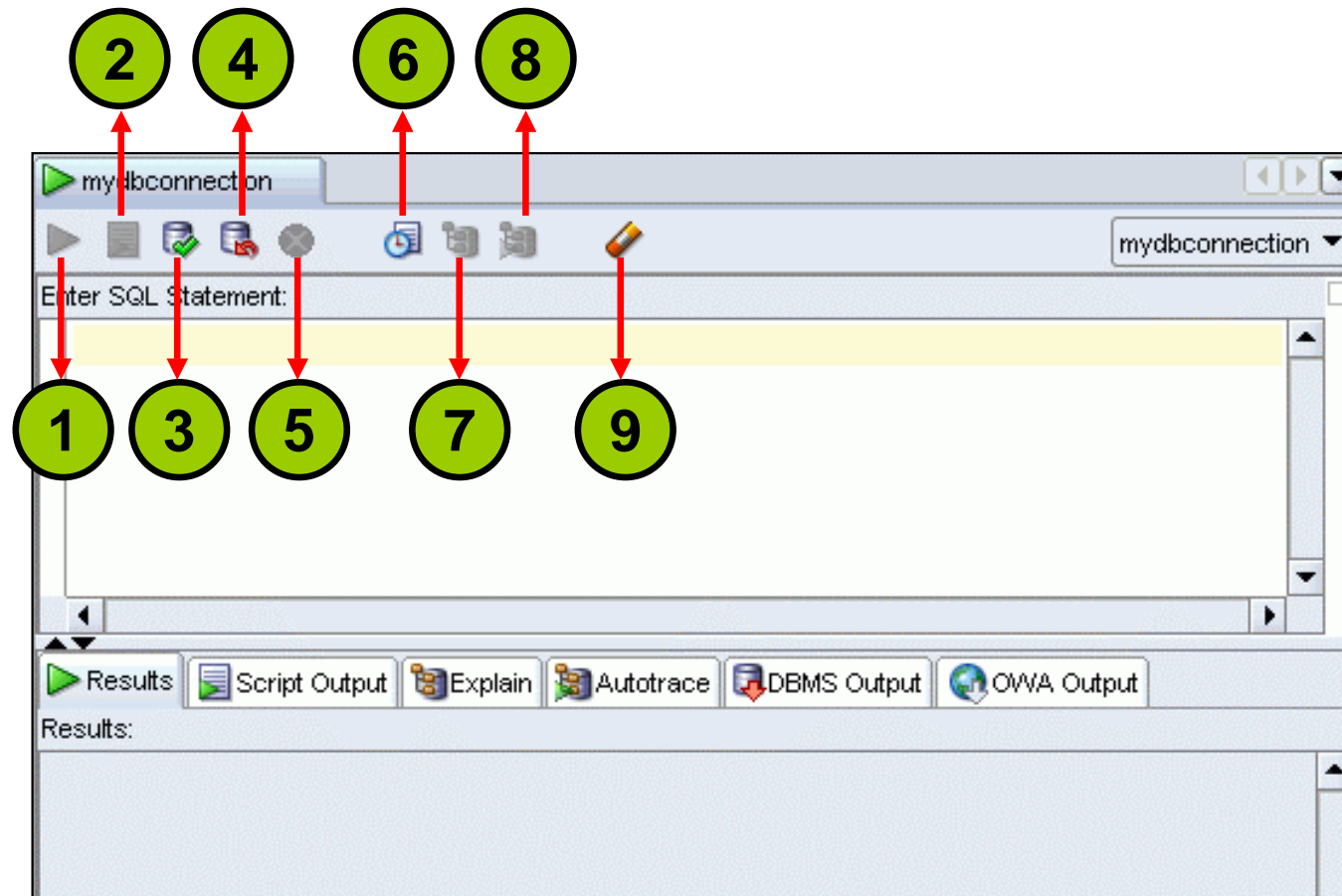
Buttons: Help, OK, Cancel

# 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.

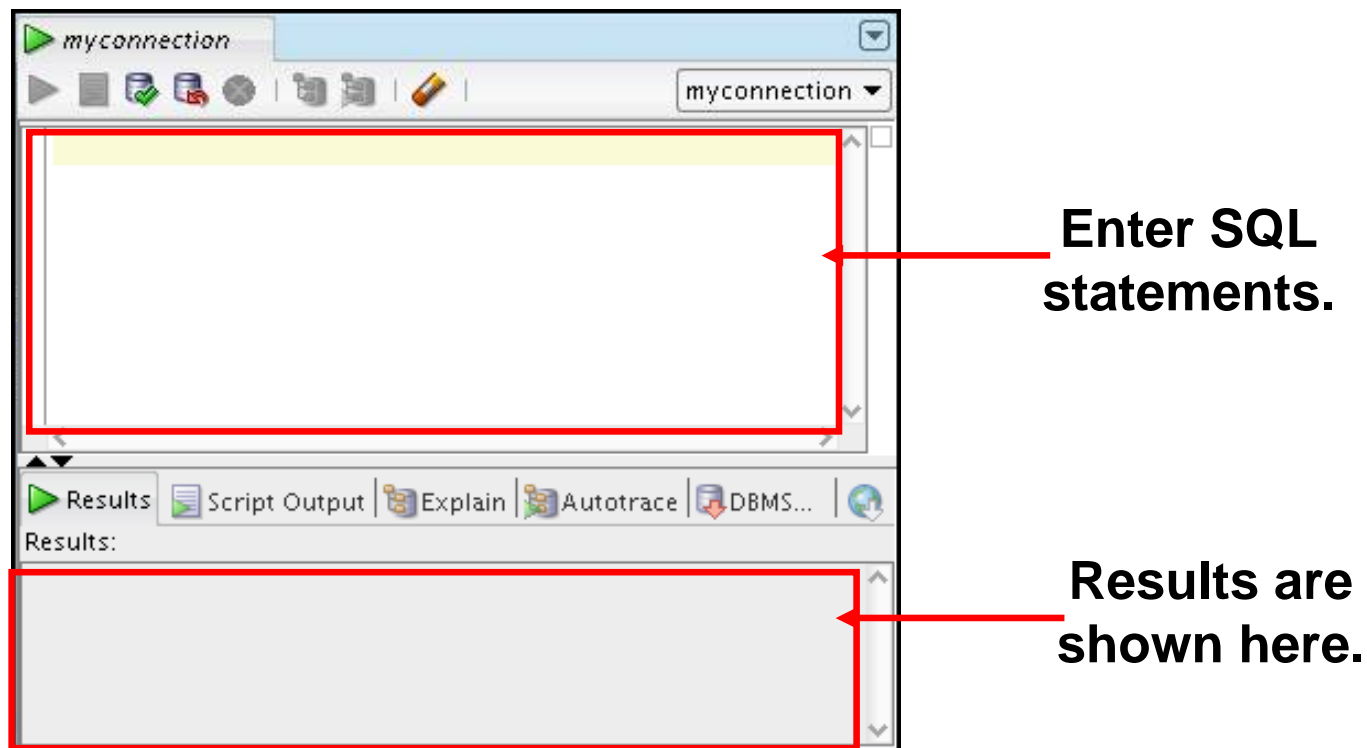


# Using the SQL Worksheet



# 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.



# Executing SQL Statements

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

The screenshot illustrates the workflow for executing an SQL statement in Oracle SQL Developer. The main window, titled 'myconnection', contains an 'Enter SQL Statement' box with the following query:

```
SELECT employee_id, last_name
FROM employees;
```

Annotations indicate the following steps:

- F9**: Pressed to execute the SQL statement. A red arrow points from this label to the 'Execute' button (a green play icon) in the toolbar.
- F5**: Pressed to view the script output. A red arrow points from this label to the 'Script Output' button (a document icon) in the toolbar.
- F9**: Pressed to view the results. A red arrow points from this label to the 'Results' button (a green play icon) in the toolbar.

Two windows display the output of the query:

**Results Window:**

	EMPLOYEE_ID	LAST_NAME
1	100	King
2	101	Kochhar
3	102	De Haan
4	103	Hunold
5	104	Ernst

**Script Output Window:**

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
105	Austin



# Saving SQL Scripts

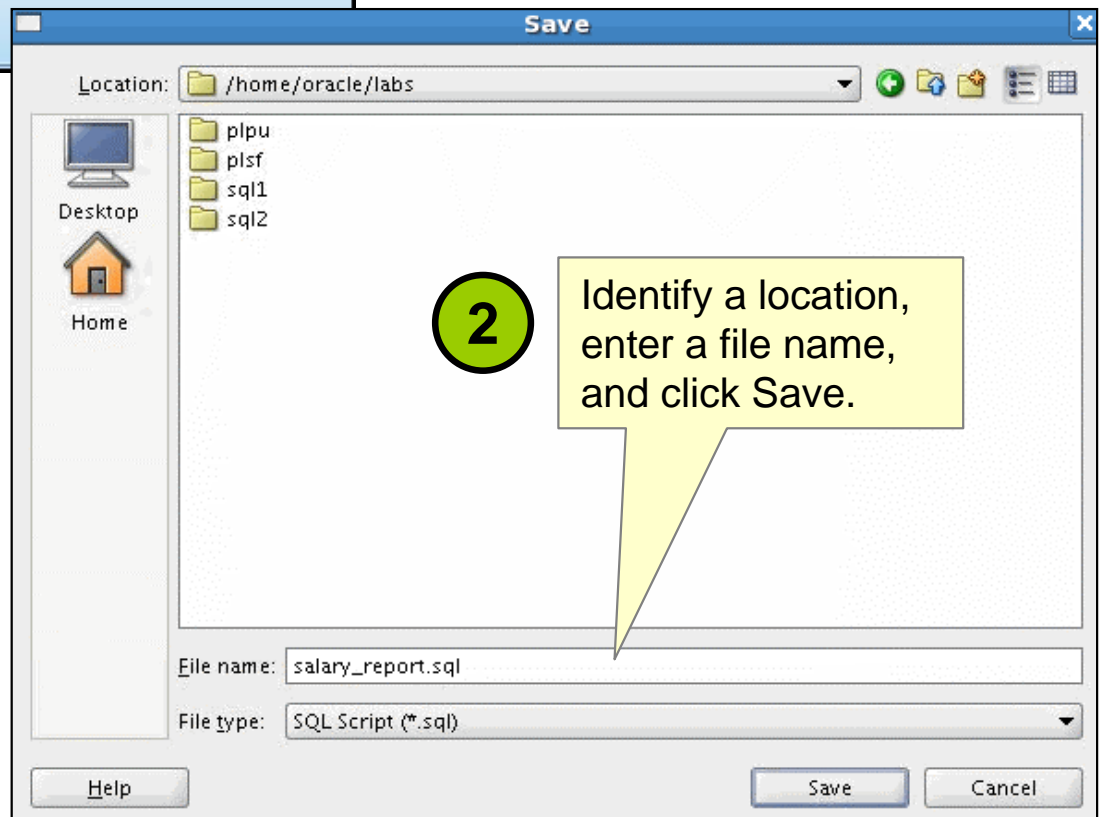
1

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



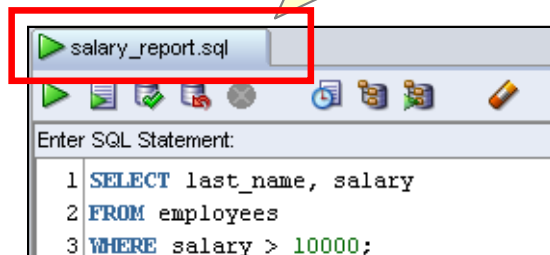
2

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.



# Executing Saved Script Files: Method 1

1. Use the Files tab to locate the script file that you want to open.

2. Double-click the script to display the code in the SQL Worksheet.

To run the code, click either:

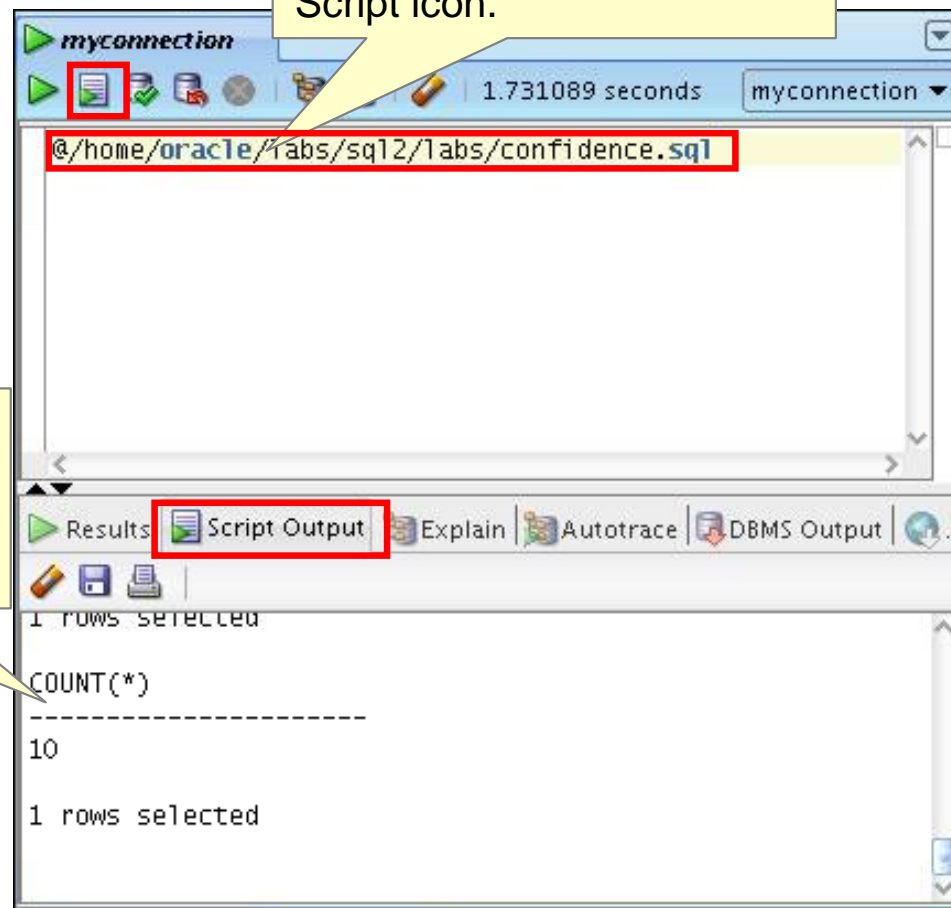
- Execute Script (F9), or
- Run Script (F5)

2. Select a connection from the drop-down list.

```
SELECT count(*) FROM tab;
SELECT count(*) FROM employees;
SELECT count(*) FROM countries;
SELECT count(*) FROM regions;
SELECT count(*) FROM locations;
SELECT count(*) FROM departments;
SELECT count(*) FROM jobs;
SELECT count(*) FROM job_history;
```

# Executing Saved Script Files: Method 2

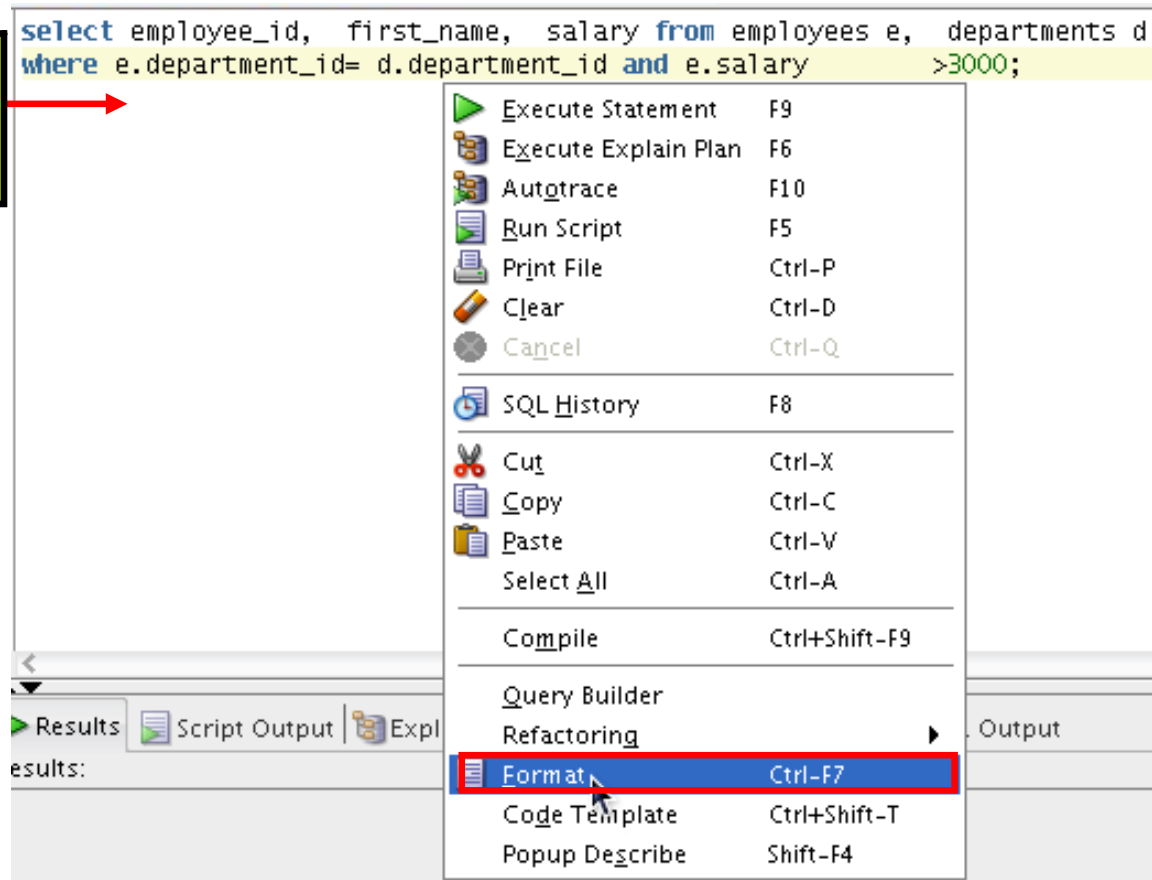
Use the @ command followed by the location and name of the file you want to execute, and click the Run Script icon.



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

# Formatting the SQL Code

**Before  
formatting**

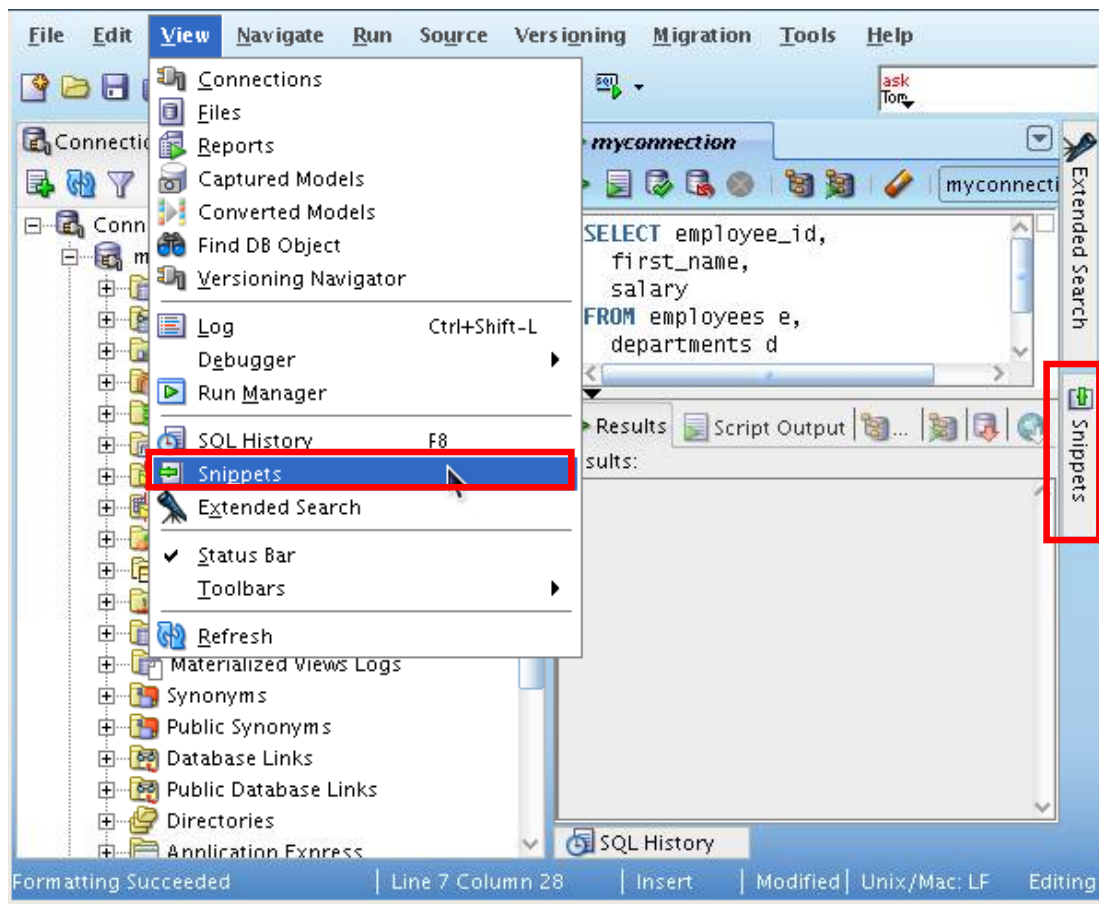


**After  
formatting**

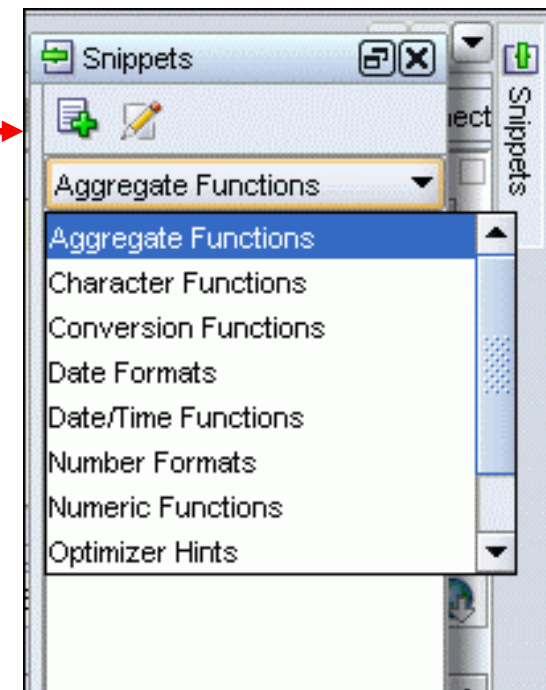
A screenshot showing the same SQL query after formatting. The code is now properly indented and uses uppercase keywords: `SELECT employee_id,  
first_name,  
salary  
FROM employees e,  
departments d  
WHERE e.department_id= d.department_id  
AND e.salary > 3000;`. A red arrow points from the 'After formatting' label to the formatted text.

# Using Snippets

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

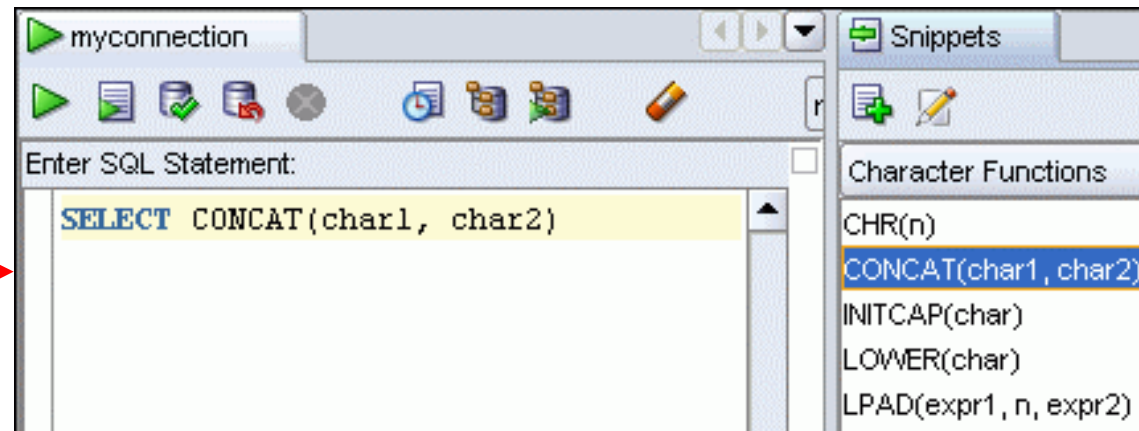


When you place your cursor here, it shows the Snippets window. From the drop-down list, you can select the functions category that you want.

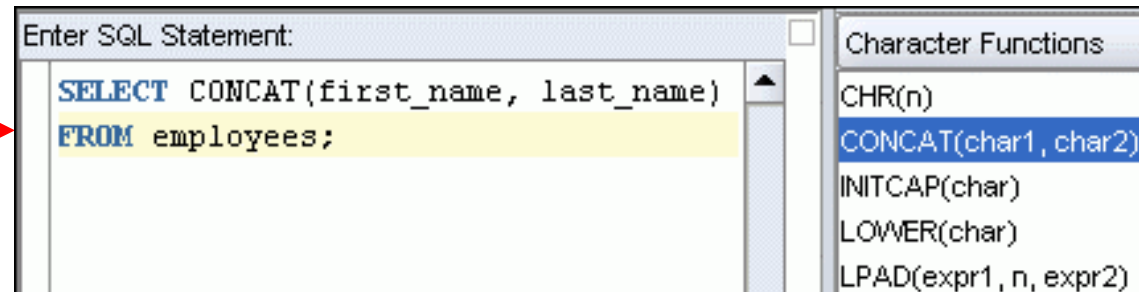


# Using Snippets: Example

**Inserting a snippet**

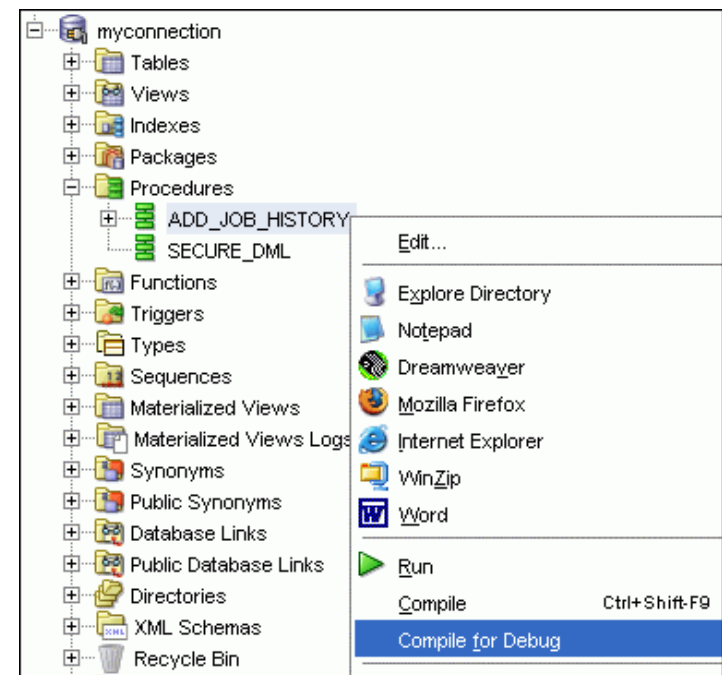


**Editing the snippet**



# Debugging Procedures and Functions

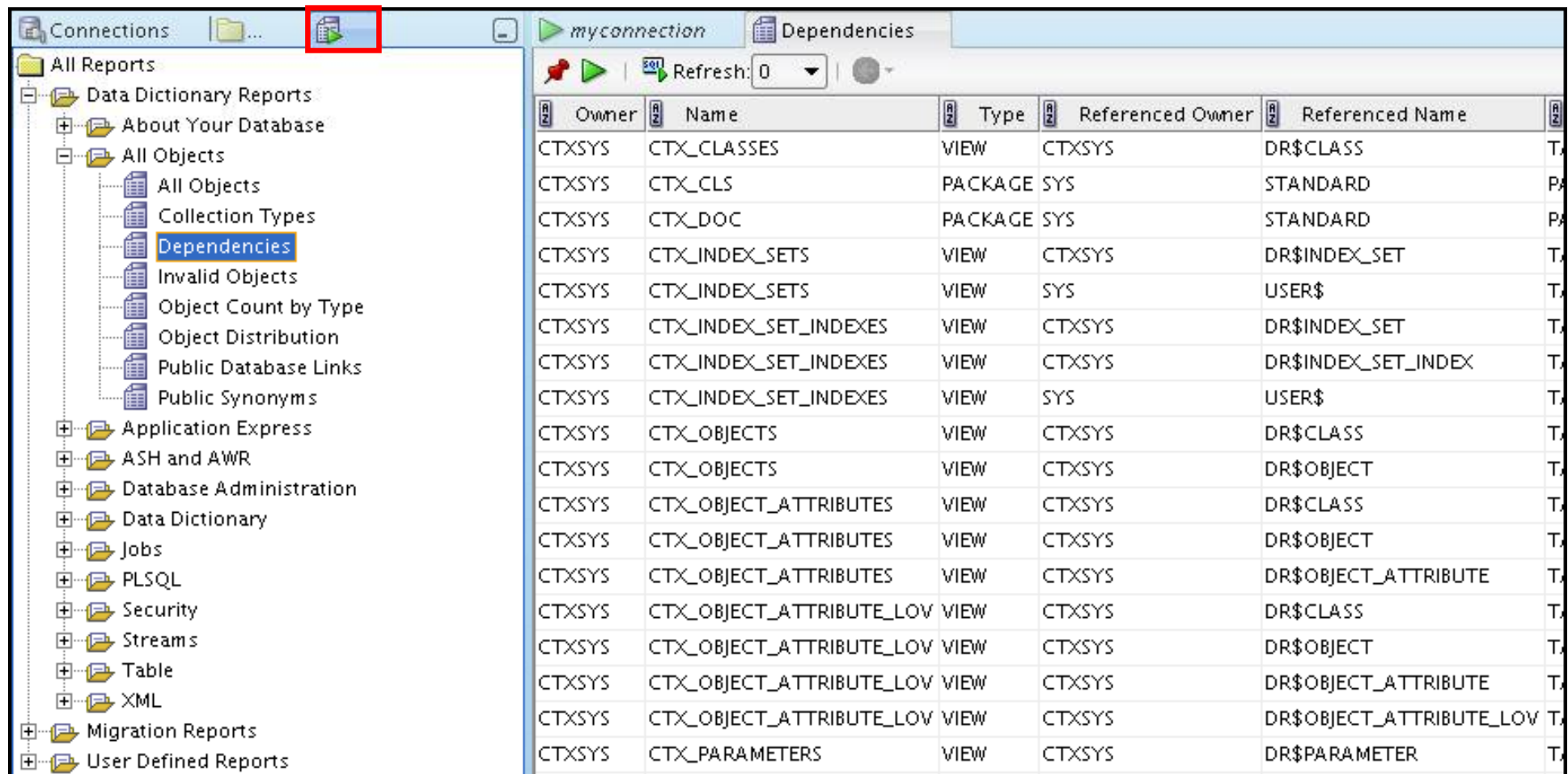
- Use SQL Developer to debug PL/SQL functions and procedures.
- Use the “Compile for Debug” option to perform a PL/SQL compilation so that the procedure can be debugged.
- Use Debug menu options to set breakpoints, and to perform step into and step over tasks.





# Database Reporting

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



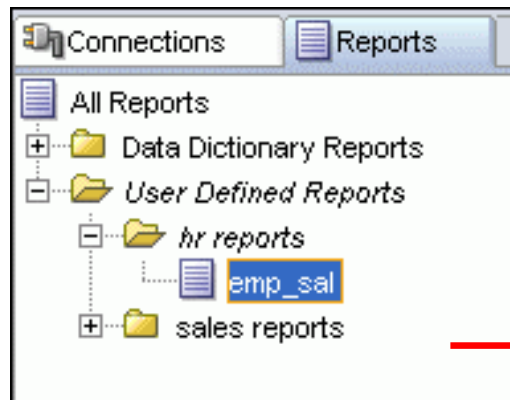
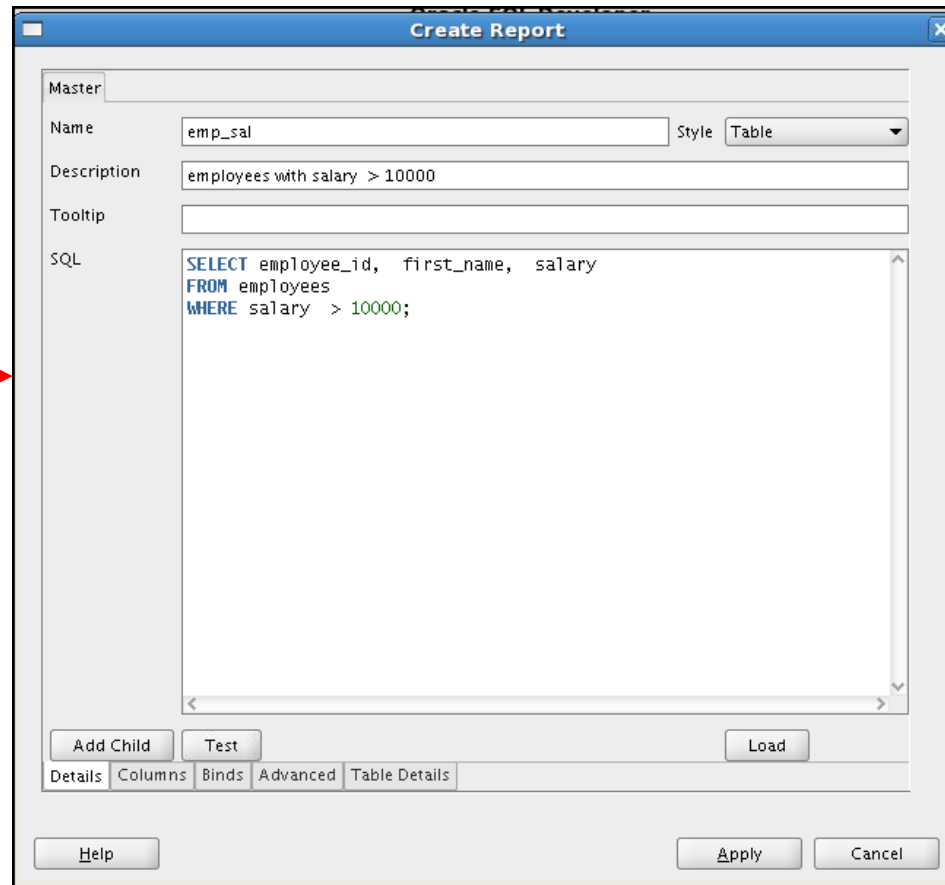
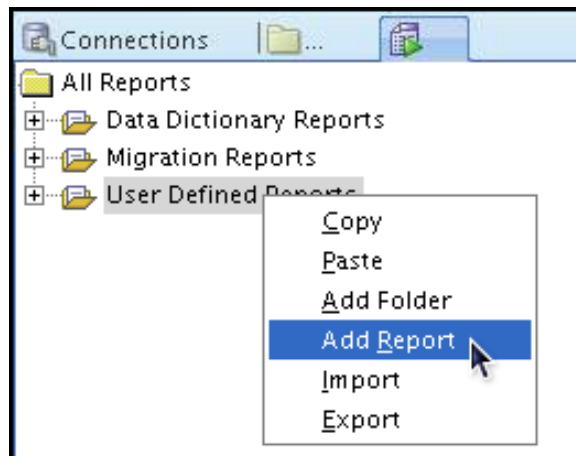
The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane is open, and the 'Dependencies' report is selected under 'Data Dictionary Reports'. The main pane displays a table of database dependencies for the 'myconnection' connection. The table has columns for Owner, Name, Type, Referenced Owner, and Referenced Name. The 'Dependencies' report icon in the left pane is highlighted with a red box.

Owner	Name	Type	Referenced Owner	Referenced Name
CTXSYS	CTX_CLASSES	VIEW	CTXSYS	DR\$CLASS
CTXSYS	CTX_CLS	PACKAGE	SYS	STANDARD
CTXSYS	CTX_DOC	PACKAGE	SYS	STANDARD
CTXSYS	CTX_INDEX_SETS	VIEW	CTXSYS	DR\$INDEX_SET
CTXSYS	CTX_INDEX_SETS	VIEW	SYS	USER\$
CTXSYS	CTX_INDEX_SET_INDEXES	VIEW	CTXSYS	DR\$INDEX_SET
CTXSYS	CTX_INDEX_SET_INDEXES	VIEW	CTXSYS	DR\$INDEX_SET_INDEX
CTXSYS	CTX_INDEX_SET_INDEXES	VIEW	SYS	USER\$
CTXSYS	CTX_OBJECTS	VIEW	CTXSYS	DR\$CLASS
CTXSYS	CTX_OBJECTS	VIEW	CTXSYS	DR\$OBJECT
CTXSYS	CTX_OBJECT_ATTRIBUTES	VIEW	CTXSYS	DR\$CLASS
CTXSYS	CTX_OBJECT_ATTRIBUTES	VIEW	CTXSYS	DR\$OBJECT
CTXSYS	CTX_OBJECT_ATTRIBUTES	VIEW	CTXSYS	DR\$OBJECT_ATTRIBUTE
CTXSYS	CTX_OBJECT_ATTRIBUTE_LOV	VIEW	CTXSYS	DR\$CLASS
CTXSYS	CTX_OBJECT_ATTRIBUTE_LOV	VIEW	CTXSYS	DR\$OBJECT
CTXSYS	CTX_OBJECT_ATTRIBUTE_LOV	VIEW	CTXSYS	DR\$OBJECT_ATTRIBUTE
CTXSYS	CTX_OBJECT_ATTRIBUTE_LOV	VIEW	CTXSYS	DR\$OBJECT_ATTRIBUTE_LOV
CTXSYS	CTX_PARAMETERS	VIEW	CTXSYS	DR\$PARAMETER



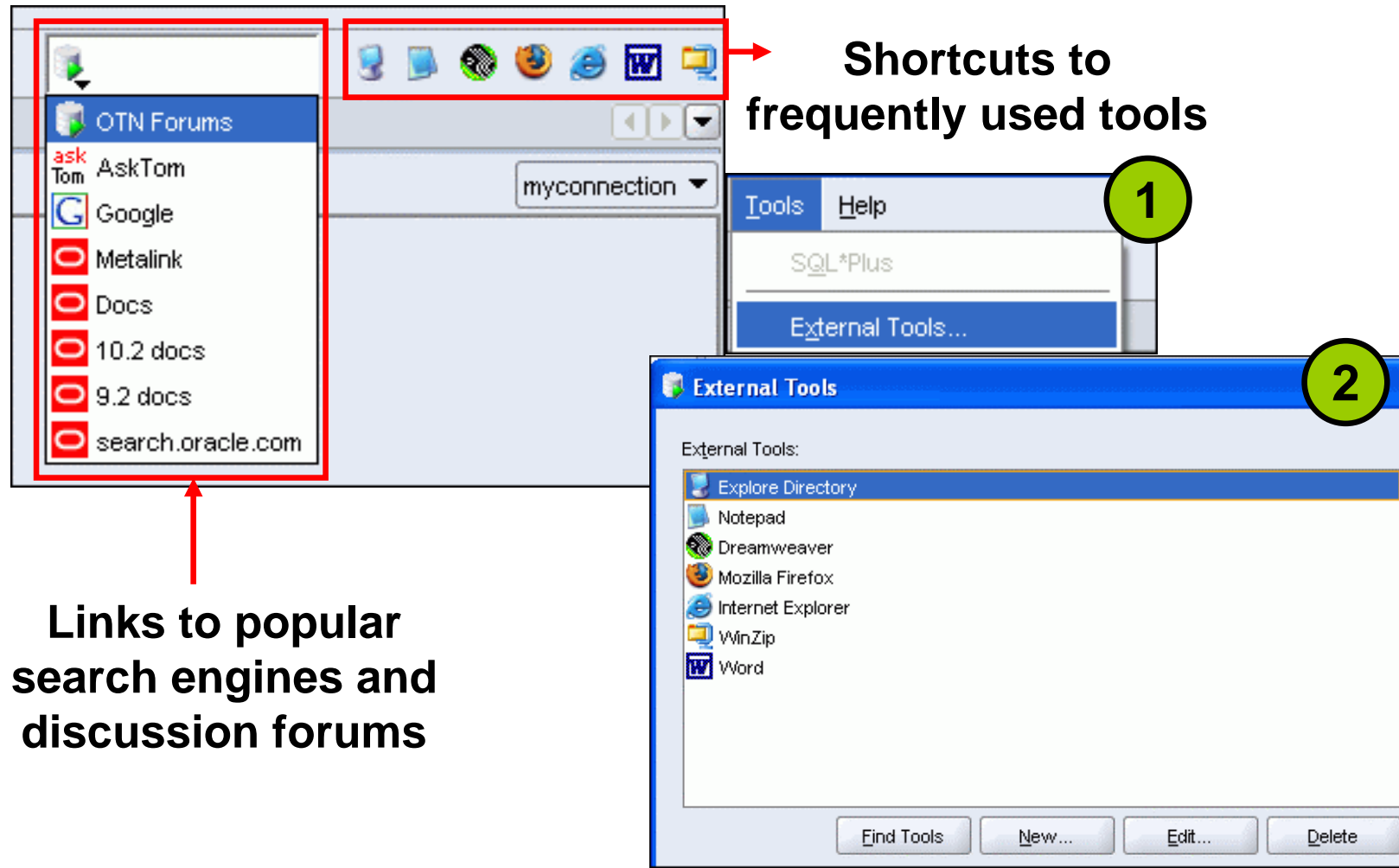
# Creating a User-Defined Report

Create and save user-defined reports for repeated use.



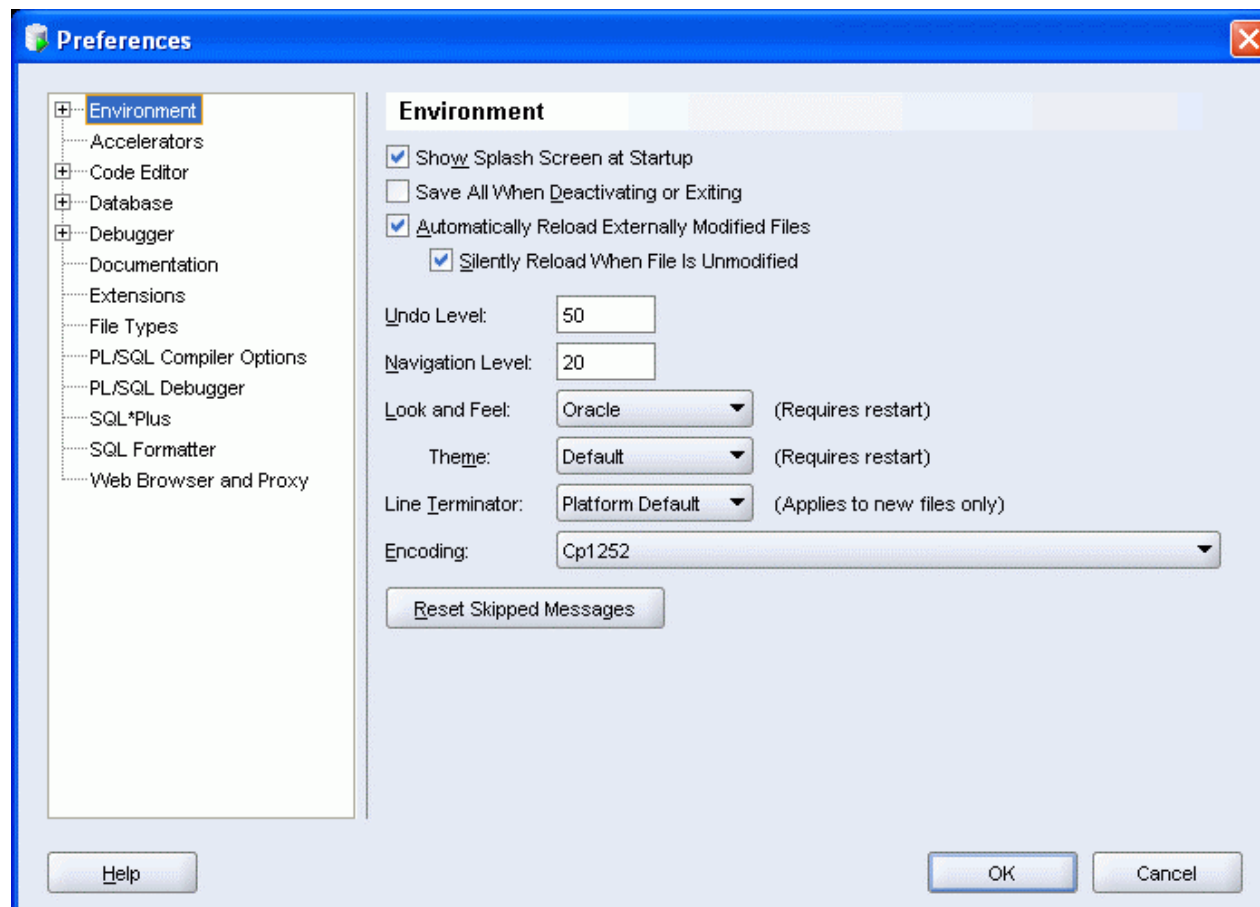
**Organize reports in folders.**

# Search Engines and External Tools

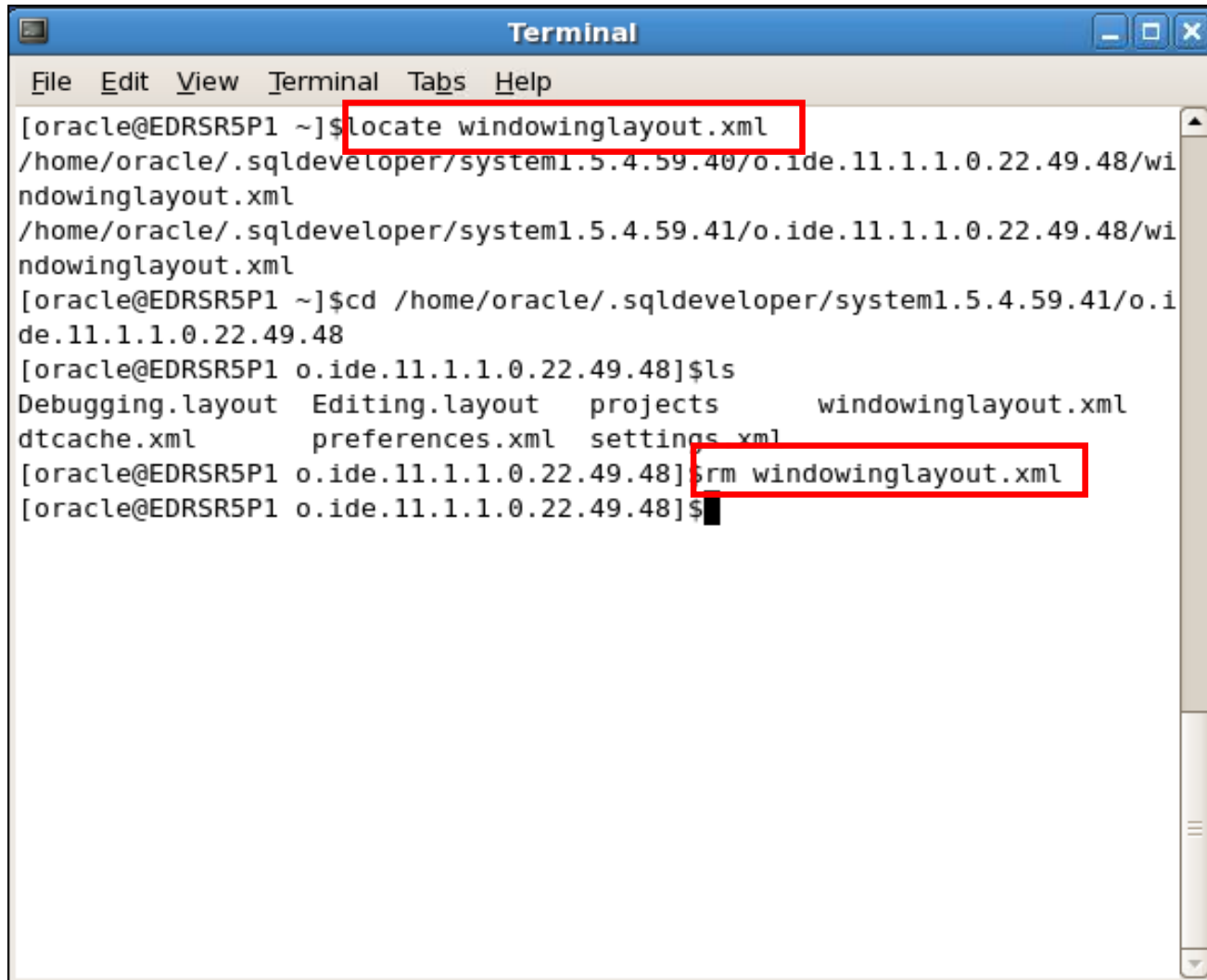


# Setting Preferences

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



# Resetting the SQL Developer Layout

A screenshot of a terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a series of commands and their outputs. Two red rectangles highlight the commands `locate windowinglayout.xml` and `rm windowinglayout.xml`.

```
[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]$
```

# 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