Testing and Debugging Procedures with SQL Developer

Before You Begin

Purpose

This tutorial covers how to execute a DDL script, review changes to the database objects, create, execute, test and debug a procedure.

Time to Complete

Approximately 60 minutes

Introduction

Oracle SQL Developer is a free and fully supported graphical tool that enhances productivity and simplifies database development tasks. Using SQL Developer, users can browse, edit and create database objects, run SQL statements, edit and debug PL/SQL statements, build PL/SQL unit tests, run reports, and place files under version control.

In this tutorial, you use SQL Developer Release 4.1 to examine various tasks.

Prerequisites

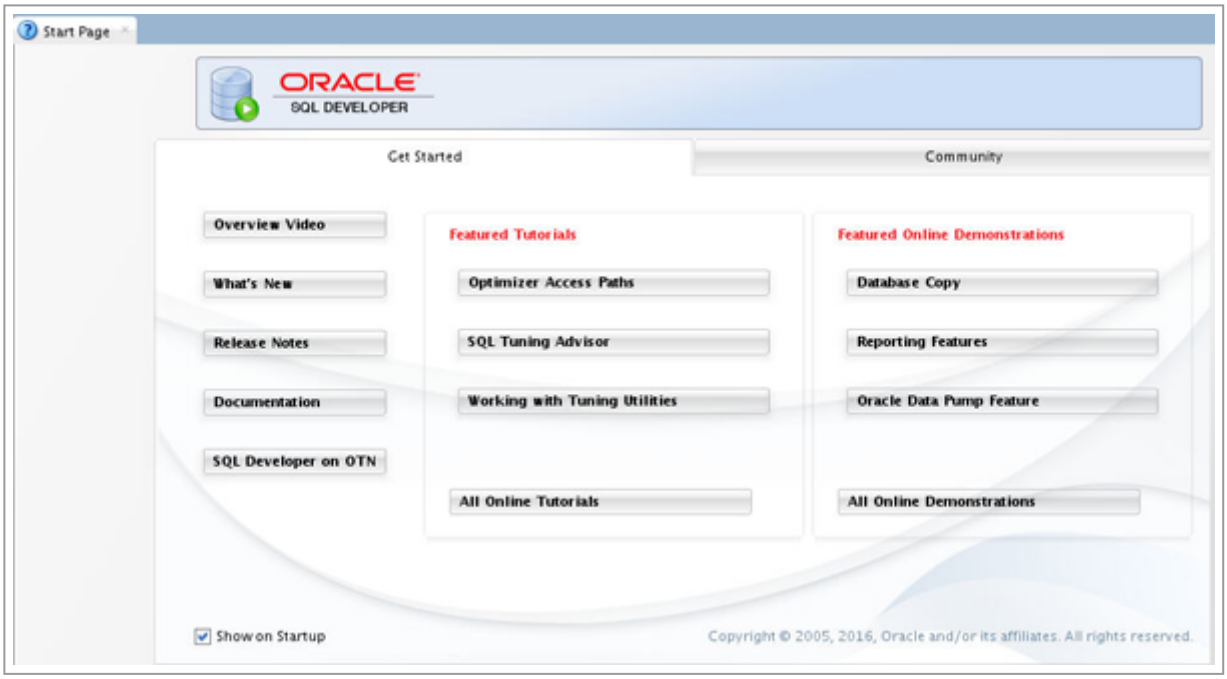
Before starting this tutorial, you should:

- Have installed Oracle SQL Developer Release 4.1 or above
- Have access to an Oracle Database 11g database that has the sample schema installed.
- Grant HR user `DEBUG CONNECT SESSION` and `DEBUG ANY PROCEDURE` privileges.
- Performed the Re-engineering Your Database Using Oracle SQL Developer Data Modeler 4.1 (https://apexapps.oracle.com/pls/apex/f?p=44785:112:0::::P112_CONTENT_ID:19004) tutorial.
- Downloaded and unzipped the files.zip (files/files.zip) into your working directory.

Creating a Database Connection

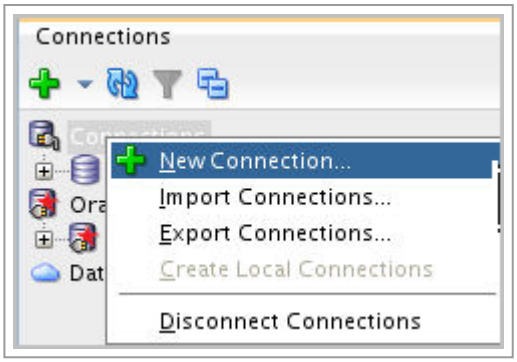
In this topic, you will create a database connection to the HR Schema in SQL Developer:

1. Double click on the SQL Developer icon on the Desktop.
2. The first time SQL Developer is open, the "Start Page" is displayed. You can deselect the **Show on Startup** check box to turn it off.



Description of this image (files/tab_01_01.txt)

3. In the **Connections** tab, right click Connections and select **New Connection**

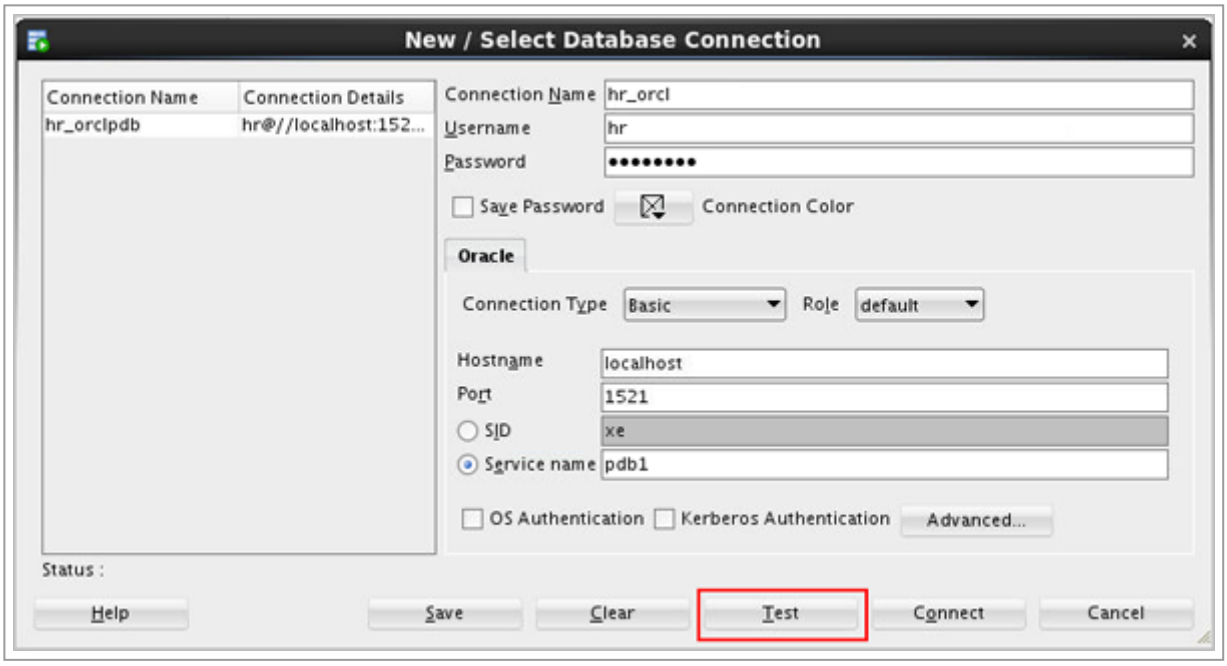


Description of this image (files/tab01_02.txt)

4. Instructions and result (including collapsible image with text file for accessibility):

   A New / Update Database Connection dialog opens. Enter the connection details as follows and click **Test**.

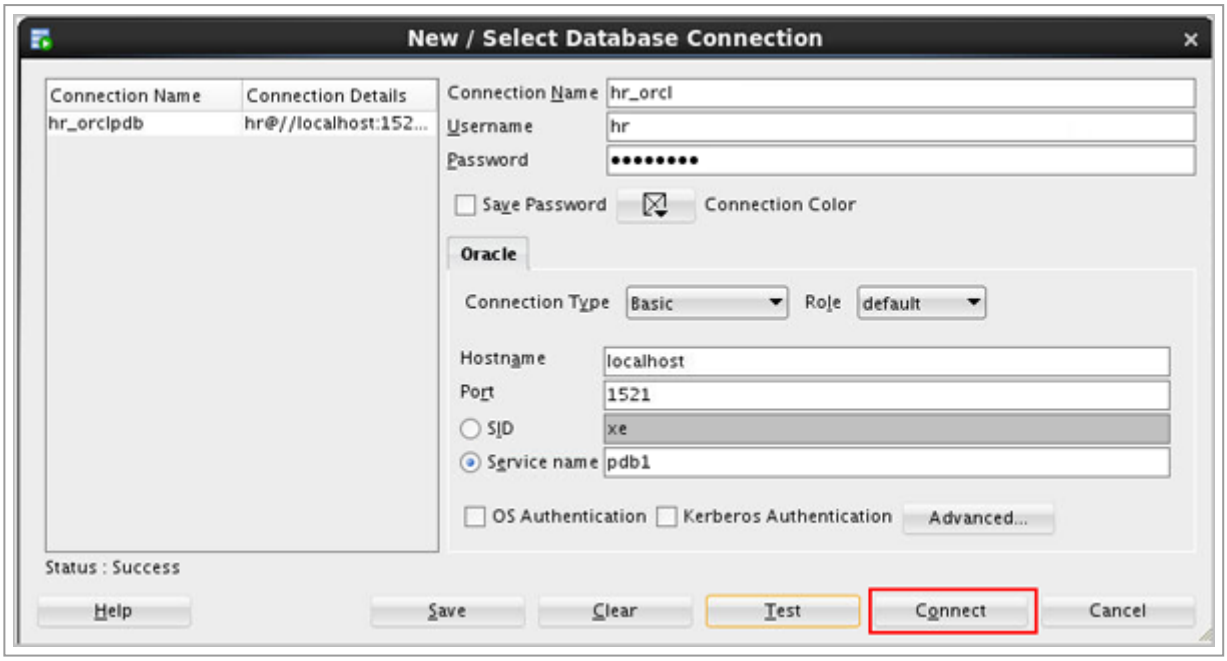
   Connection Name: **hr_orcl**
   User Name: **hr**
   Password: **<your password>** (Select the **Save Password** checkbox)
   SID: **<your own SID>**



Description of this image (files/tab01_03.txt)

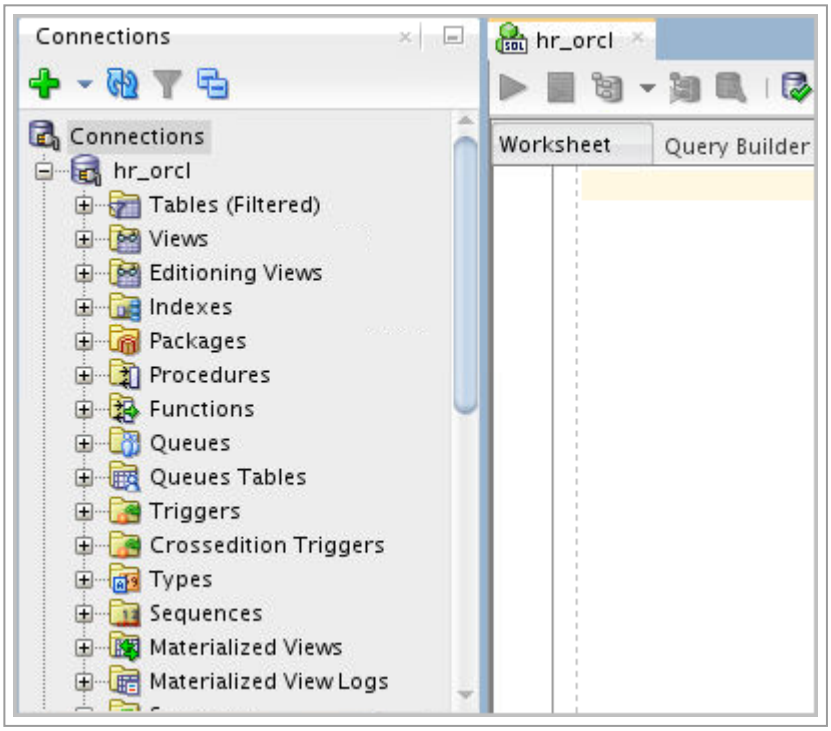
Note: In this tutorial the Service Name is specified instead of SID.

5. The status of your test is 'Success'. Click **Connect** in the New/ Update Database Connection dialog to create the connection.



Description of this image (files/tab01_04.txt)

6. Expand the **hr_orcl** connection. Notice all the object types. Expand **Tables**. In the next section, you examine the objects currently in the hr schema.

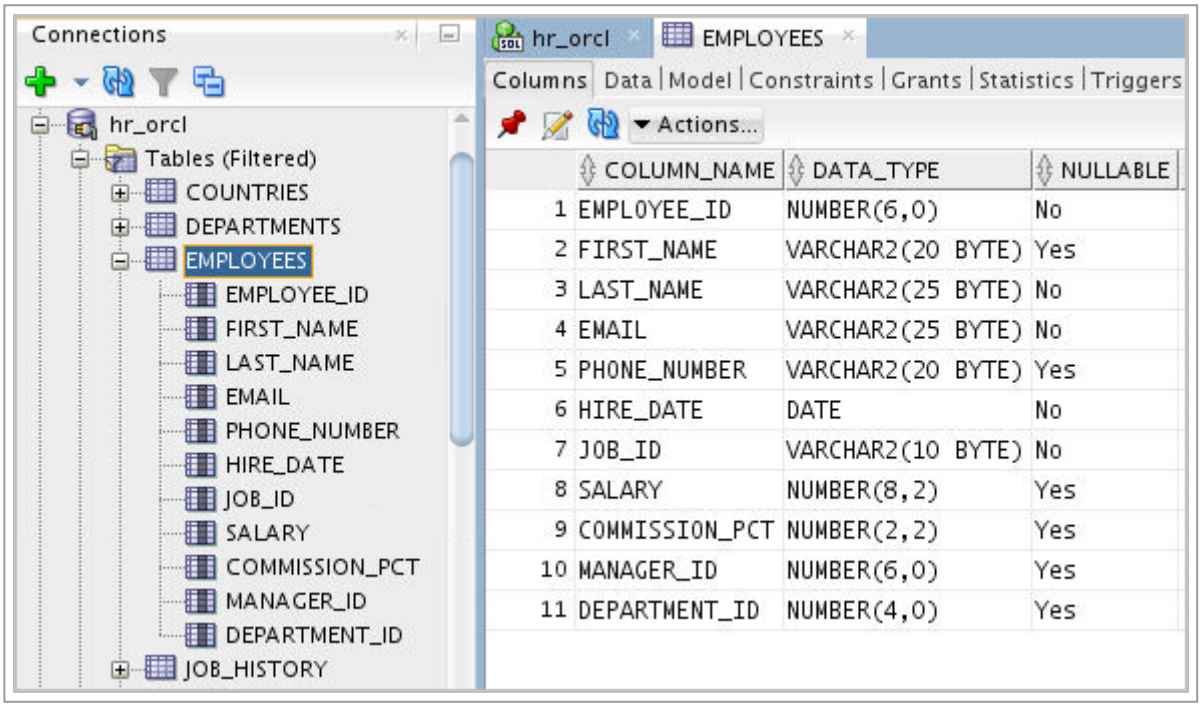


Description of this image (files/tab01_05.txt)

Review Existing Objects in the HR Schema

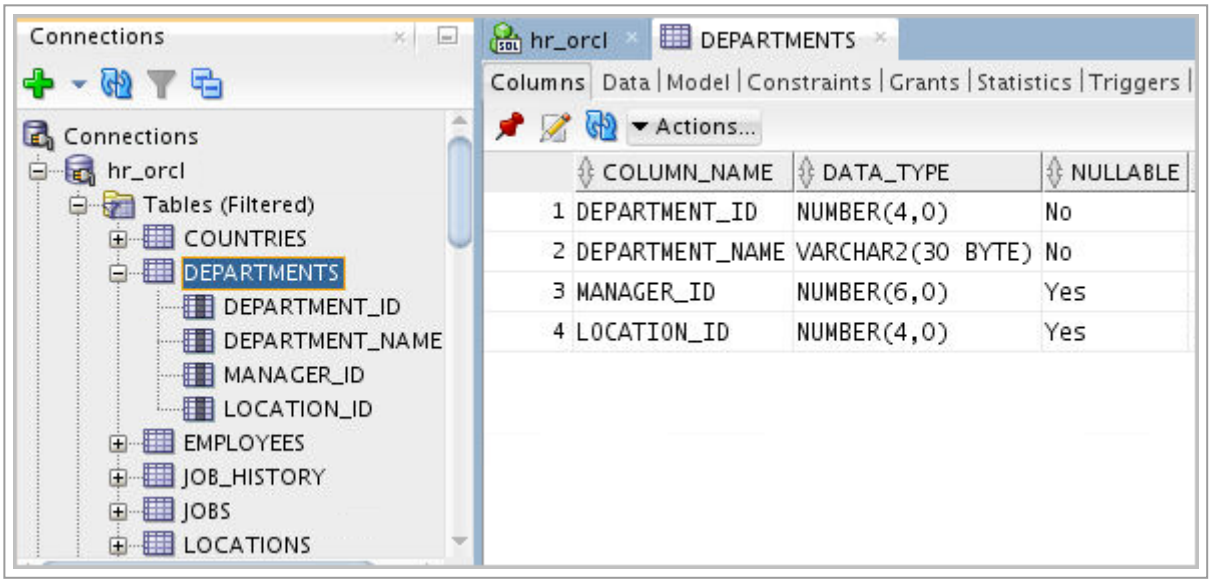
In this topic, you review the existing objects in the hr schema.

1. Expand the **EMPLOYEES** table. Notice that the column definitions are listed.



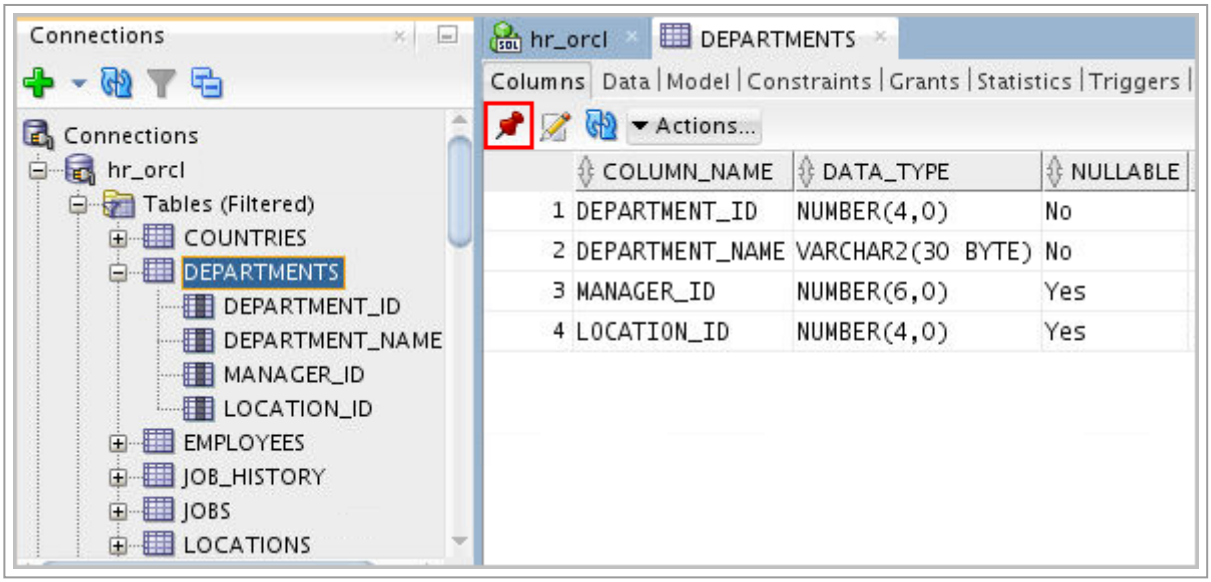
Description of this image (files/tab02_01.txt)

2. Click the **DEPARTMENTS** table in the navigator.



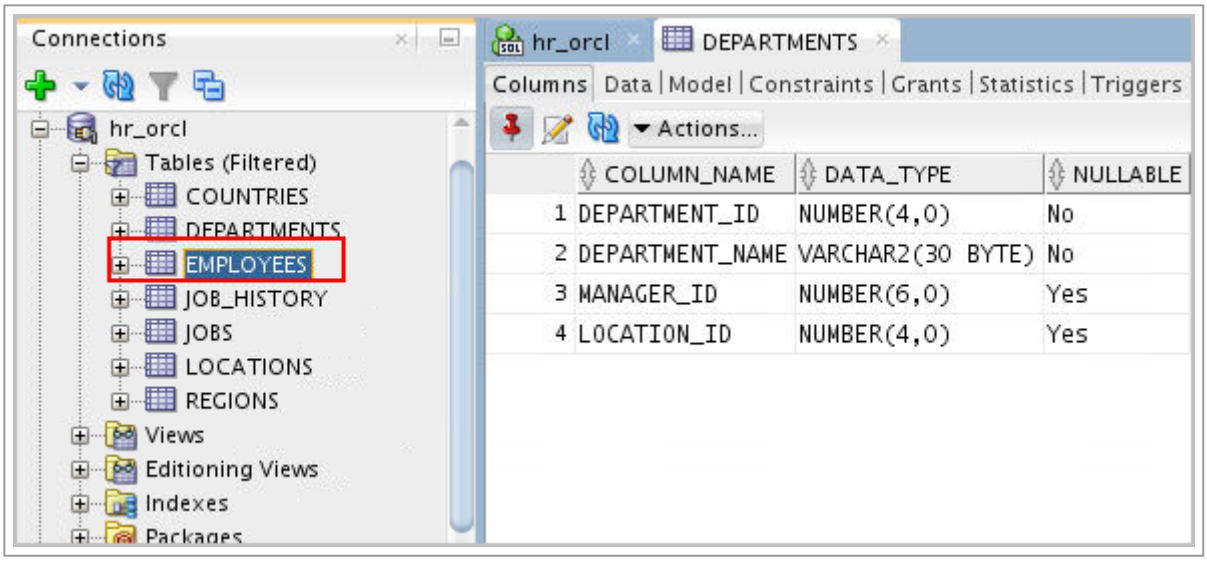
Description of this image (files/tab02_02.txt)

3. Notice that the information in the `EMPLOYEES` tab was replaced by the `DEPARTMENTS` table information. If you want the table information in the tab to remain frozen, select the **Pin** icon to Freeze the pane.



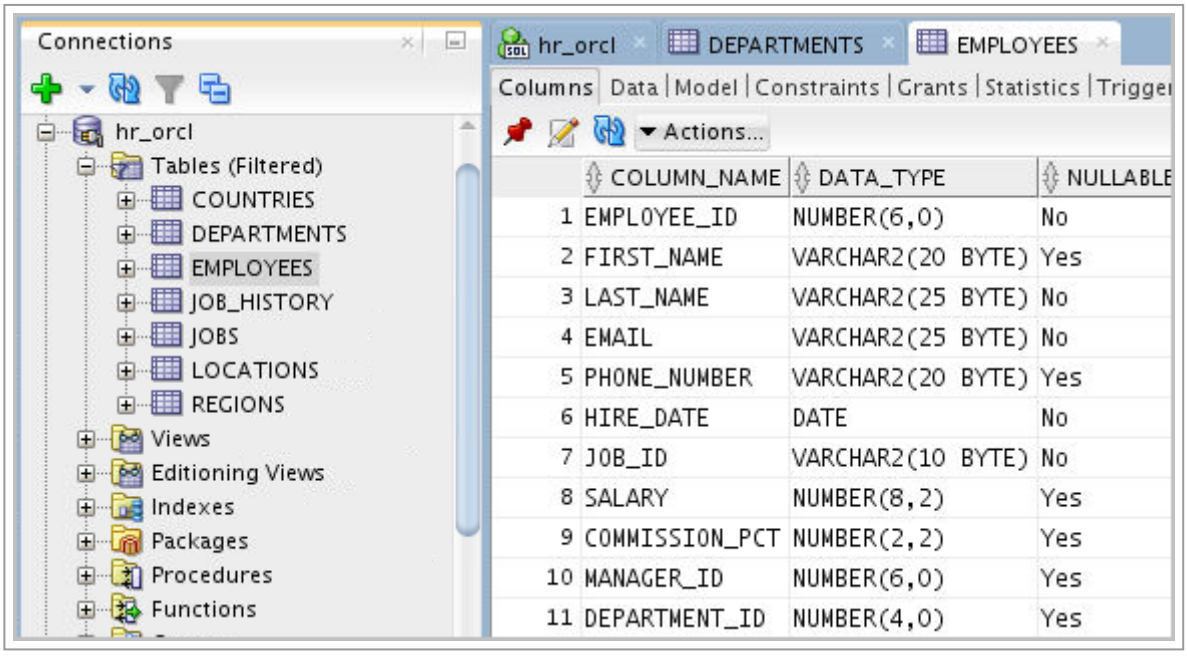
Description of this image (files/tab02_03.txt)

4. Then click the **`EMPLOYEES`** table again in the navigator.



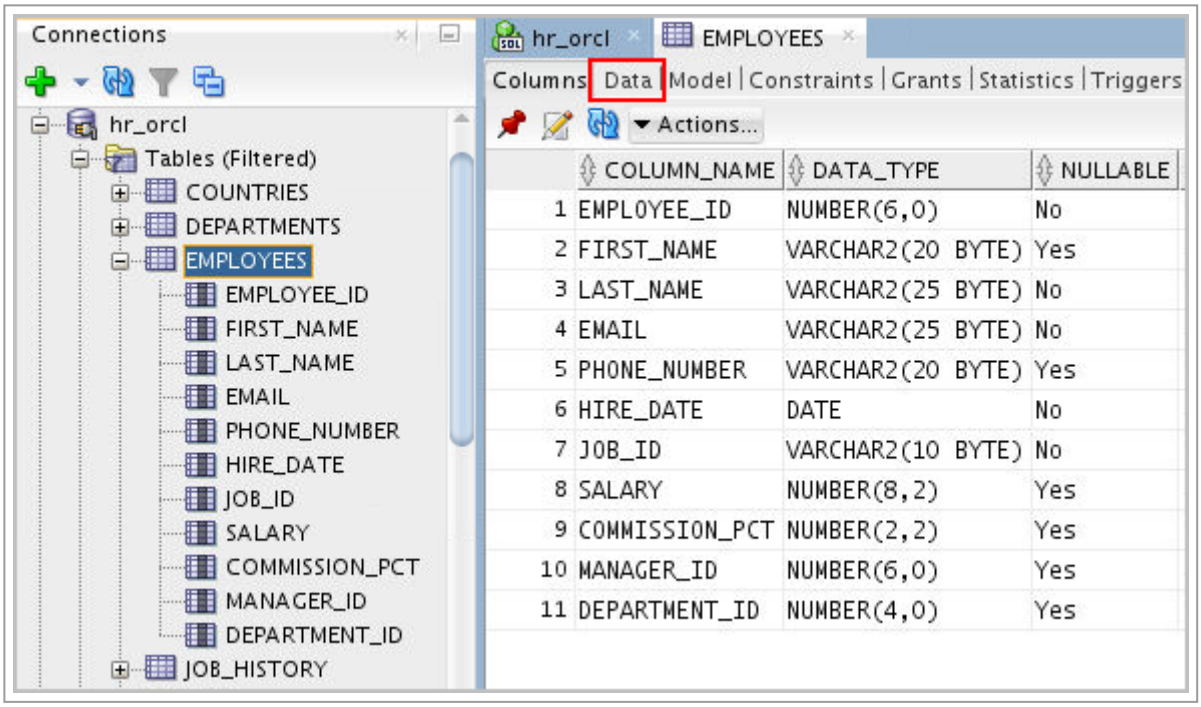
Description of this image (files/tab02_04.txt)

5. Notice this time you have 2 tabs, one for each of the tables because the `DEPARTMENTS` table pane is frozen.



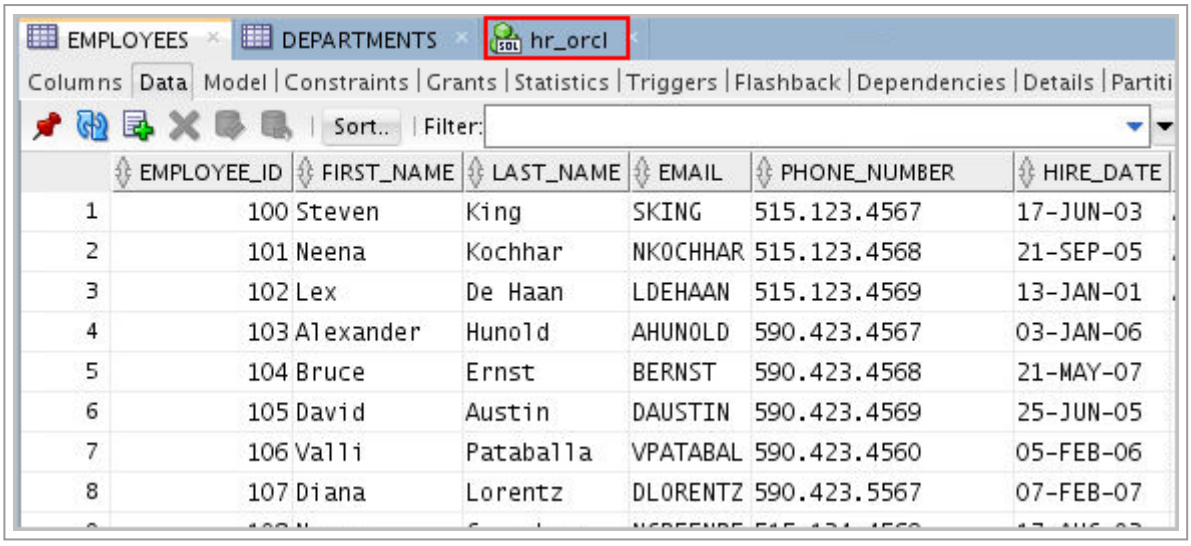
Description of this image (files/tab02_05.txt)

6. You can see the data in the `EMPLOYEES` table. Click the **Data** subtab.



Description of this image (files/tab02_06.txt)

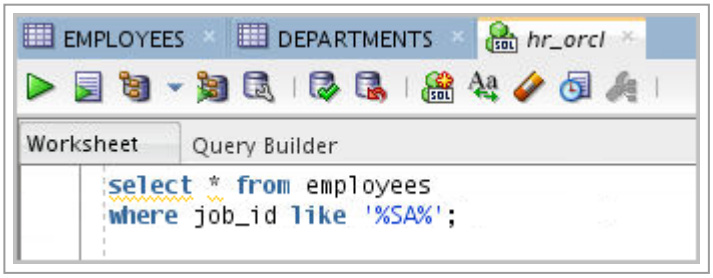
7. The data in the `EMPLOYEES` table is displayed. You can also enter a SQL statement in the SQL Worksheet. Click the **hr_orcl** tab.



Description of this image (files/tab02_07.txt)

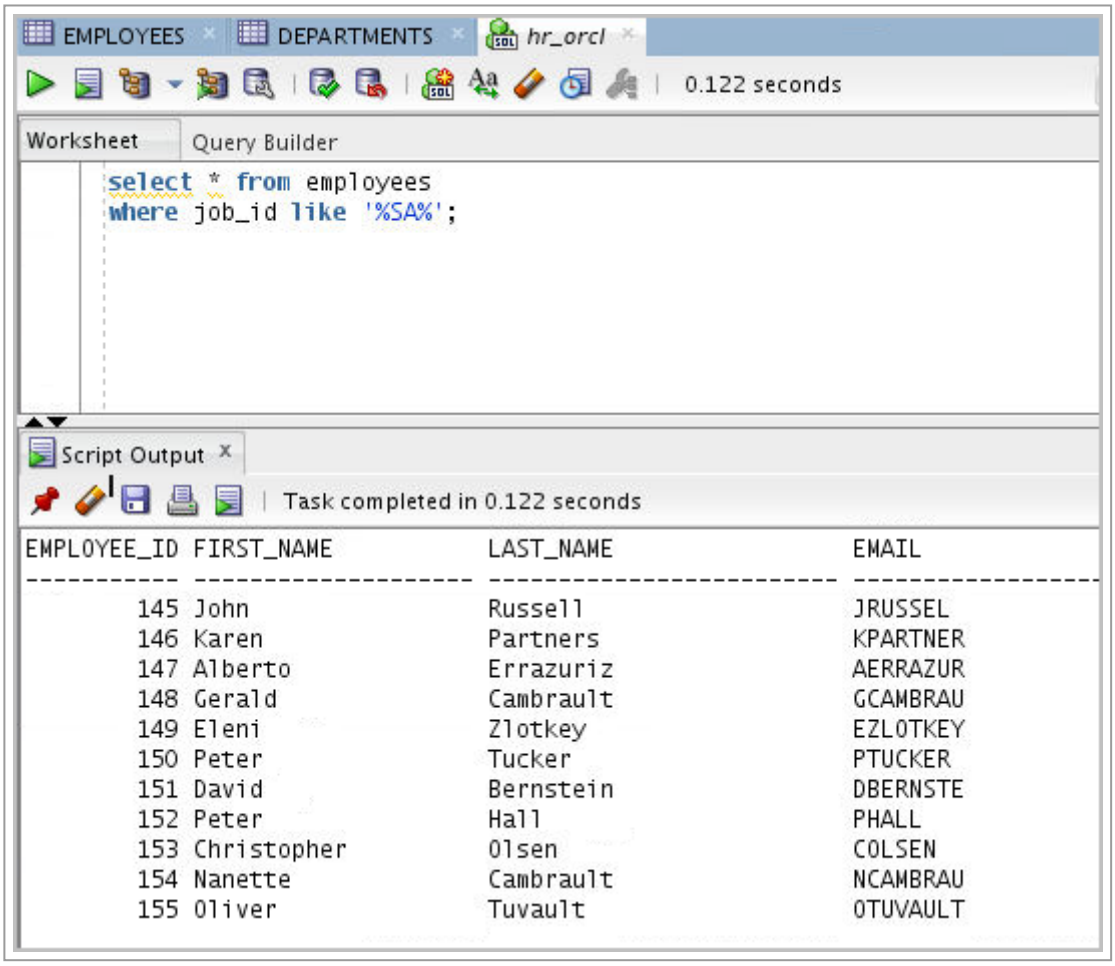
8. Enter the following SQL statement and select the Execute SQL Statement icon.

```
select * from employees
where job_id like '%SA%';
```



Description of this image (files/tab02_08.txt)

9. The Query Results are displayed. In the next topic, you run the script you generated in the previous tutorial on Data Modeler.

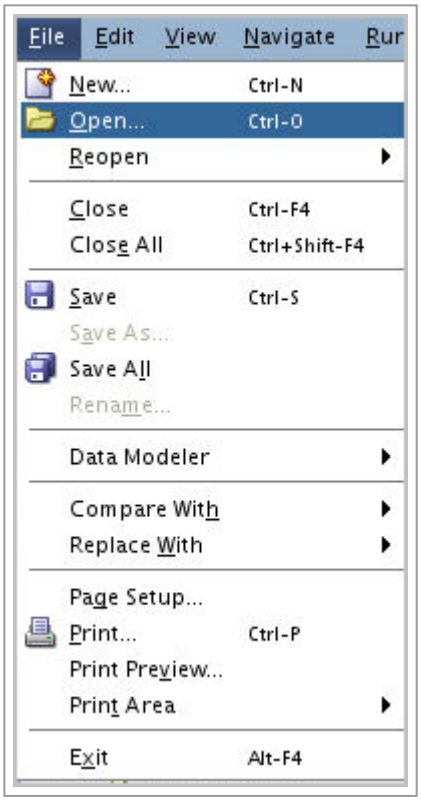


Description of this image (files/tab02_09.txt)

Executing a DDL Script

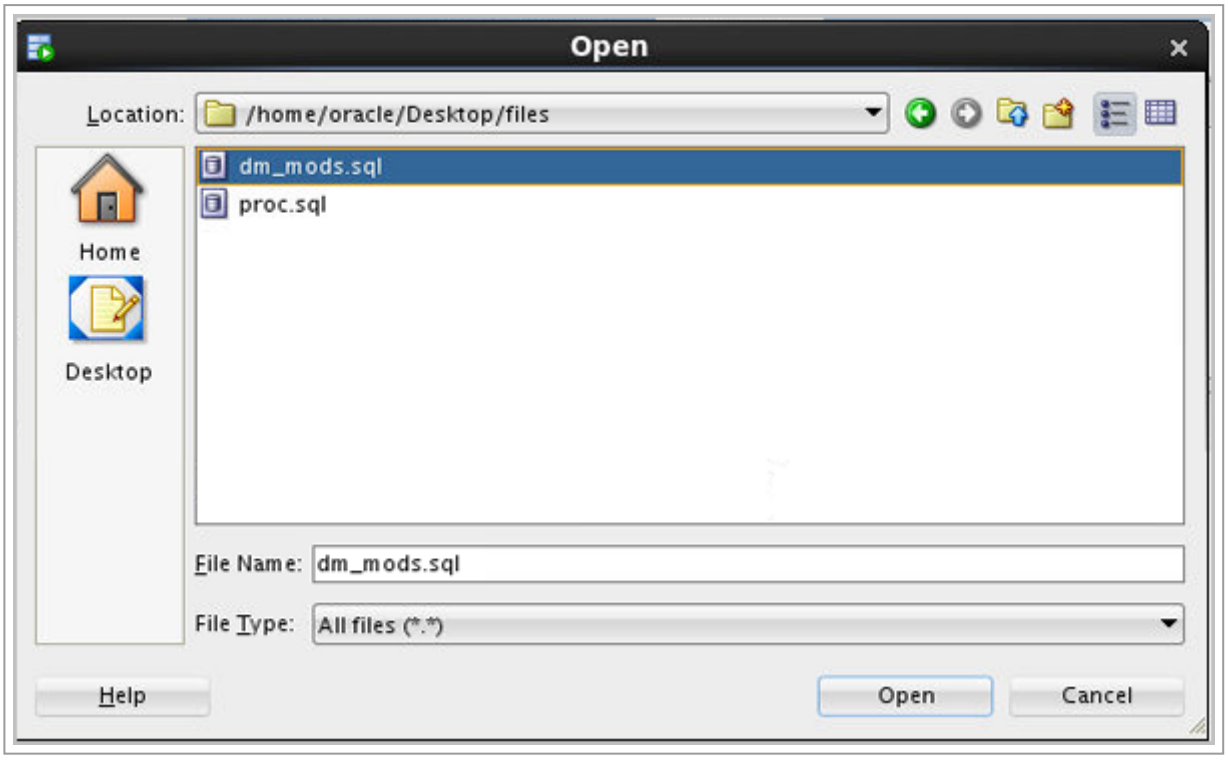
In this topic, you execute the DDL script you generated in the Data Modeler tutorial. If you did not complete the previous tutorial, you can access the solution using the **dm_mods.sql** in the files folder**.**

1. Select **File > Open**.



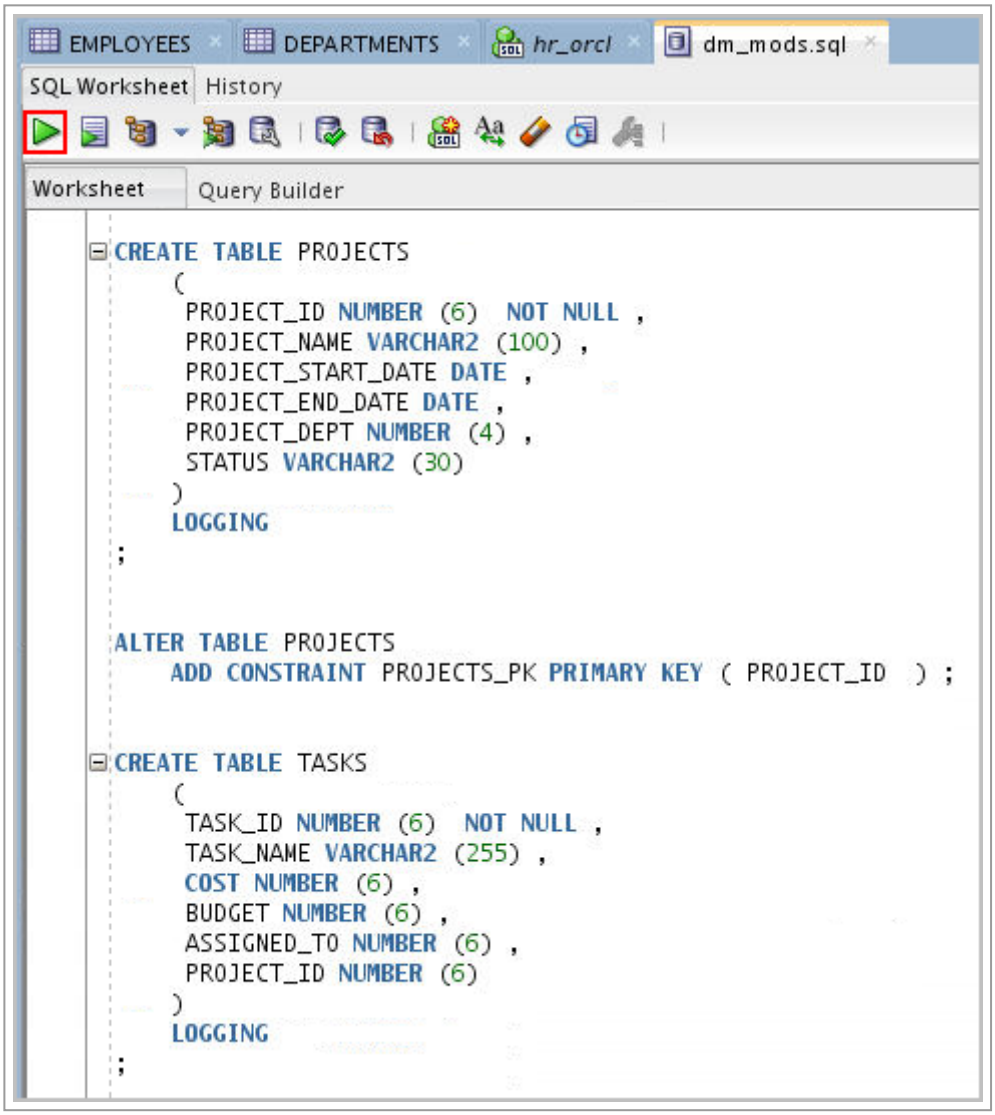
Description of this image (files/tab03_01.txt)

2. Locate the **dm_mods.sql file** and click **Open**.



Description of this image (files/tab03_02.txt)

3. This SQL file contains all the DDL to change the HR Schema so that it is synchronized with the model changes you made in the previous tutorial. When you execute this script, the `PROJECTS` and `TASKS` tables will be created, and the new `COST_CENTER` column will be added to the `DEPARTMENTS` table. Scroll down to review the DDL.

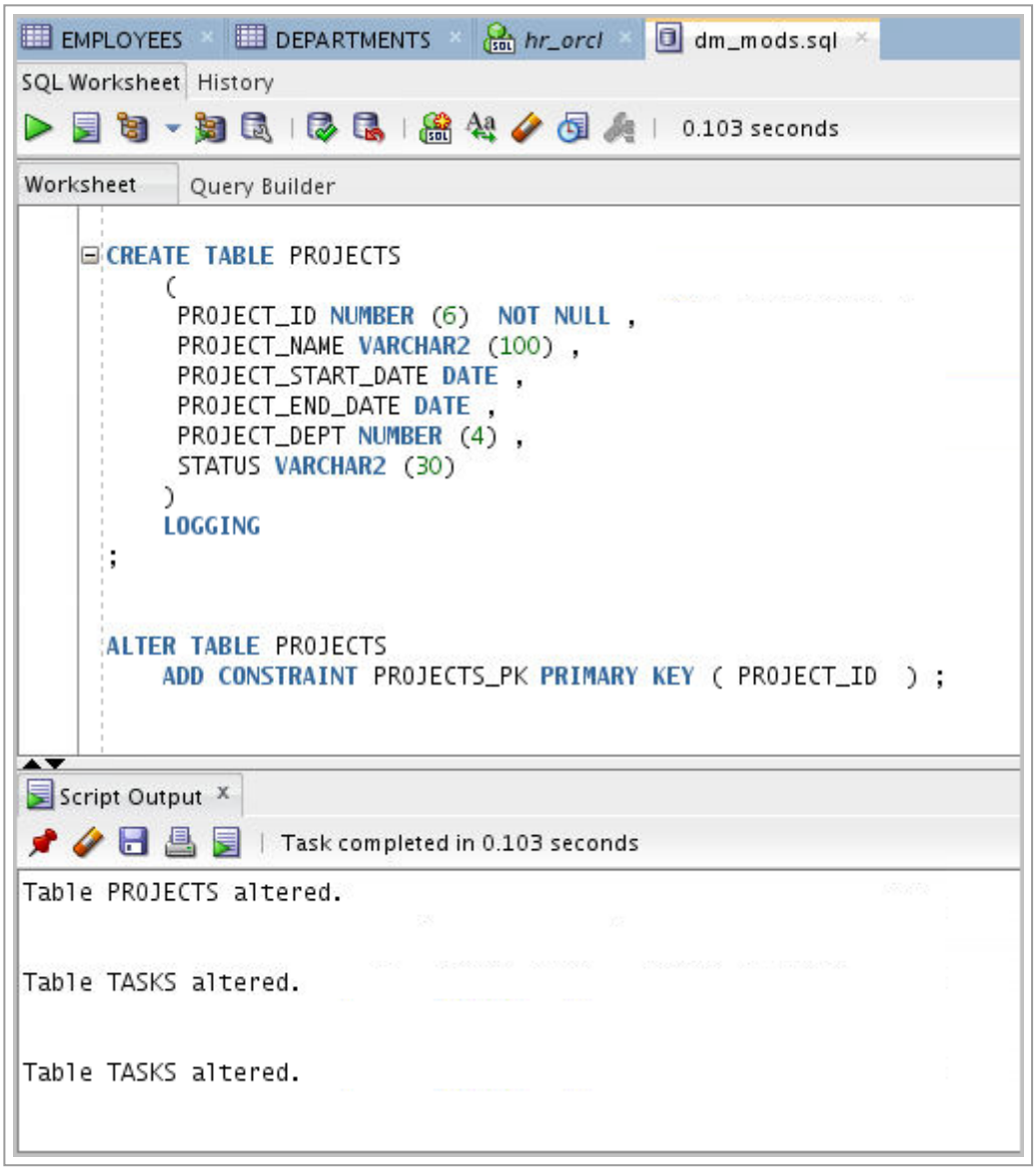


Description of this image (files/tab03_03.txt)

4. Select the **hr_orcl** connection from the list and click **OK**.

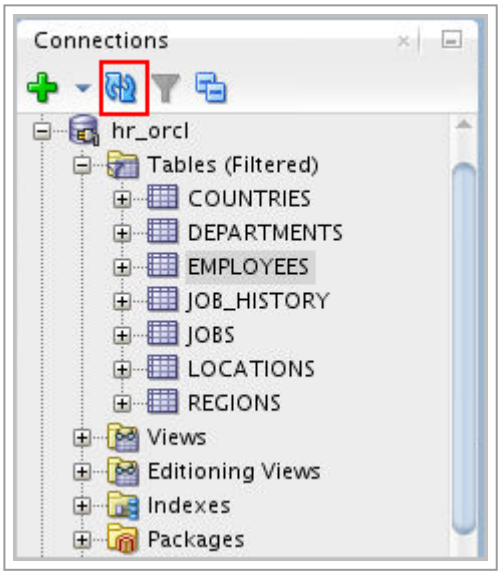
Description of this image (files/tab03_04.txt)

5. All the statements in the DDL script executed successfully



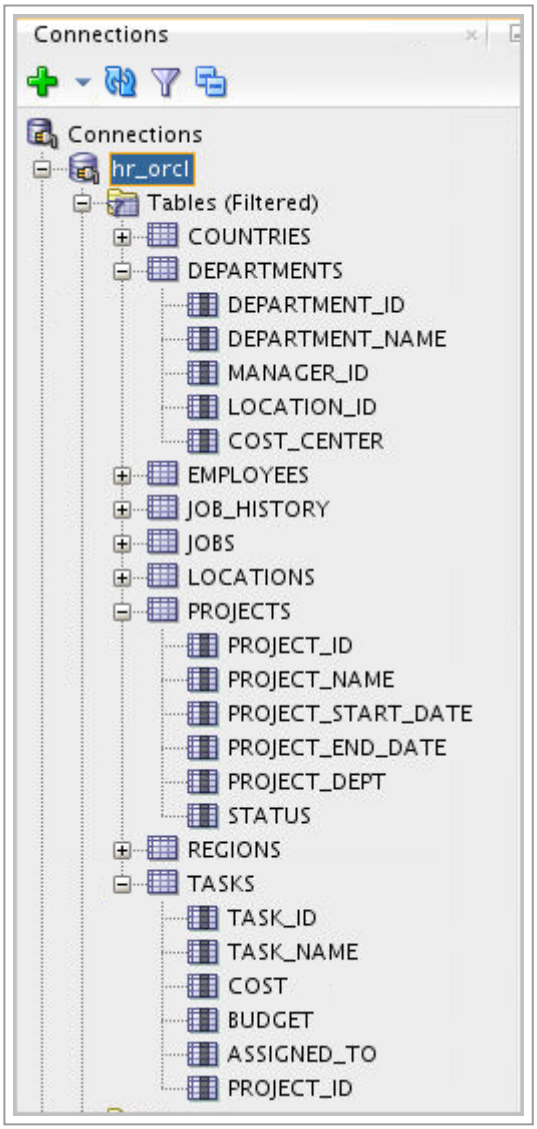
Description of this image (files/tab03_05.txt)

6. Click the **Refresh** icon to refresh the list of tables.



Description of this image (files/tab03_06.txt)

7. Notice that the new tables `PROJECTS` and `TASKS` are contained in the list. Expand the **DEPARTMENTS**, **PROJECTS** and **TASKS** table nodes and review the results. In the next topic, you create a procedure and run it.



Description of this image (files/tab03_07.txt)

Creating and Executing a Procedure

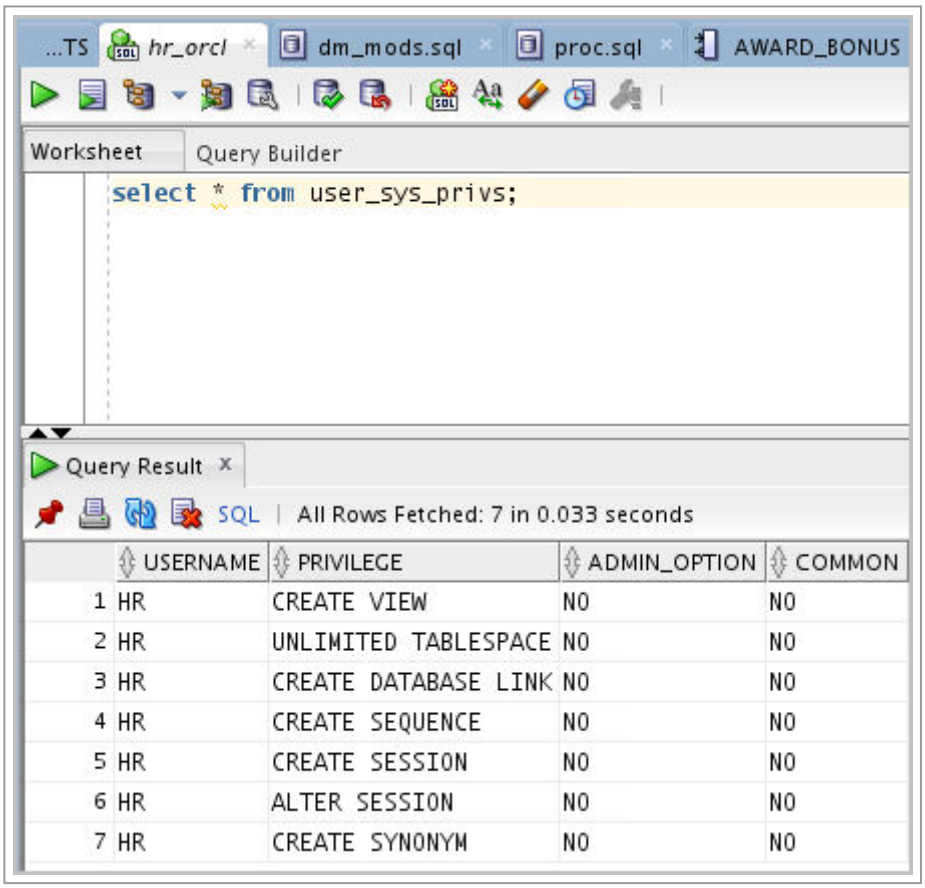
Debugging a Procedure

The procedure created in the earlier section was created with an error. You can locate errors in the code by debugging the code. You have to run a script before you actually start the debug process

Before you Debug

In order to debug a sub program you should have **DEBUG** privileges.

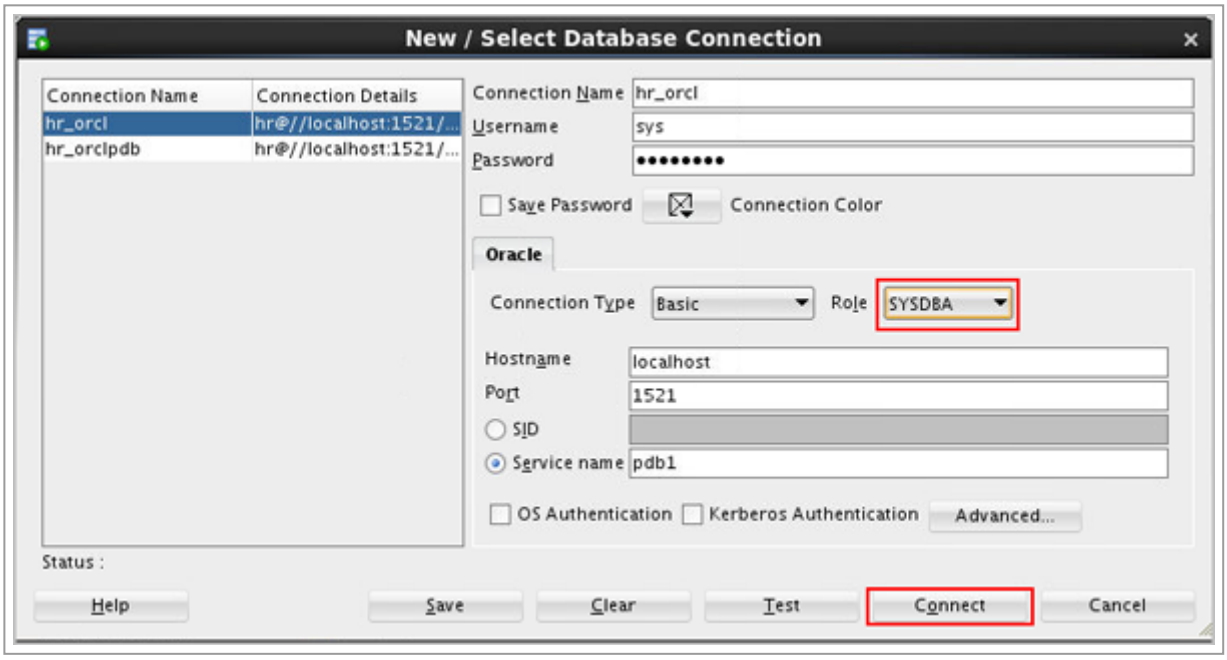
1. To check the privileges you can execute a SQL statement



Description of this image (files/tab05_01_01.txt)

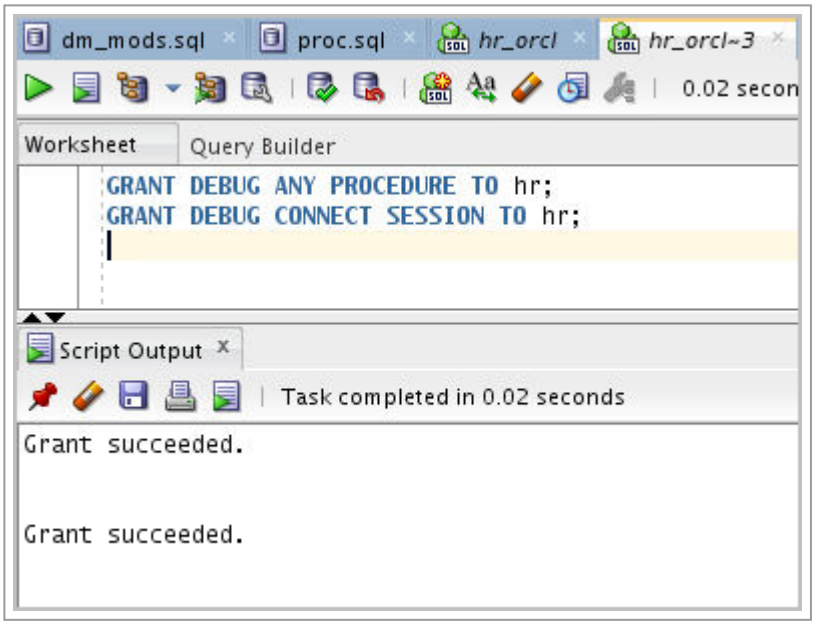
You can see that the hr user doesn't have **DEBUG CONNECT** and **DEBUG** privileges

2. If you don't have the required **DEBUG** privileges, a **SYSDBA** role user has to assign them. Login as a **SYSDBA** user

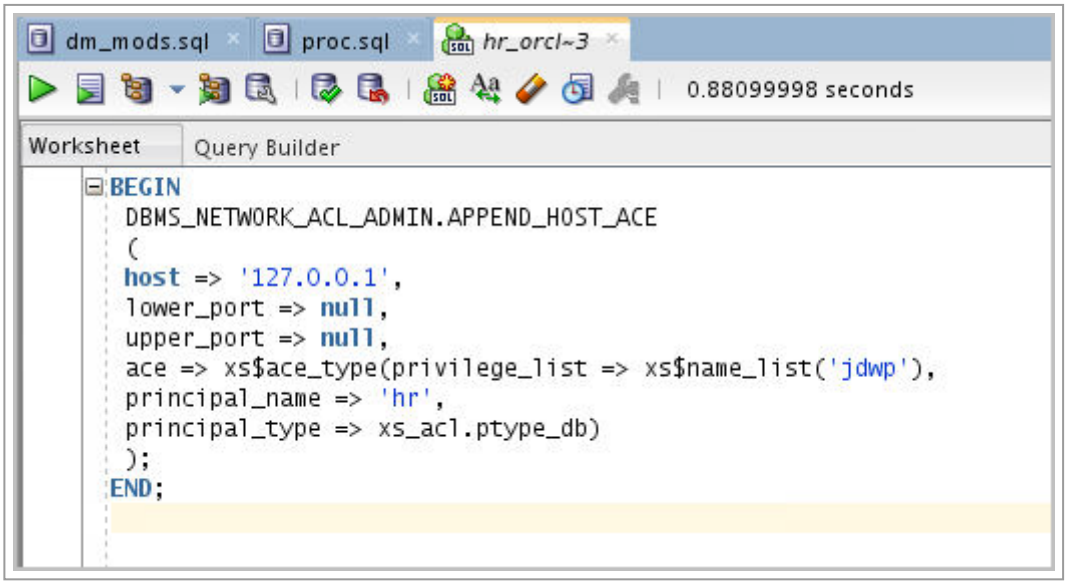


Description of this image (files/tab05_01_02.txt)

3. Execute the grant commands and ACL(Access Control List) script shown as a SYSDBA user

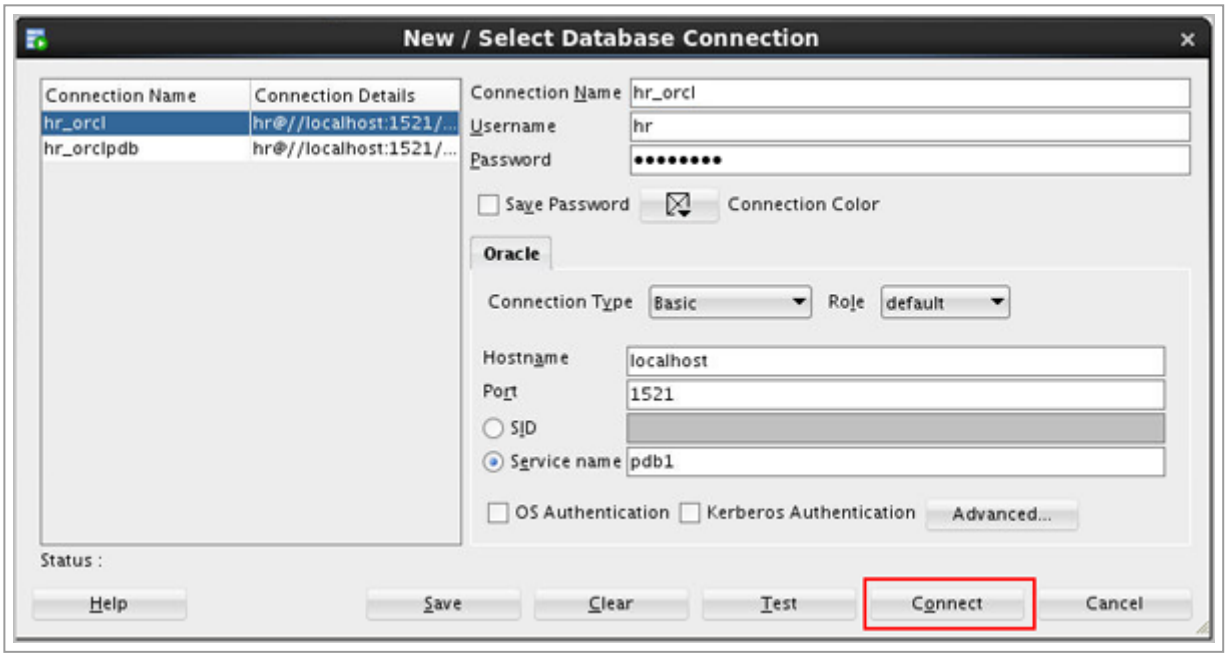


Description of this image (files/tab05_01_03-2.txt)



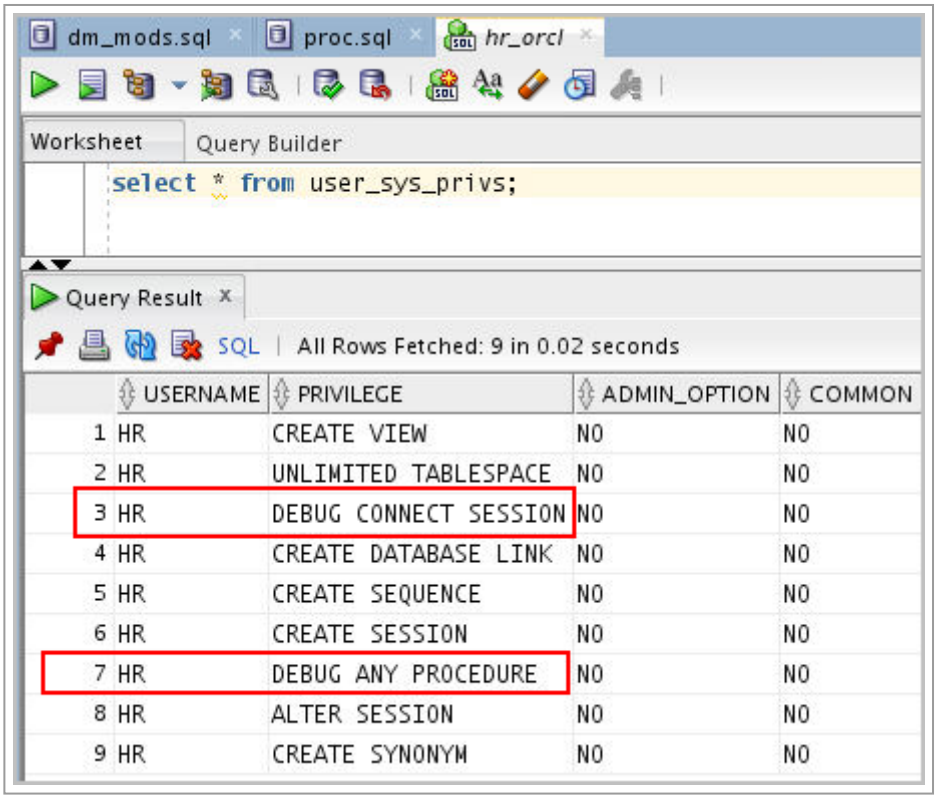
Description of this image (files/tab05_01_03.txt)

4. Now login as hr user, who has a non-sysdba user role



Description of this image (files/tab05_01_04.txt)

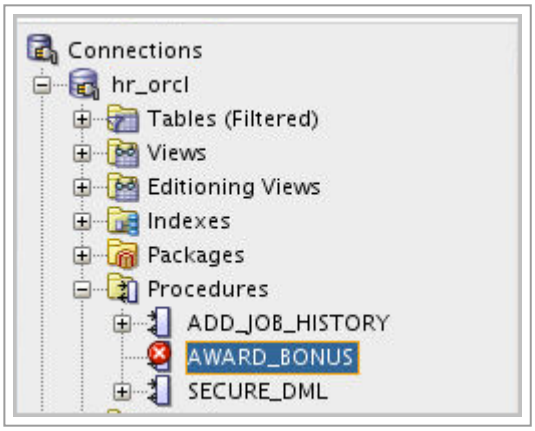
5. Execute the SQL statement shown to check whether required privileges are granted to hr user



Description of this image (files/tab05_01_05.txt)

Debugging

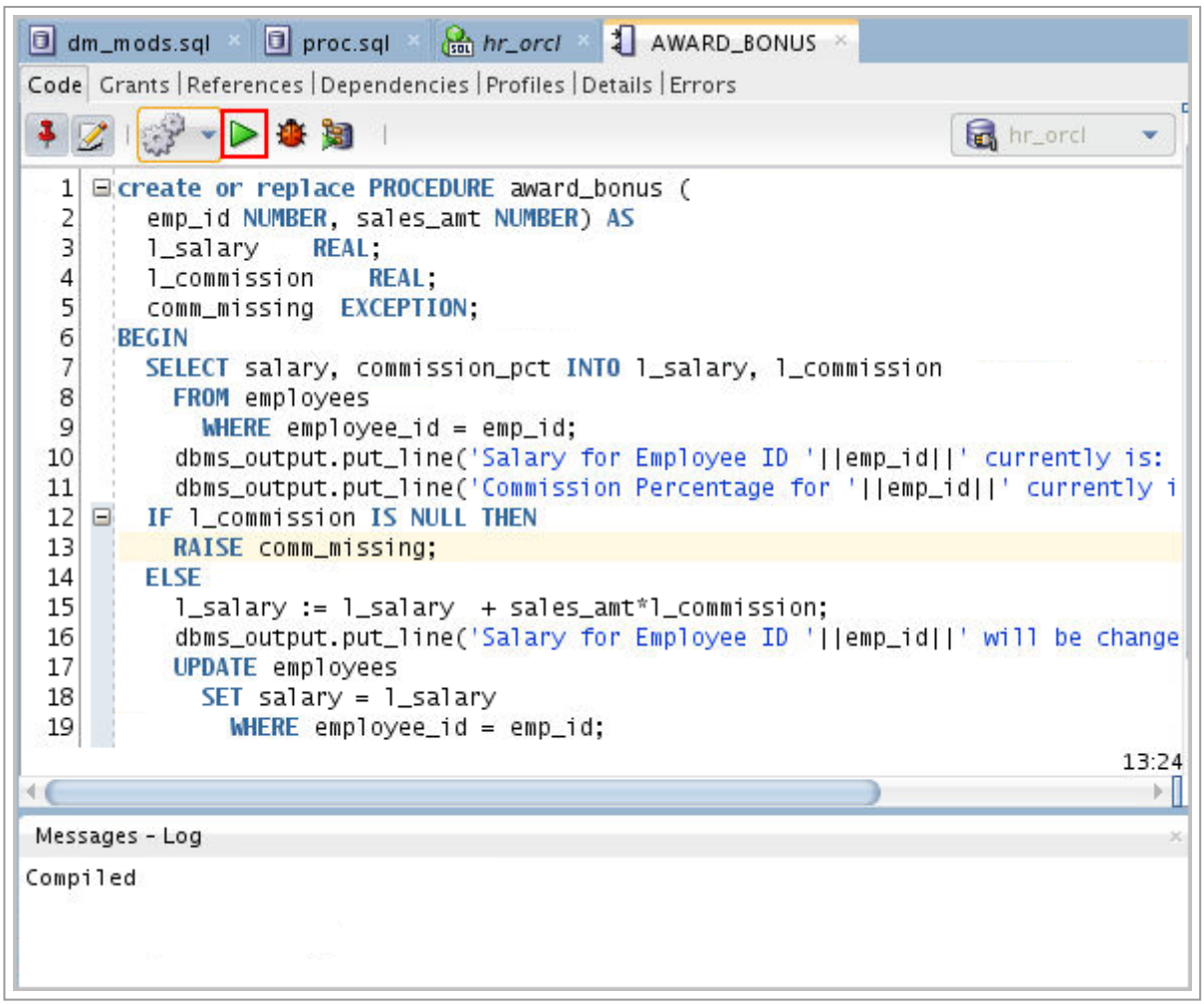
1. Now open the AWARD_BONUS procedure you created earlier. Compile the procedure



Description of this image (files/tab05_02_01.txt)

You can see the error message in the compiler log. It specifies a line number where the error occurred

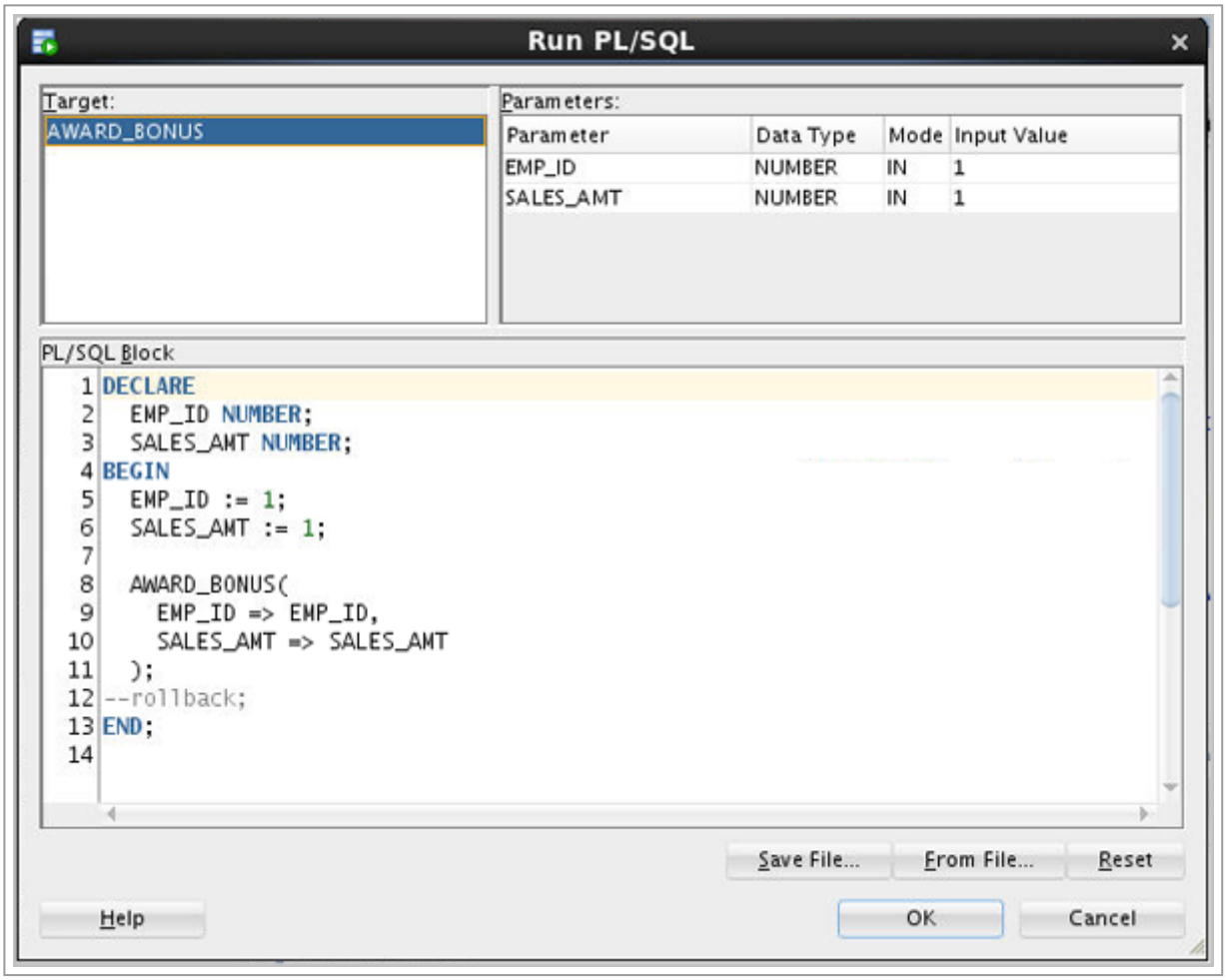
2. Modify the code in line 13 by adding a semi colon. Select the **Compile** icon.



Description of this image (files/tab05_02_05.txt)

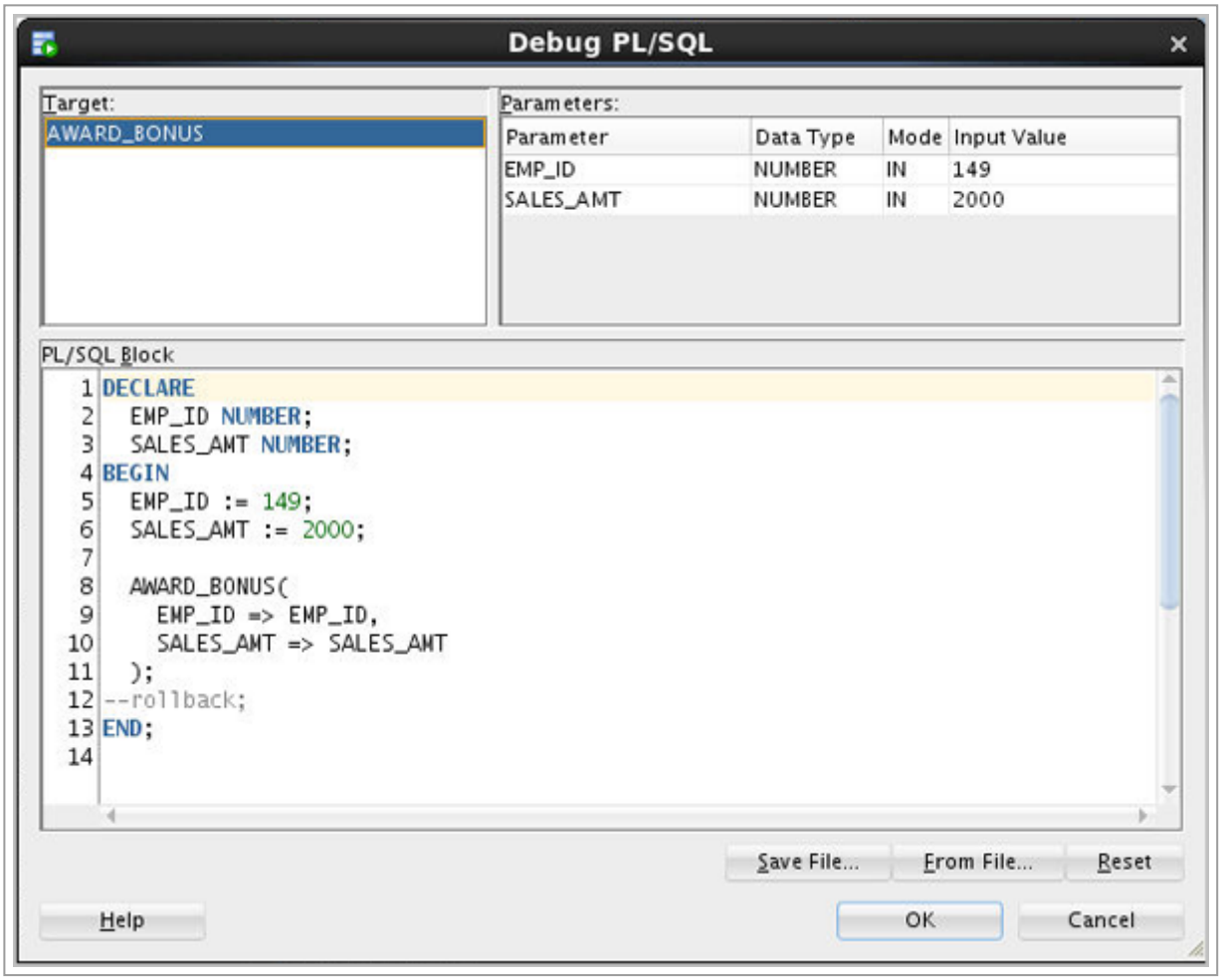
Run the procedure by clicking on Run icon.

3. The Run PL/SQL dialog window appears. Notice that the values for EMP_ID and SALES_AMT are currently set to 1.



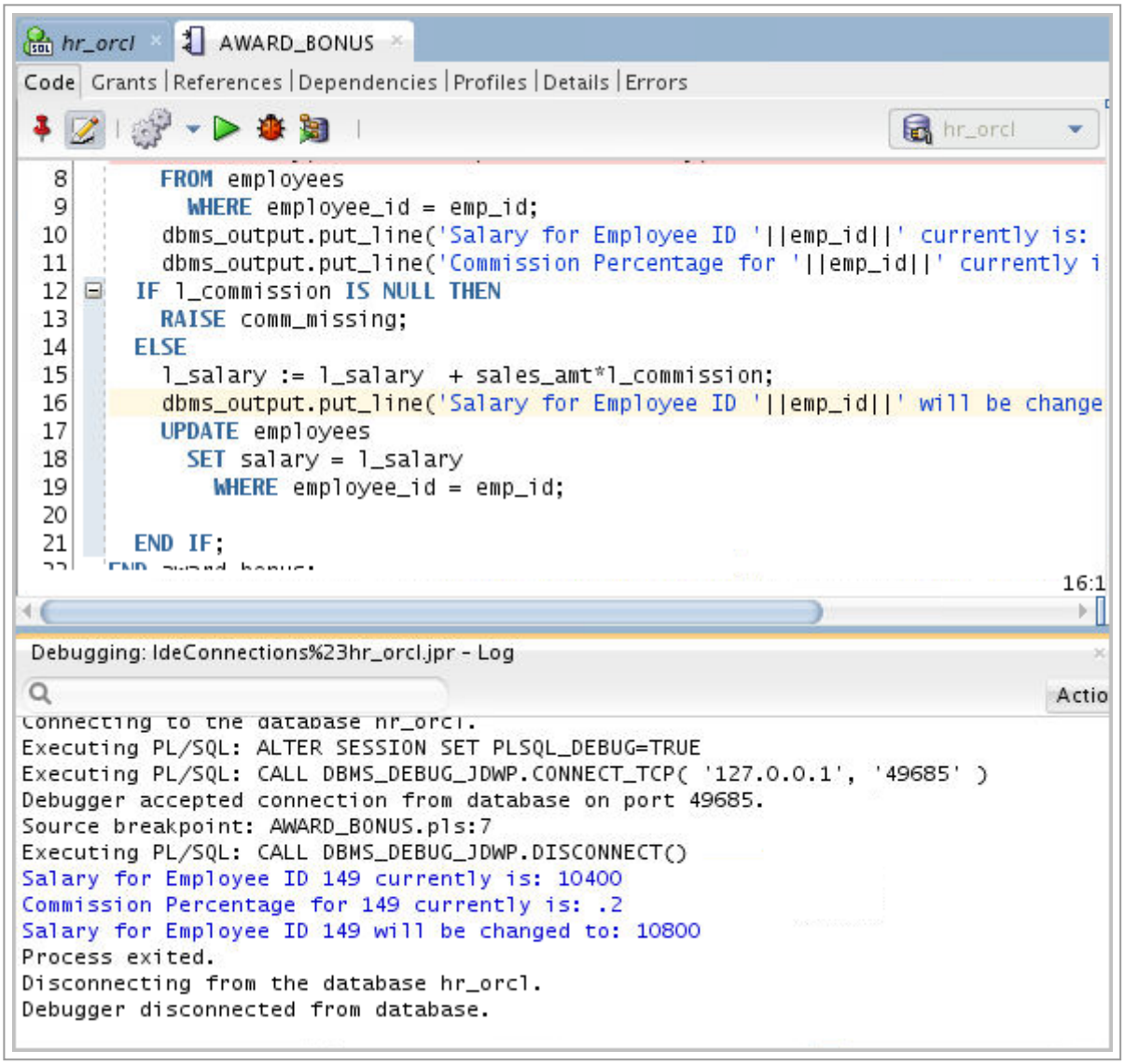
Description of this image (files/tab05_02_07.txt)

4. Change the default values to **149** for EMP_ID and **2000** for SALES_AMT and click **OK**.



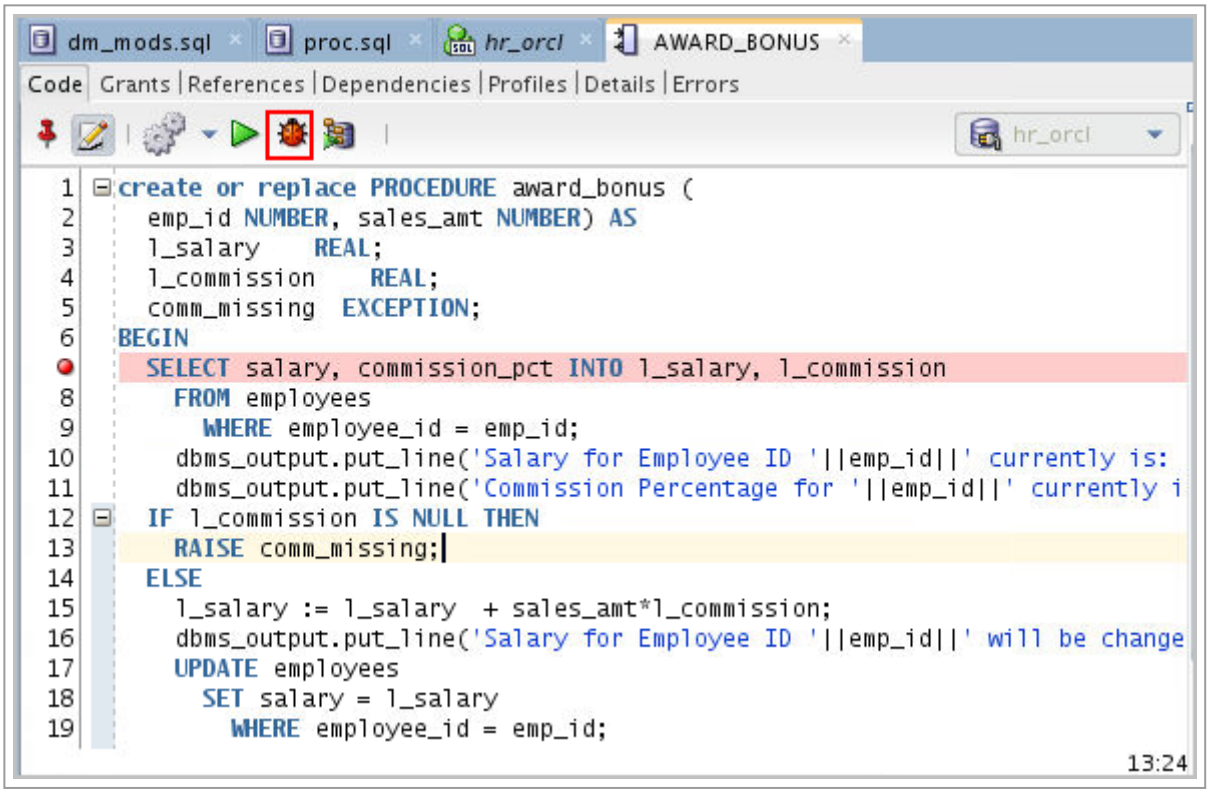
Description of this image (files/tab05_02_10.txt)

5. Note that the procedure executed successfully and the value for salary was changed. To see how debug works, you create a break point. Click the line number **7**.



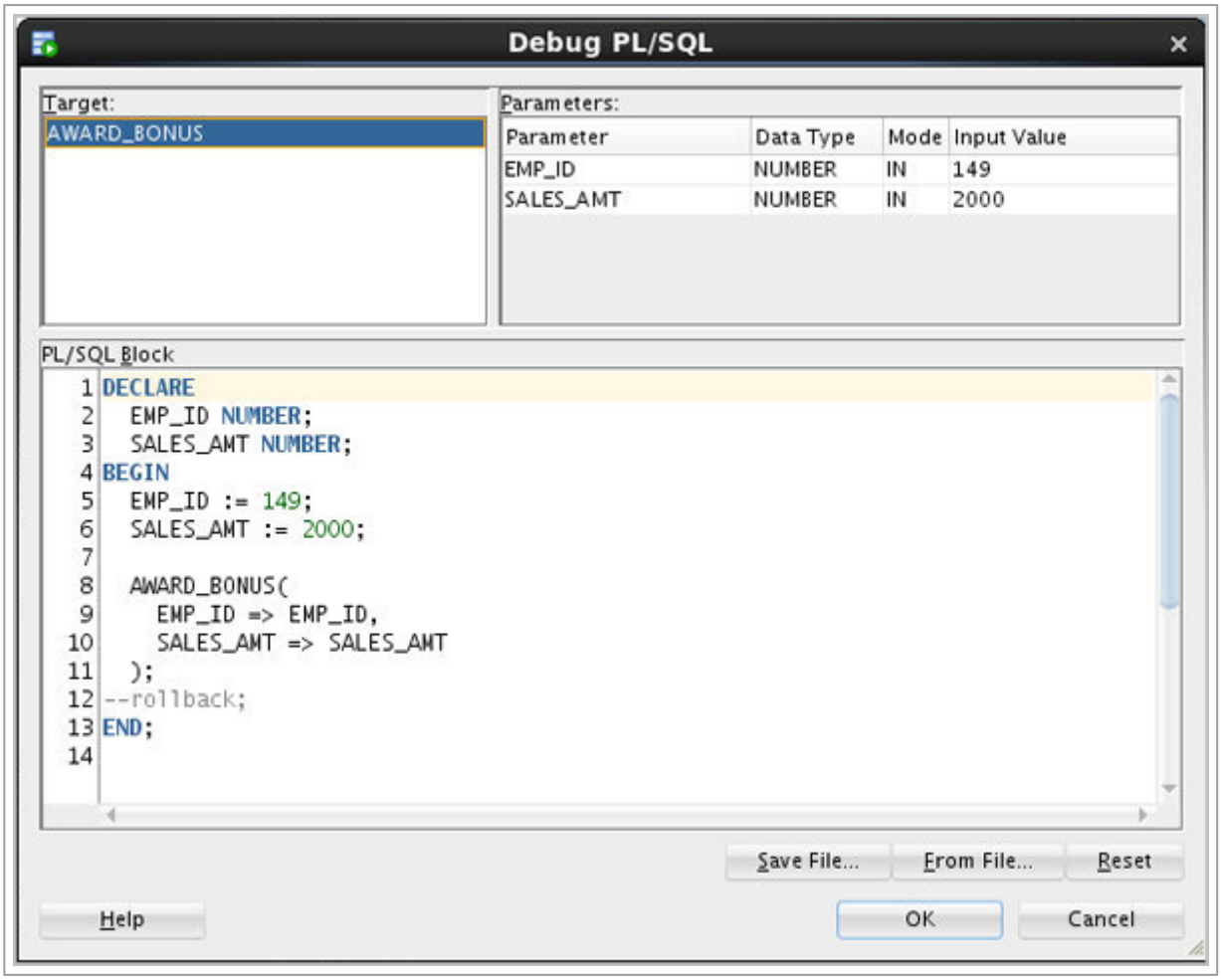
Description of this image (files/tab05_02_17.txt)

6. When a break point is created at line 7, the execution will break at line 7 and allows developer to monitor the data held in different variables. Click the **Debug** icon.



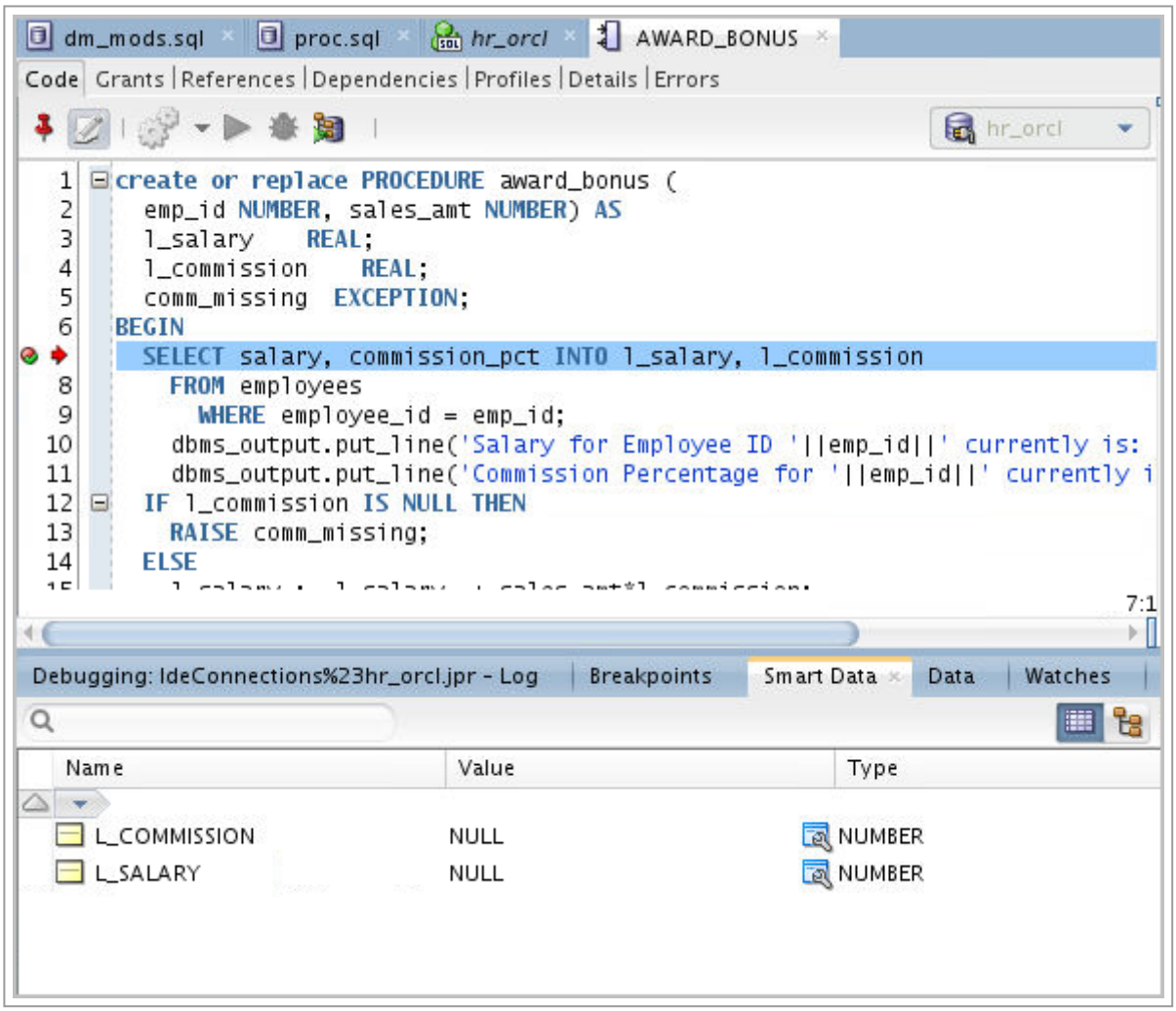
Description of this image (files/tab05_02_09.txt)

7. Click **OK** to accept the same input values as before.



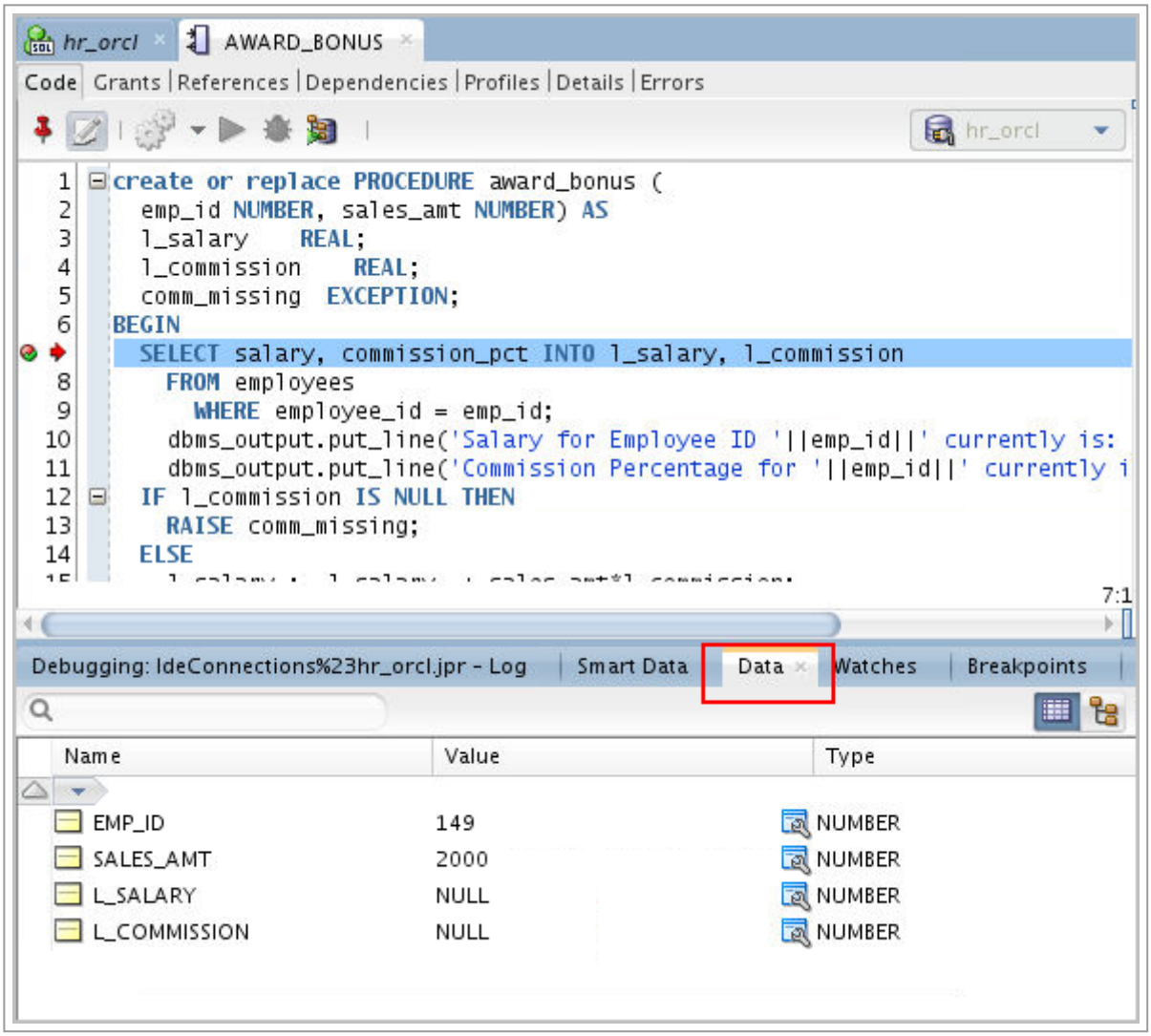
Description of this image (files/tab05_02_10.txt)

8. The debugger is running and has stopped at line 8. Click the **Smart Data** tab. The Smart Data tab holds the values of variables in the PL/SQL block. These are currently set to NULL.



Description of this image (files/tab05_02_11.txt)

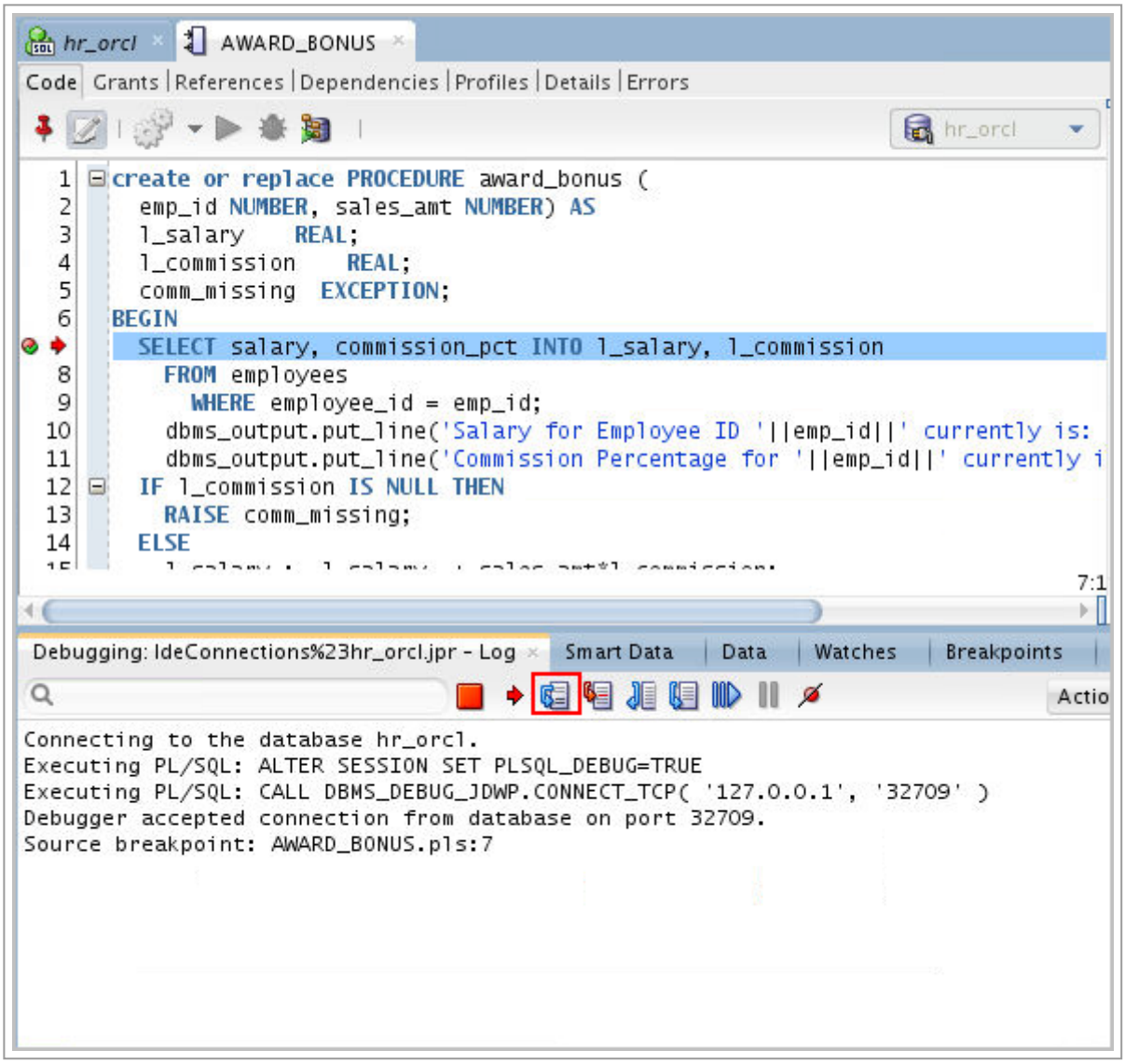
9. You can see all the data manipulated in the procedure in the **Data** tab.



Description of this image (files/tab05_02_12.txt)

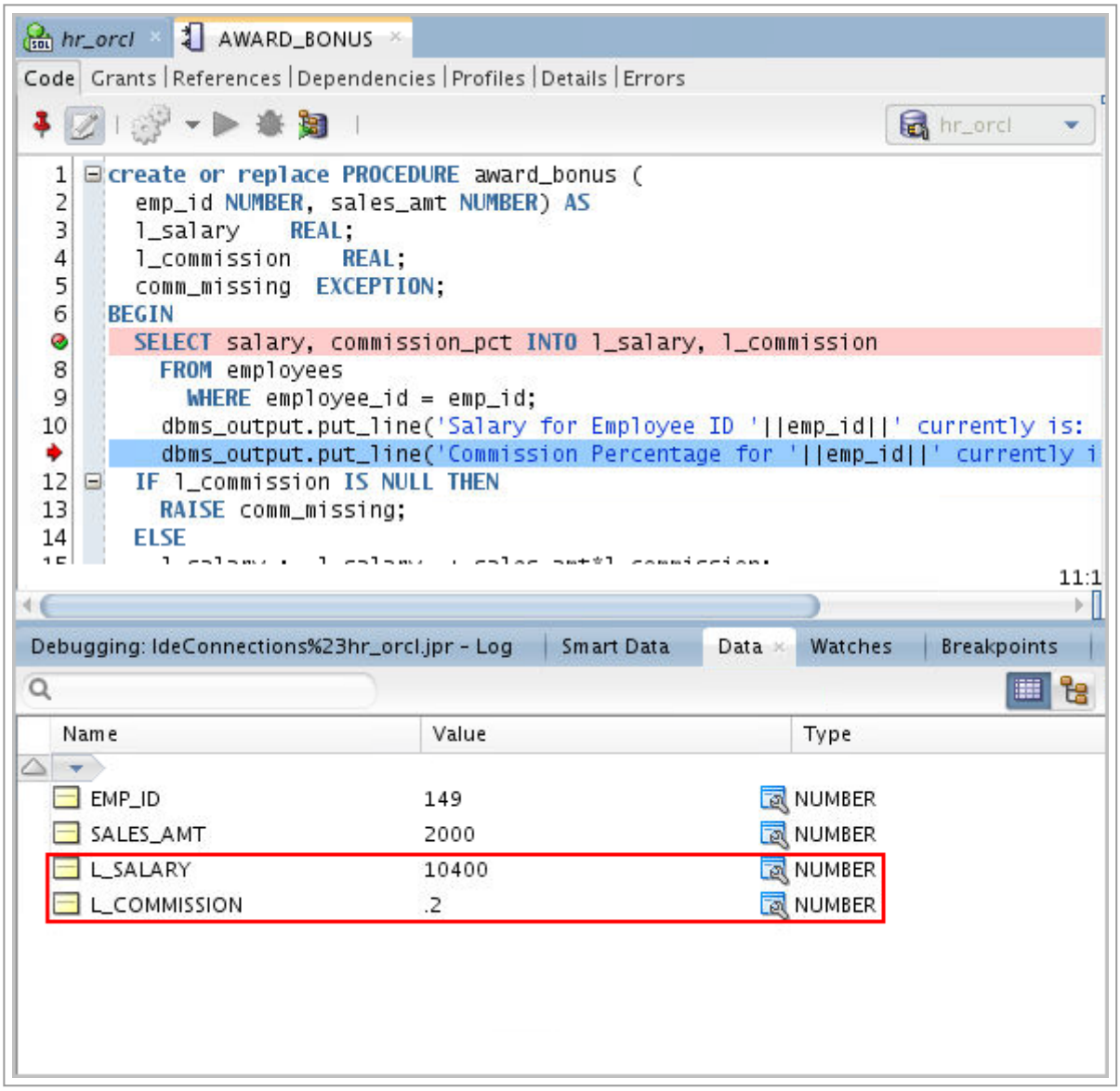
You see that the current values of l_salary and l_commission are NULL.

10. Click the **Step Over** icon to move to the next statement in the procedure.



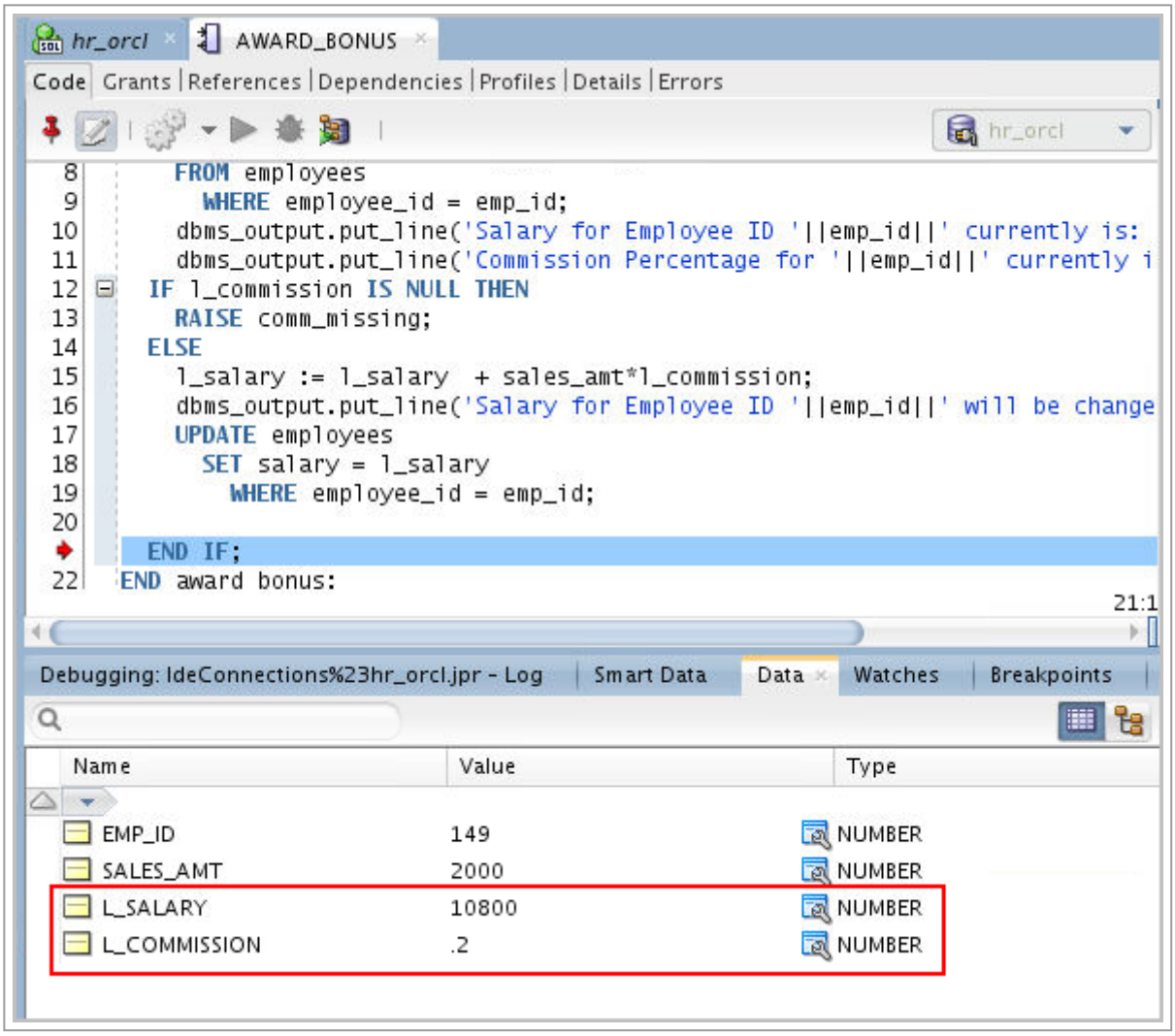
Description of this image (files/tab05_02_13.txt)

11. Notice the values for l_salary and l_commission have changed to the existing values in the database, as the execution of select statement is complete, you can see the values from the database are fetched into the variables in the procedure.



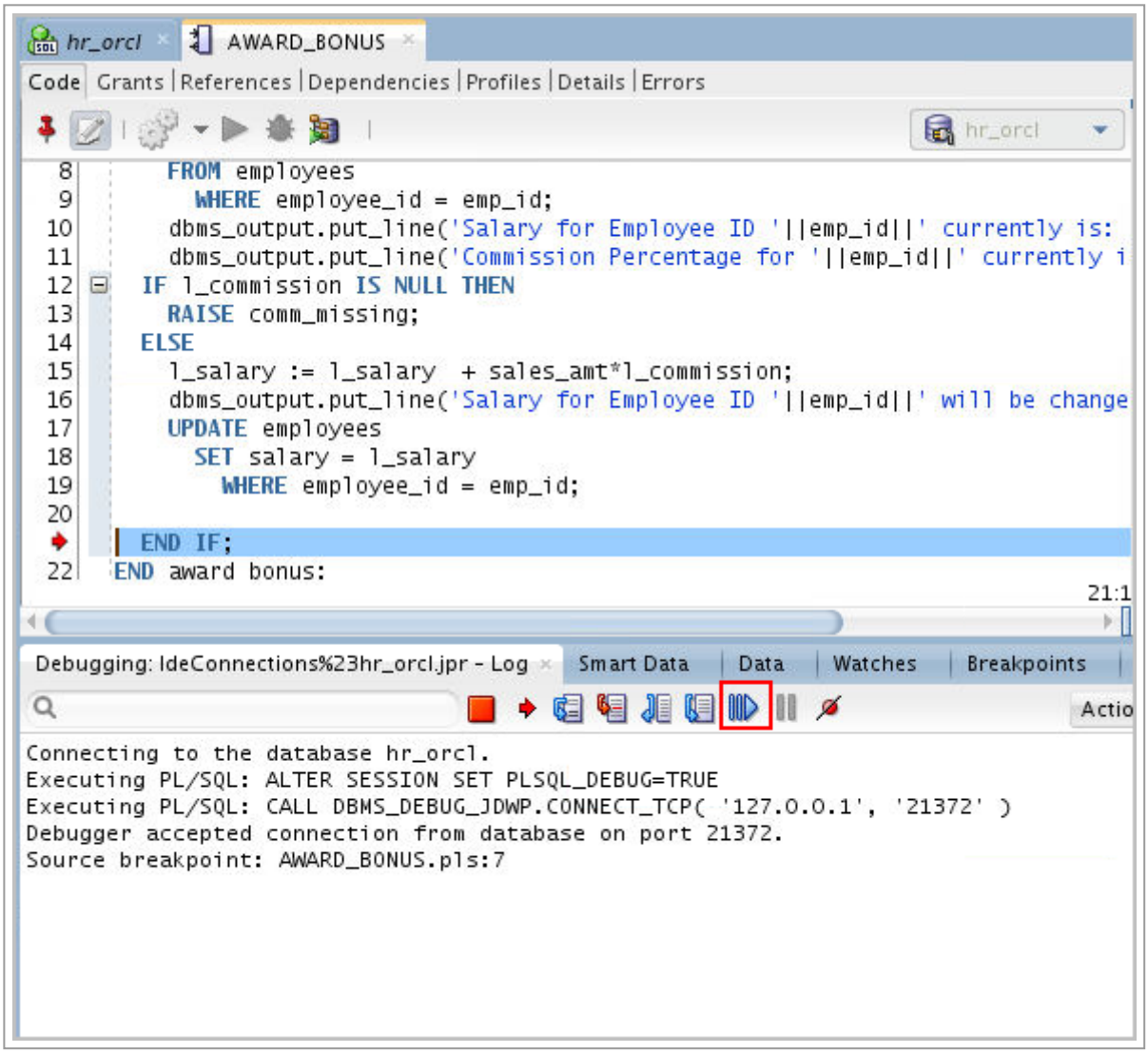
Description of this image (files/tab05_02_14.txt)

12. Click the **Step Over** icon again to move to the next statement.As the execution of the update statement completes, you can see the new values of salary and commission in the Data tab



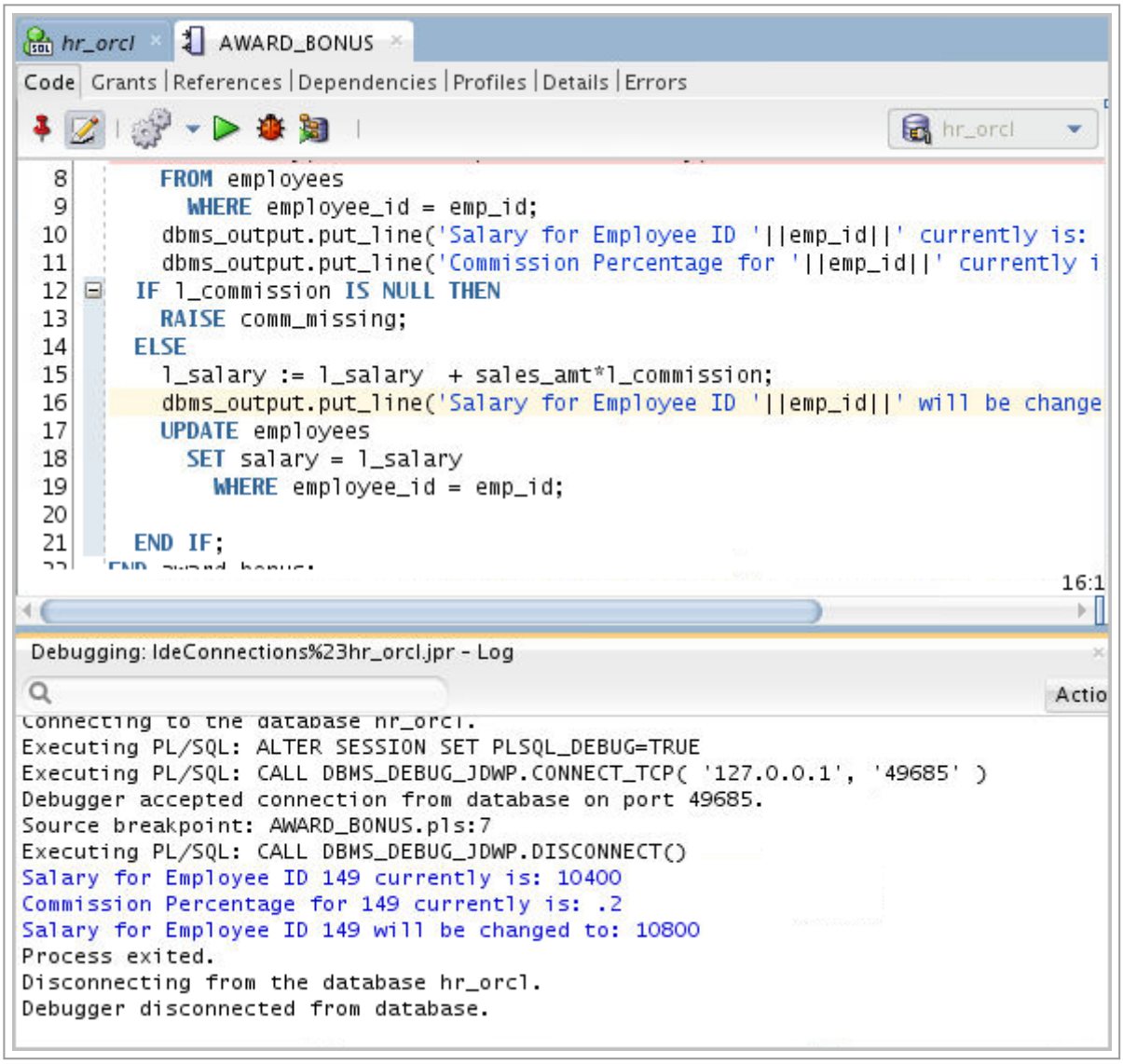
Description of this image (files/tab05_02_15.txt)

13. Notice that the debugger moved to the next statement in the procedure. You want to run the rest of the procedure, click the **Resume** icon.



Description of this image (files/tab05_02_16.txt)

14. Procedure execution and debugging is complete. In the next topic, you create a test repository so that you can create and run a unit test.

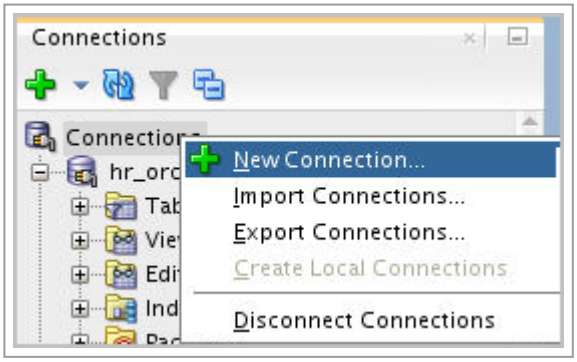


Description of this image (files/tab05_02_17.txt)

Creating a Unit Test Repository

In this topic, you create a database user called UNIT_TEST_REPOS. You create this user to hold the Unit Testing Repository data. You will then create the repository in the schema of the user that you created.

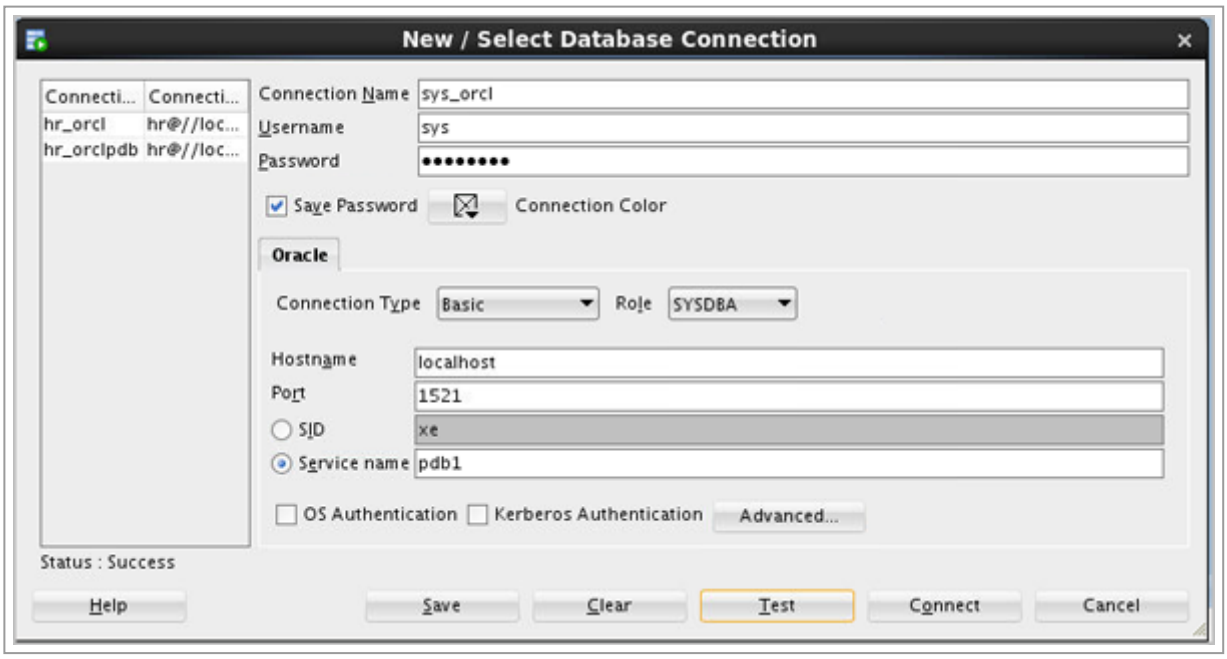
1. Create a connection for the SYS User. Right-click **Connections** and select **New Connection**.



Description of this image (files/tab06_01.txt)

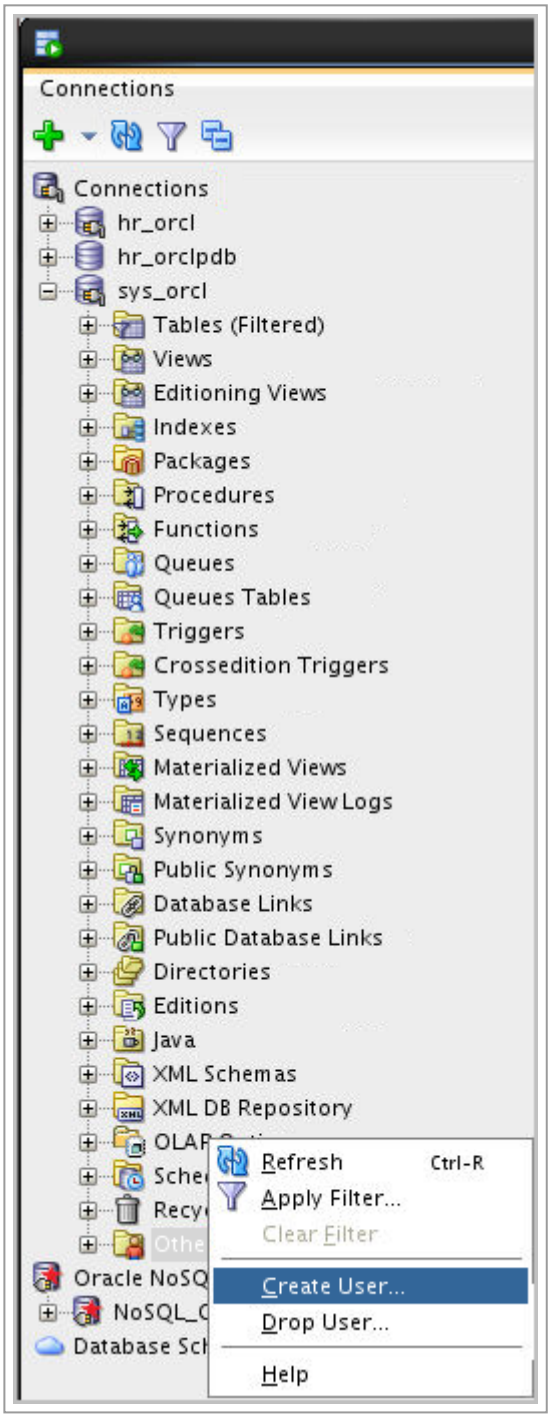
2. Enter the following information and click **Connect**.

Connection Name: **sys_orcl**
Username: **sys**
Password: **oracle**
Select **Save Password** checkbox
Role: **SYSDBA**
Service Name: **pdb1**



Description of this image (files/tab06_02.txt)

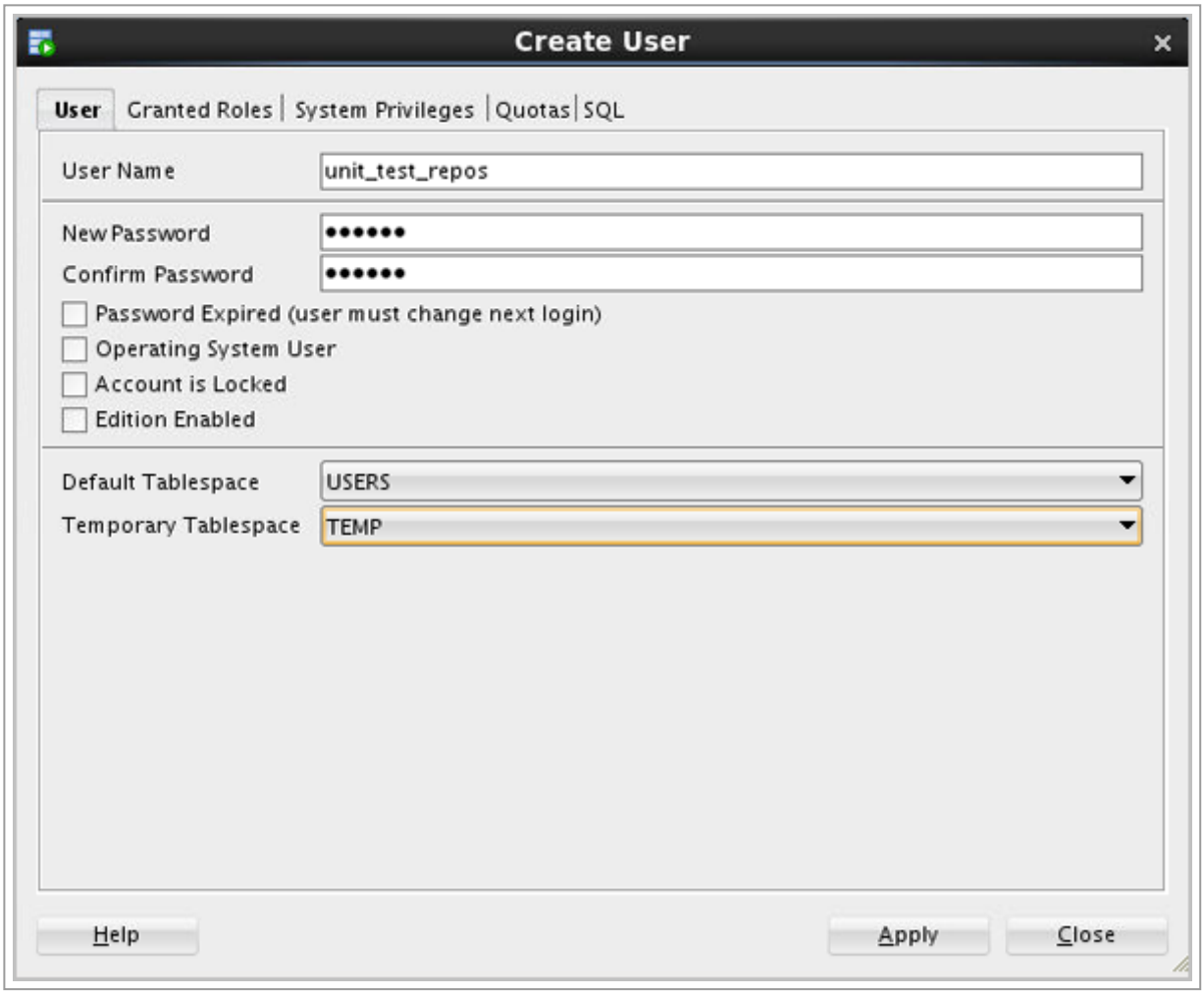
3. Your connection was created successfully. Collapse the **hr_orcl** connection. Expand the **sys_orcl** connection and right-click **Other Users** and select **Create User**.



Description of this image (files/tab06_03.txt)

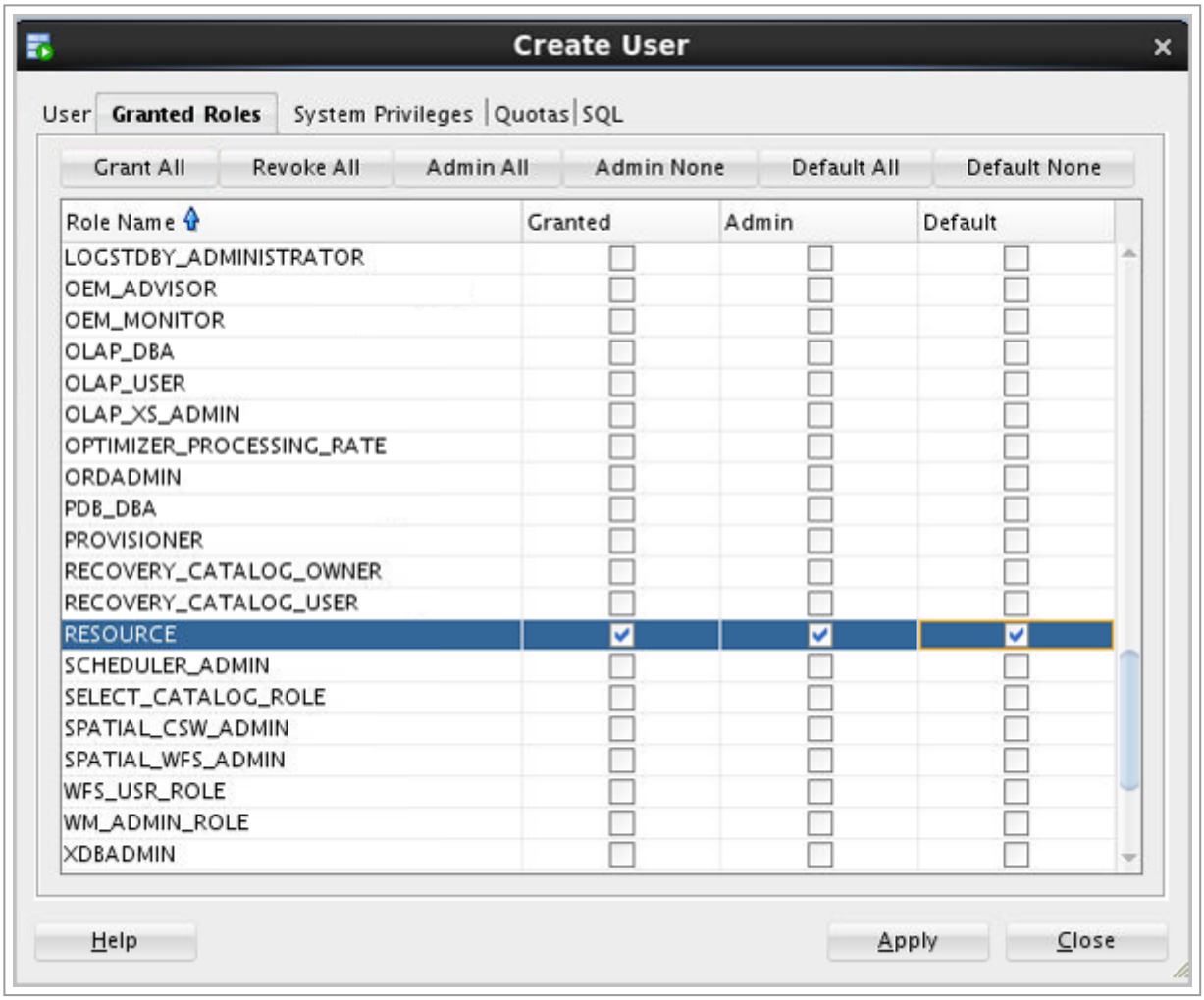
4. Enter the following information and select the **Granted Roles** tab.

   Username: **unit_test_repos**
   Password: **oracle**
   Default Tablespace: **USERS**
   Temporary Tablespace: **TEMP**



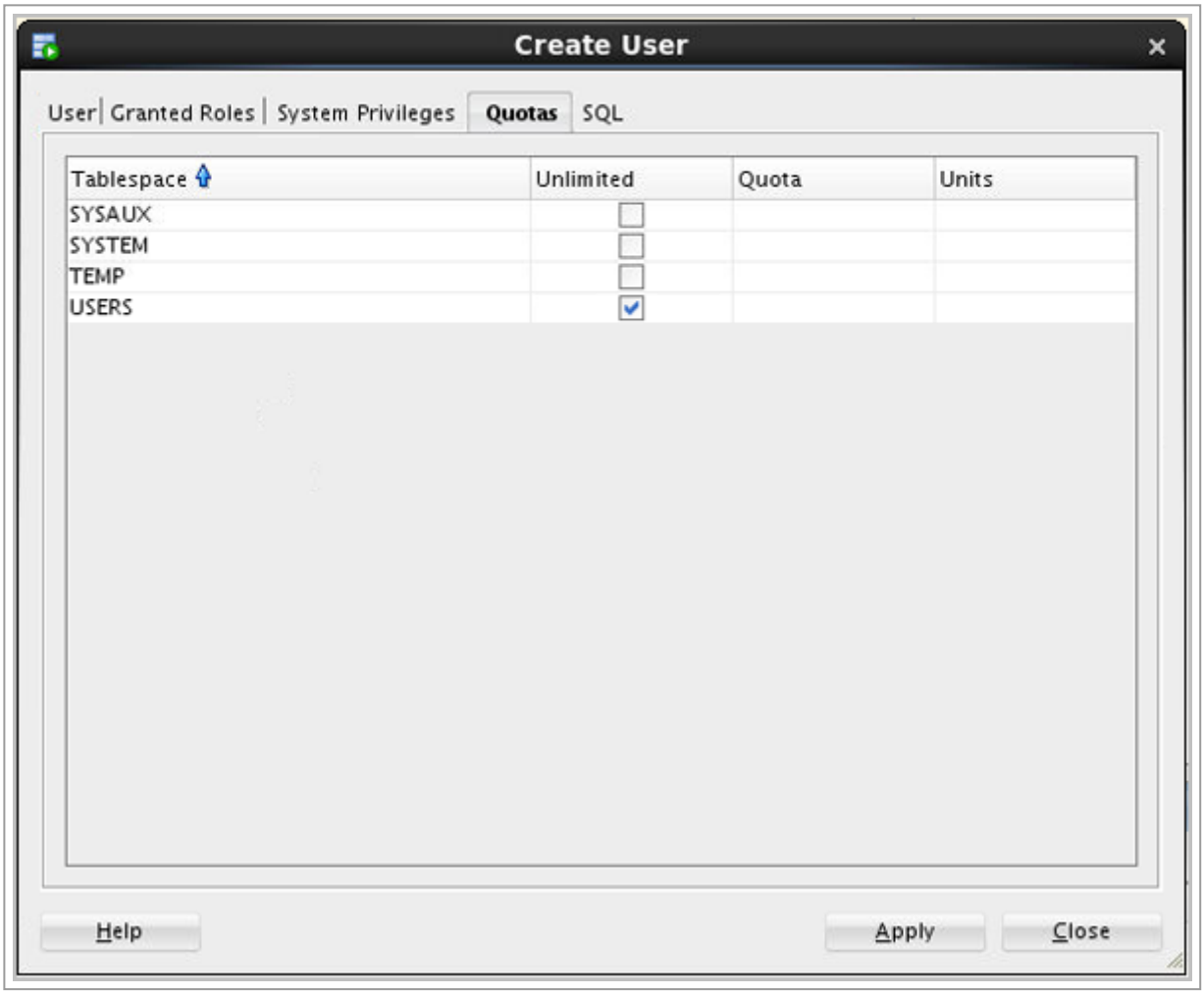
Description of this image (files/tab06_04.txt)

5. Select the **Connect** and **Resource** roles and click **Apply**.



Description of this image (files/tab06_05.txt)

6. In the Quotas tab, check the Unlimited check box for the USERS tablespace

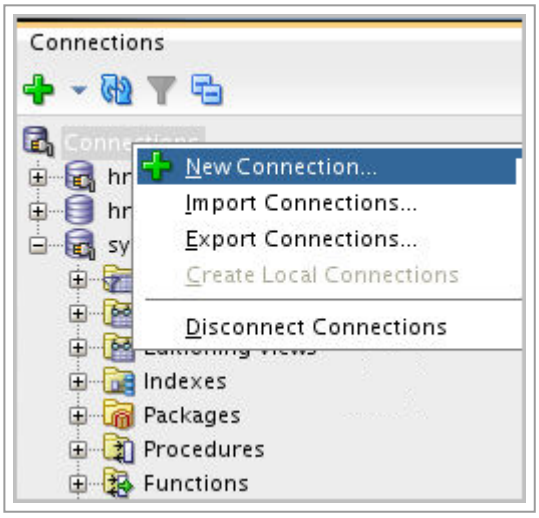


Description of this image (files/tab06_06.txt)

7. The unit_test_repos user was created successfully. Click **OK**.

Description of this image (files/tab06_07.txt)

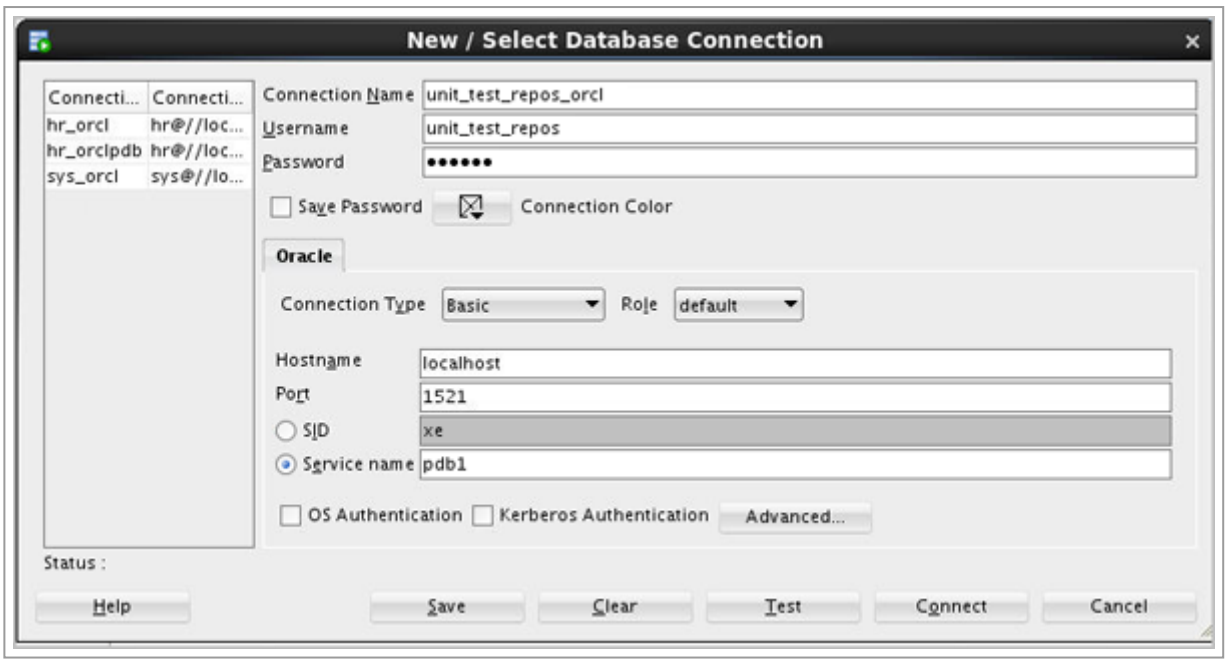
8. You now need to create a connection to the unit_test_repos user. This user will hold the unit testing repository data. Right-click **Connections** and select **New Connection**.



Description of this image (files/tab06_08.txt)

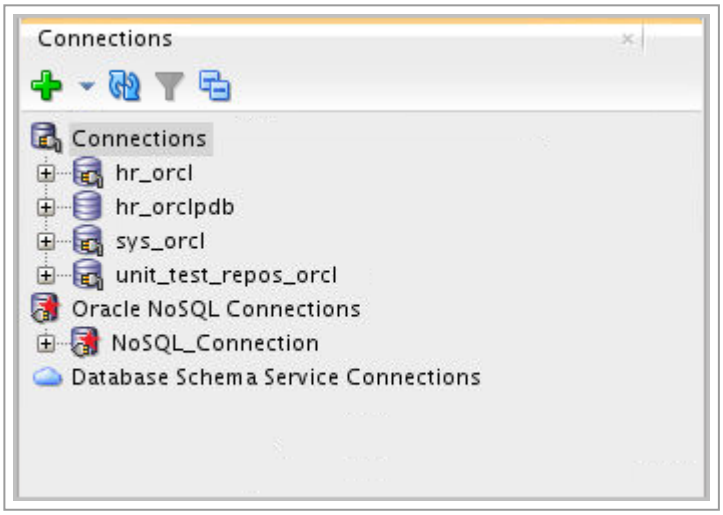
9. Enter the following information and click **Connect**.

Connection Name: **unit_test_repos_orcl**
Username: **unit_test_repos**
Password: **oracle**
Select **Save Password** checkbox
Service Name: **pdb1**



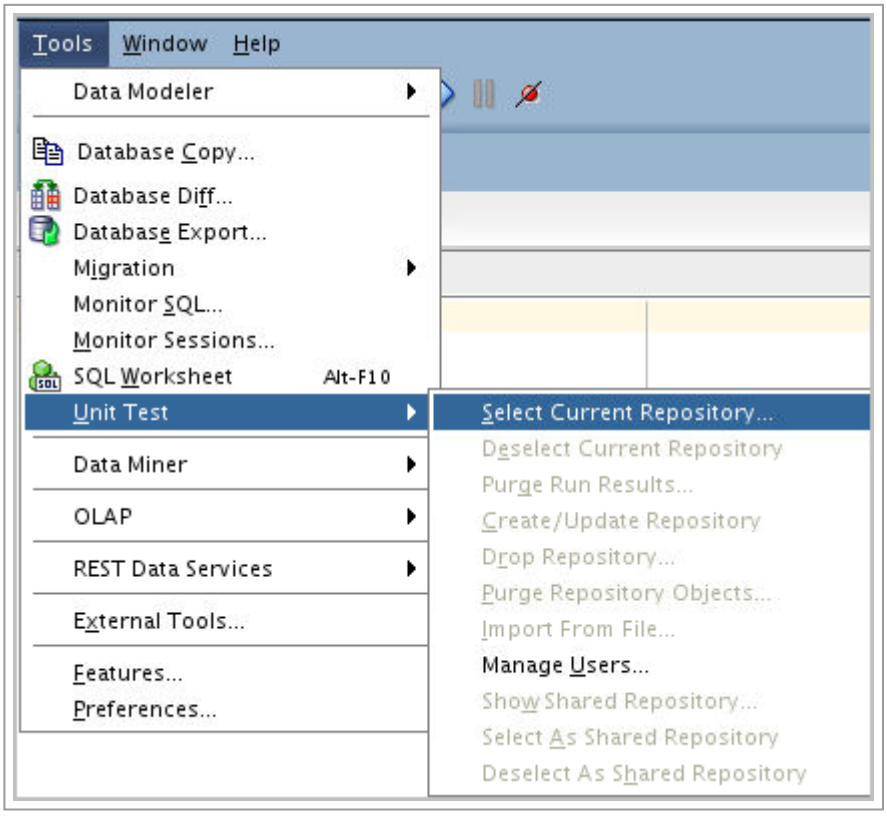
Description of this image (files/tab06_09.txt)

10. The unit_test_repos user and unit_test_repos_orcl connection were created successfully.



Description of this image (files/tab06_10.txt)

11. Select **Tools** > **Unit Test** > **Repository**, then select **Select Current Repository**.

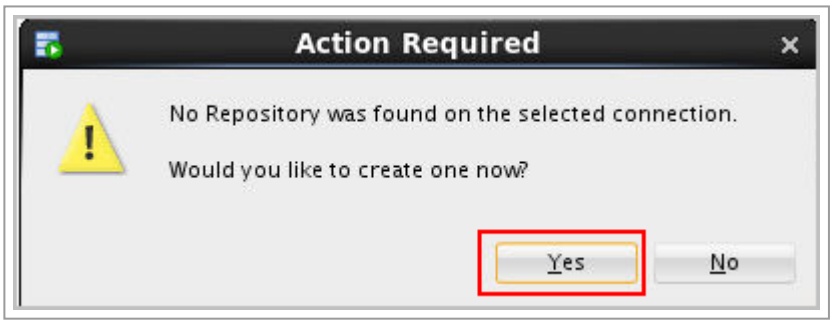


Description of this image (files/tab06_11.txt)

12. Select the **unit_test_repos_orcl** connection and click **OK**.

Description of this image (files/tab06_12.txt)

13. You would like to create a new repository. Click **Yes**.

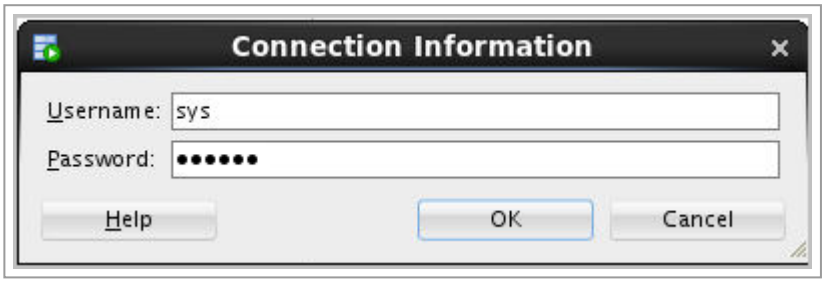


Description of this image (files/tab06_13.txt)

14. This connection does not have the permissions it needs to create the repository. Click **OK** to show the permissions that will be applied.

Description of this image (files/tab06_14.txt)

15. Enter **oracle** for the sys password and click **OK**.



Description of this image (files/tab06_15.txt)

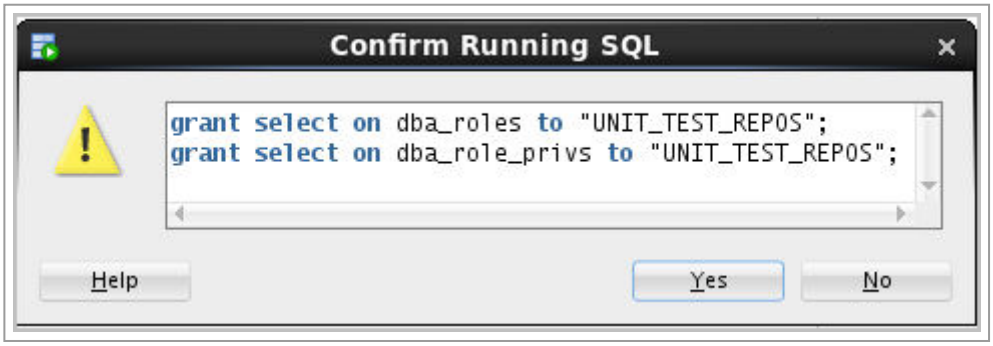
16. The grant statement is shown. Click **Yes**.

Description of this image (files/tab06_16.txt)

17. The UNIT_TEST_REPOS user needs select access to some required tables. Click **OK**.

Description of this image (files/tab06_17.txt)

18. The grant statements are displayed. Click **Yes**.

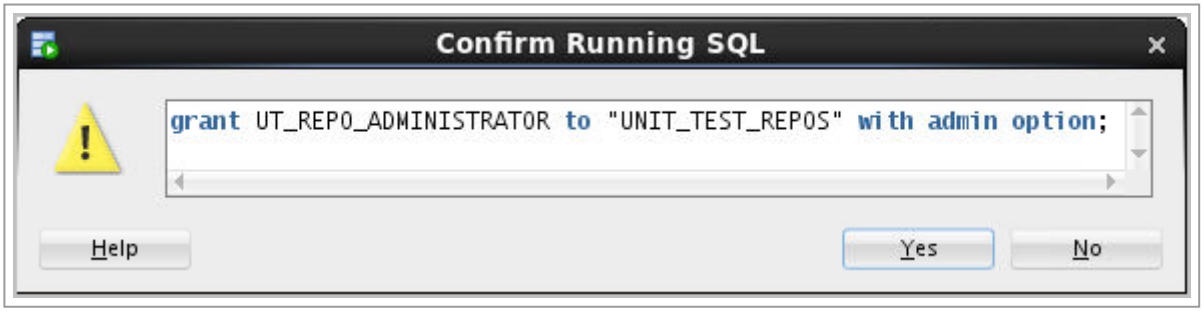


Description of this image (files/tab06_18.txt)

19. The UNIT_TEST_REPOS user does not currently have the ability to manage repository owners. Click **OK** to see the grant statements that will be executed.

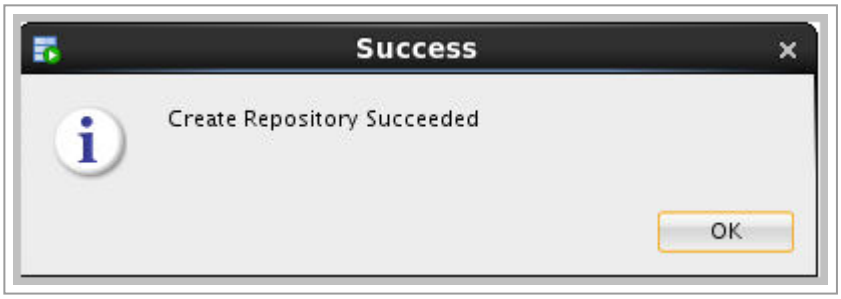
Description of this image (files/tab06_19.txt)

20. The grant statements are displayed. Click **Yes**.



Description of this image (files/tab06_20.txt)

21. Your repository was created successfully. Click **OK**.

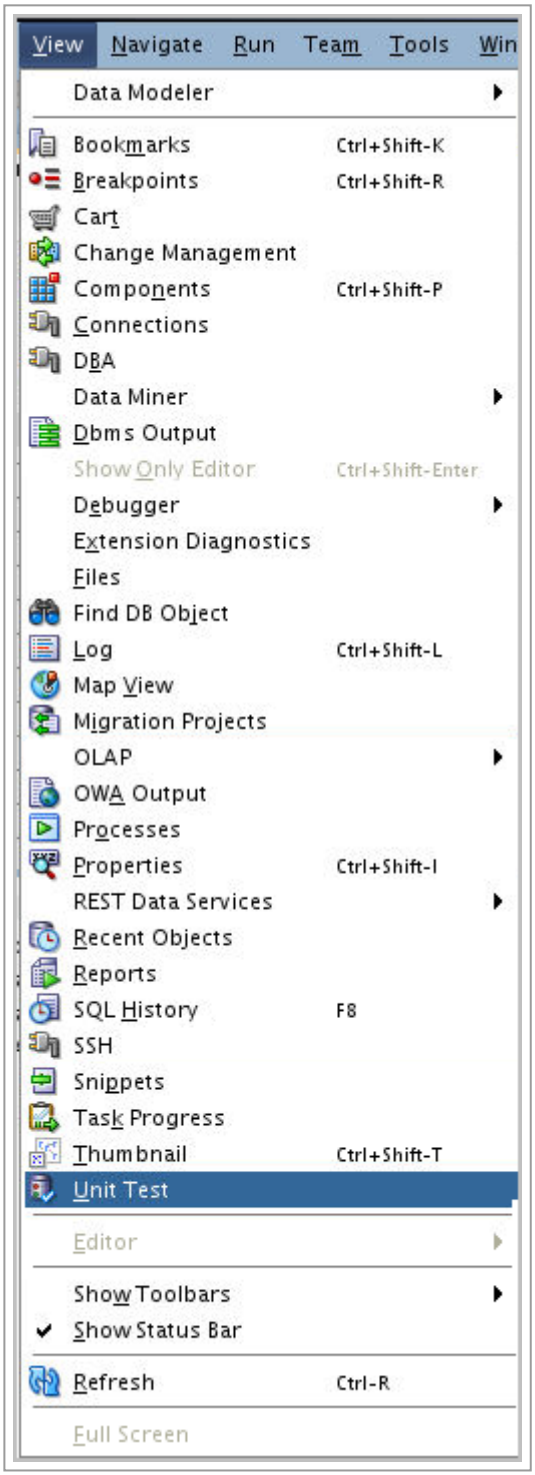


Description of this image (files/tab06_21.txt)

Creating and Running a Unit Test

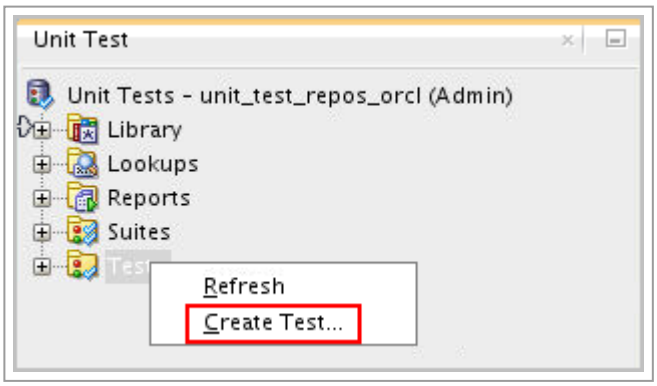
Now that the Unit Testing Repository has been created, you will create a unit test for the PL/SQL procedure you created earlier in this tutorial. Then you will run the unit test to see if various values will work

1. Select **View** > **Unit Test**.



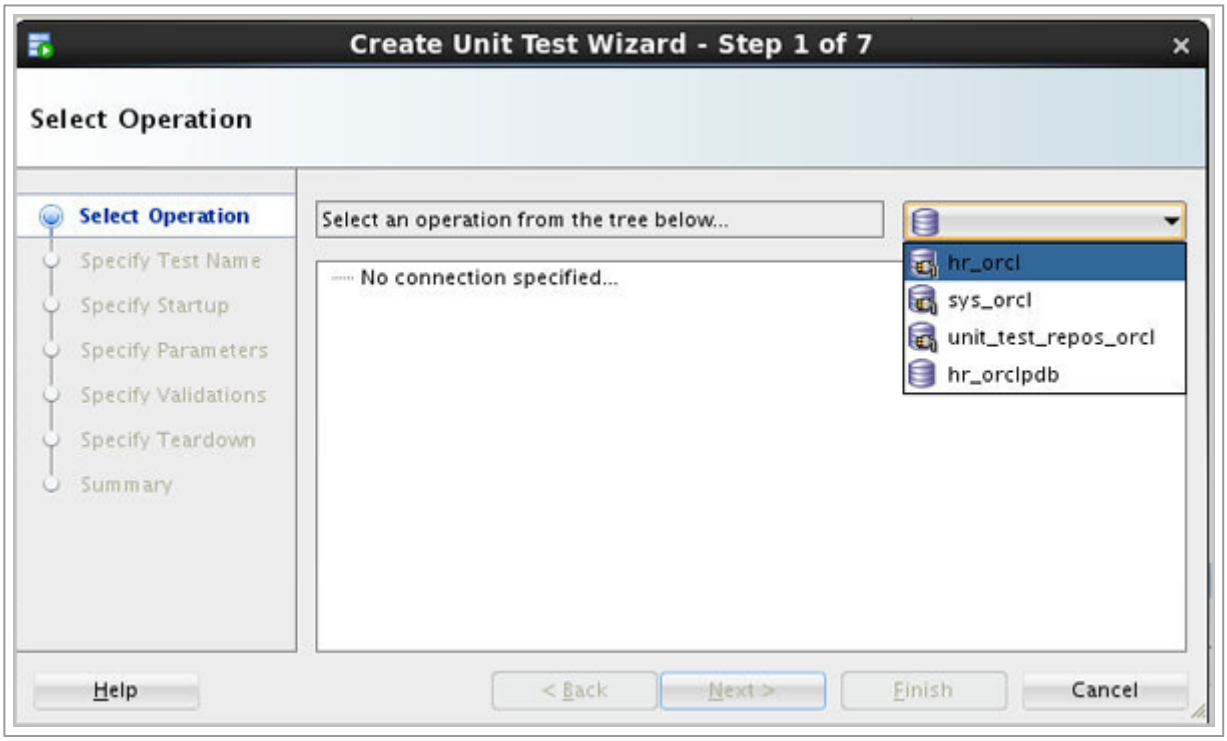
Description of this image (files/tab07_01.txt)

2. In the Unit Test navigator, right-click **Tests** and select **Create Test**.



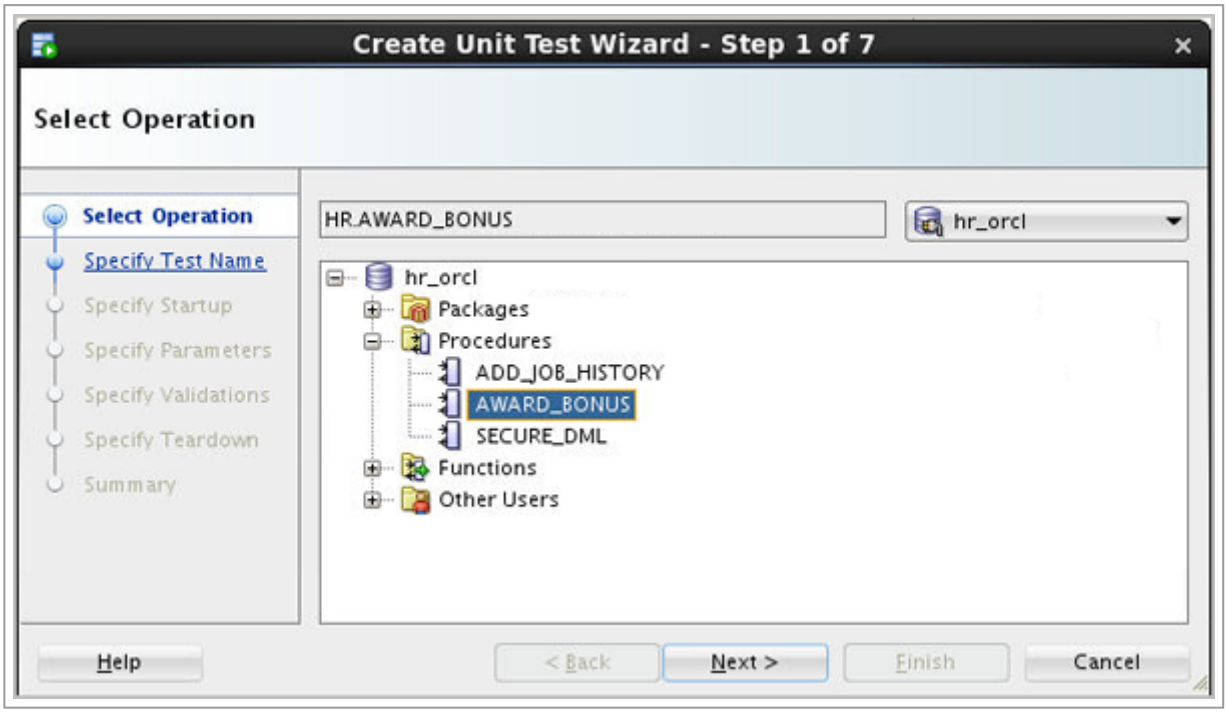
Description of this image (files/tab07_02.txt)

3. In Select Operation, select the **hr_orcl** connection that you used to create the **AWARD_BONUS** procedure.



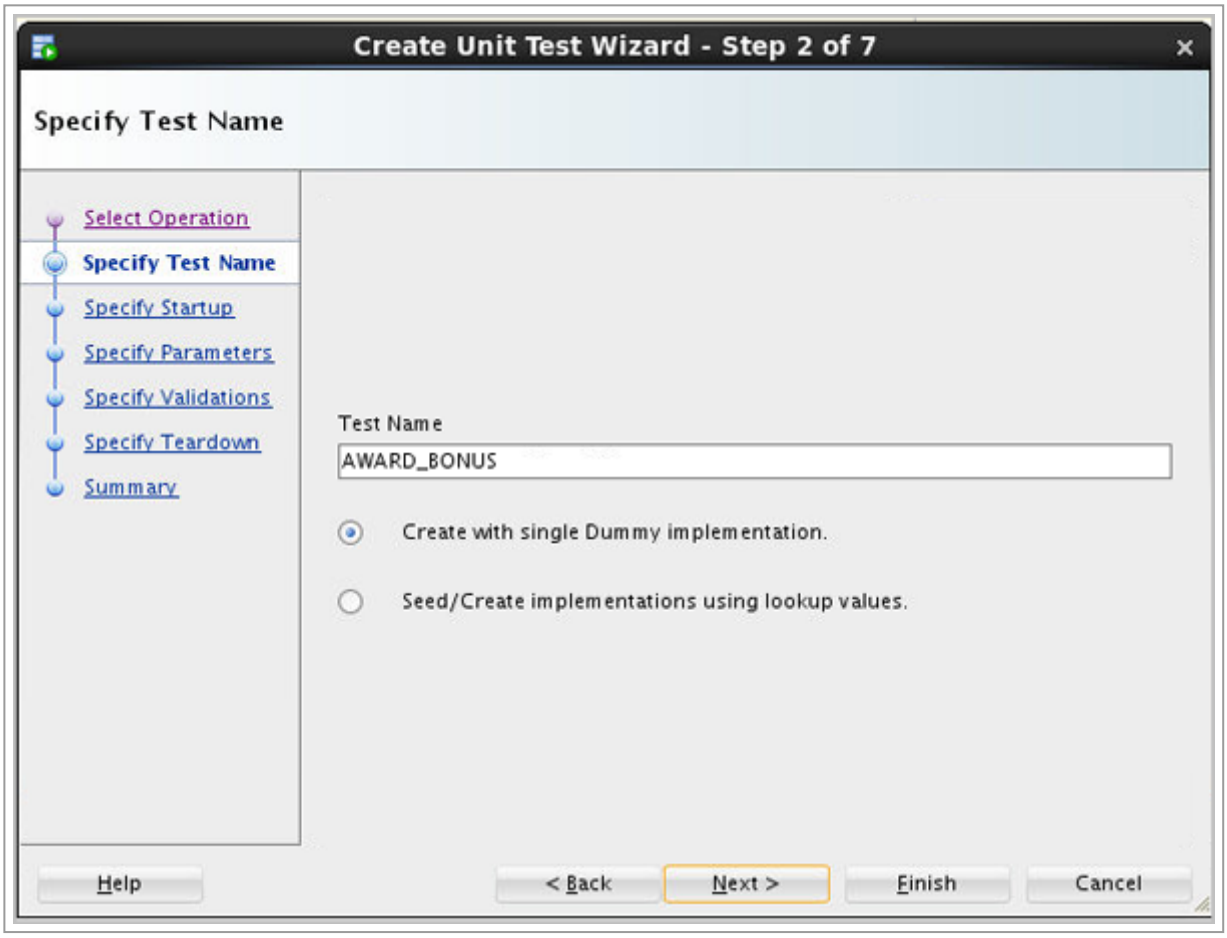
Description of this image (files/tab07_03.txt)

4. Expand **Procedures**, select **AWARD_BONUS** and click **Next**.



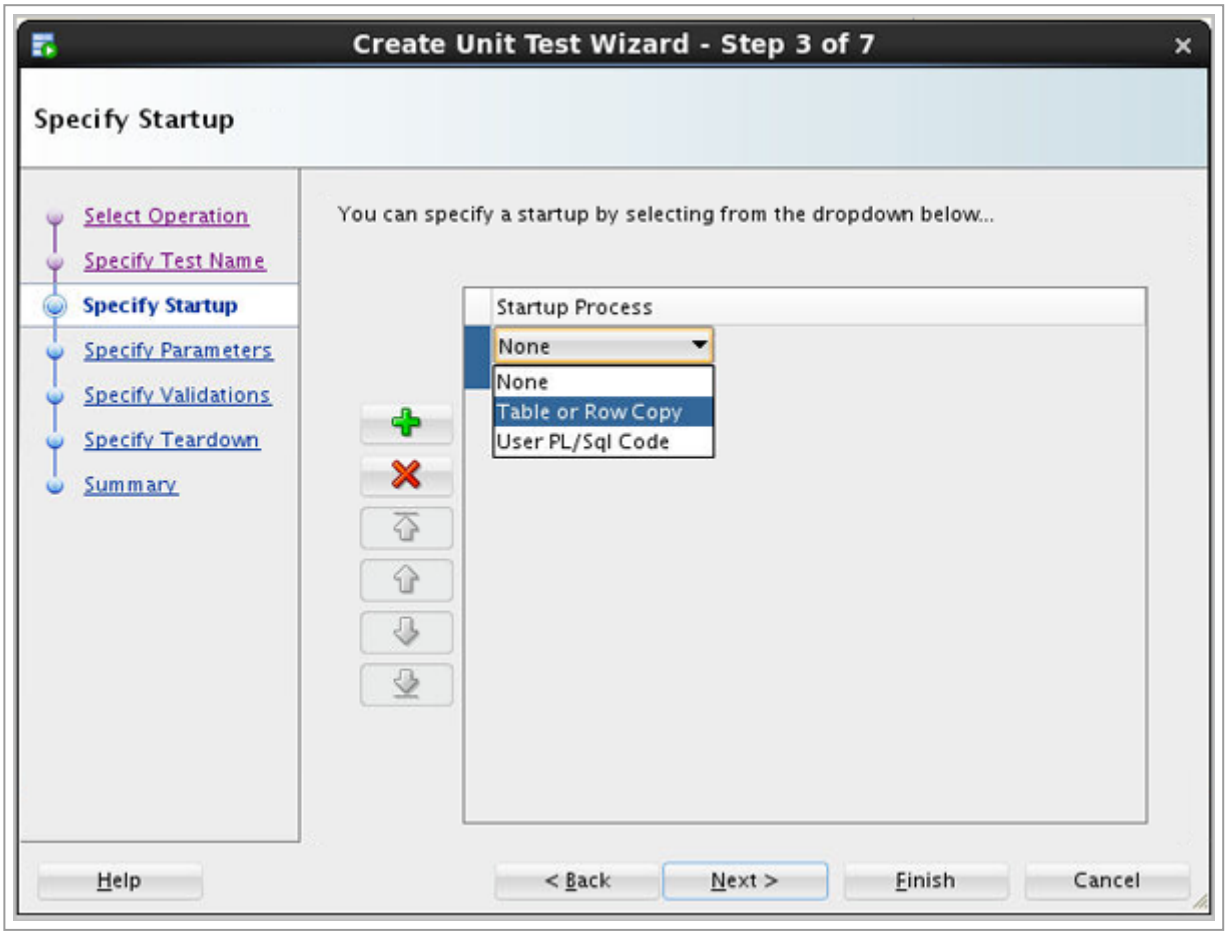
Description of this image (files/tab07_04.txt)

5. In Specify Test Name window, make sure that **AWARD_BONUS** is specified for Test Name and that **Create with single Dummy implementation** is selected, then click **Next**.



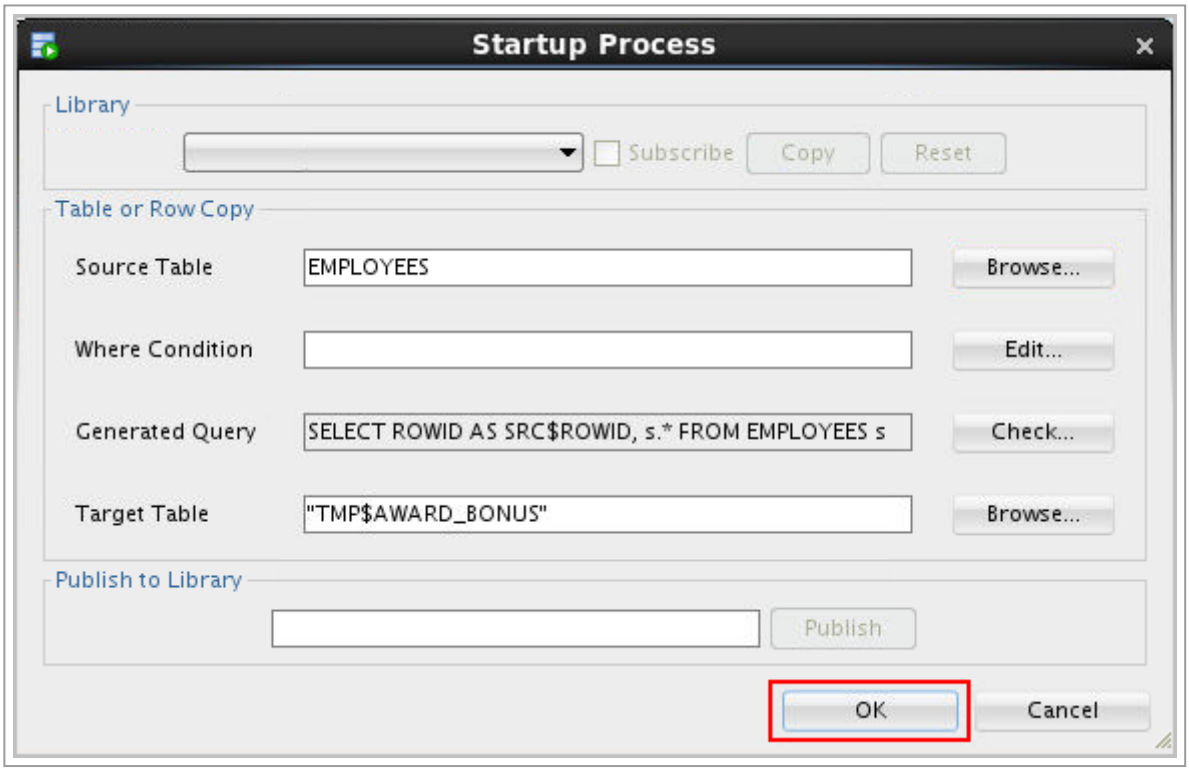
Description of this image (files/tab07_05.txt)

6. In Specify Startup window, click '+' icon and select **Table or Row Copy** from the drop down list box.



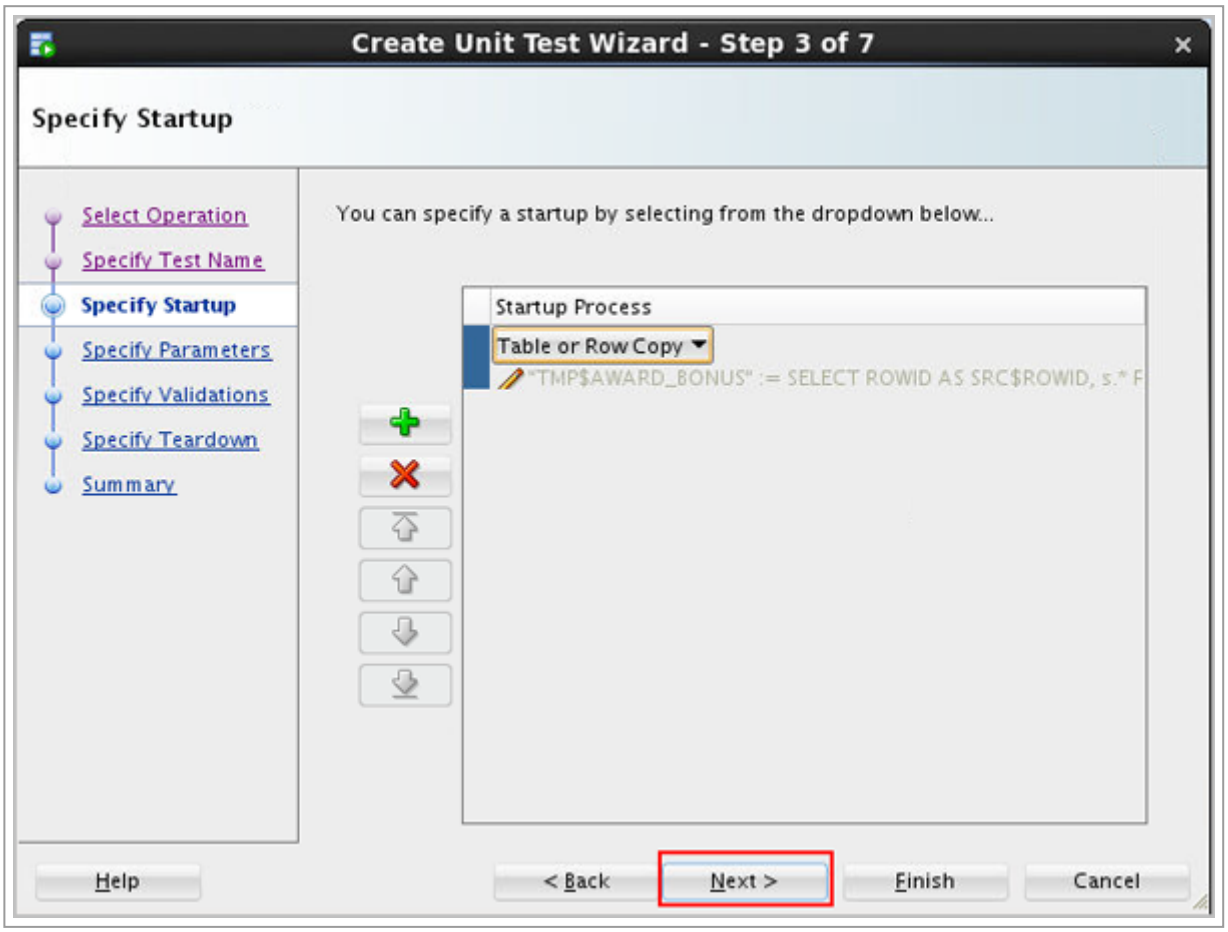
Description of this image (files/tab07_06.txt)

7. Enter **EMPLOYEES** for Source Table and click **OK**. Note that the table affected by the test will be saved to a temporary table and the query to the table is automatically generated.



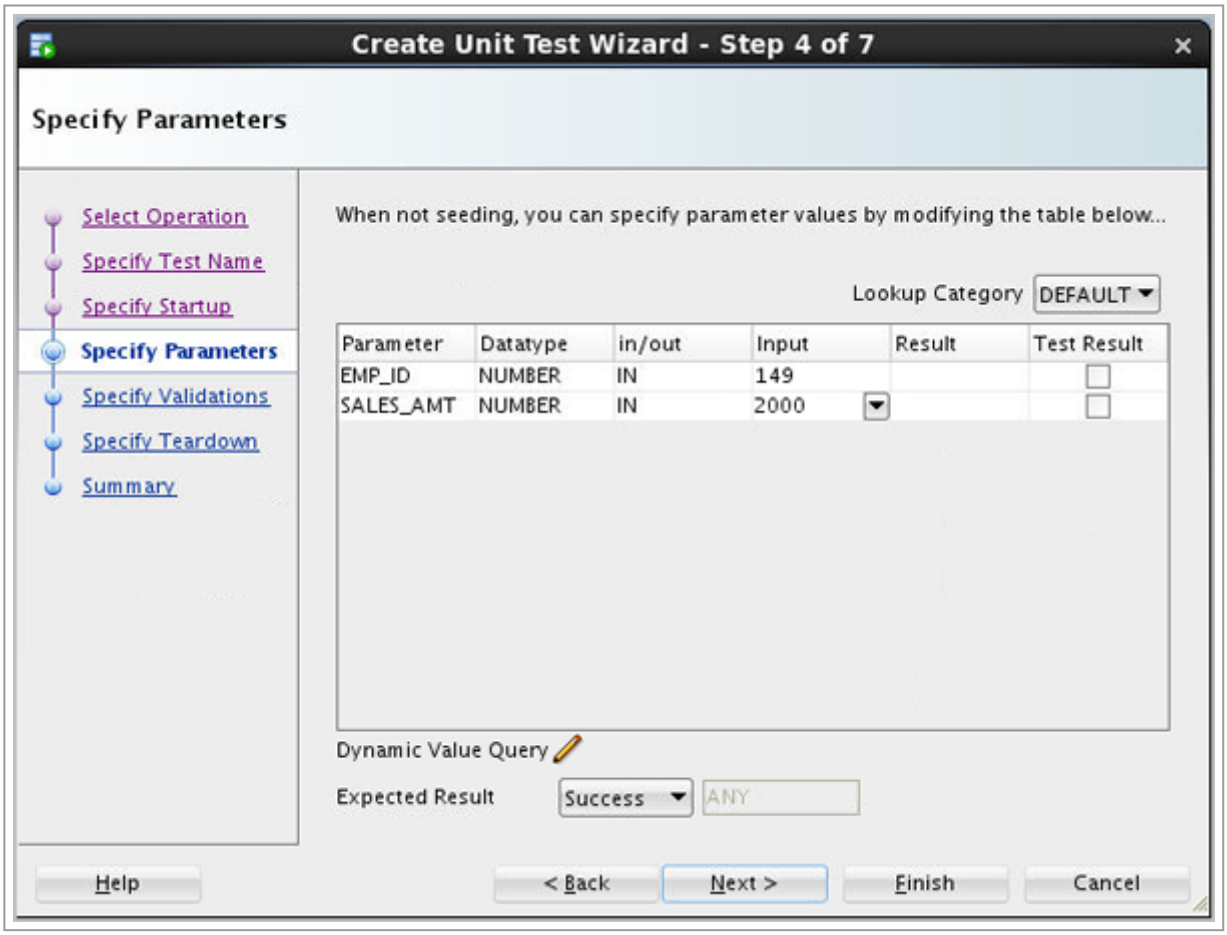
Description of this image (files/tab07_07.txt)

8. Click **Next**.



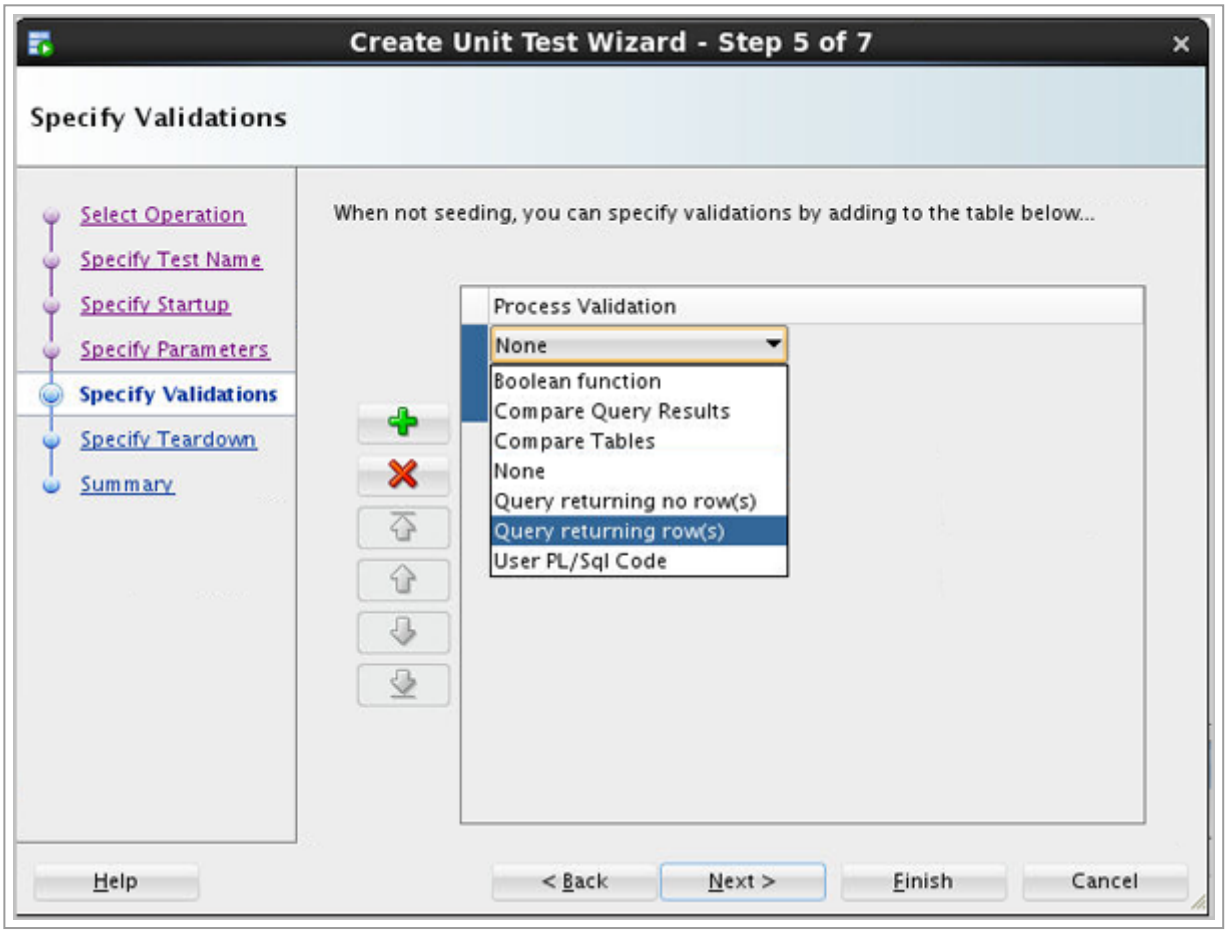
Description of this image (files/tab07_08.txt)

9. In the Specify Parameters window, change the Input string for EMP_ID to **149** and SALES_AMT to **2000** and click **Next**.



Description of this image (files/tab07_09.txt)

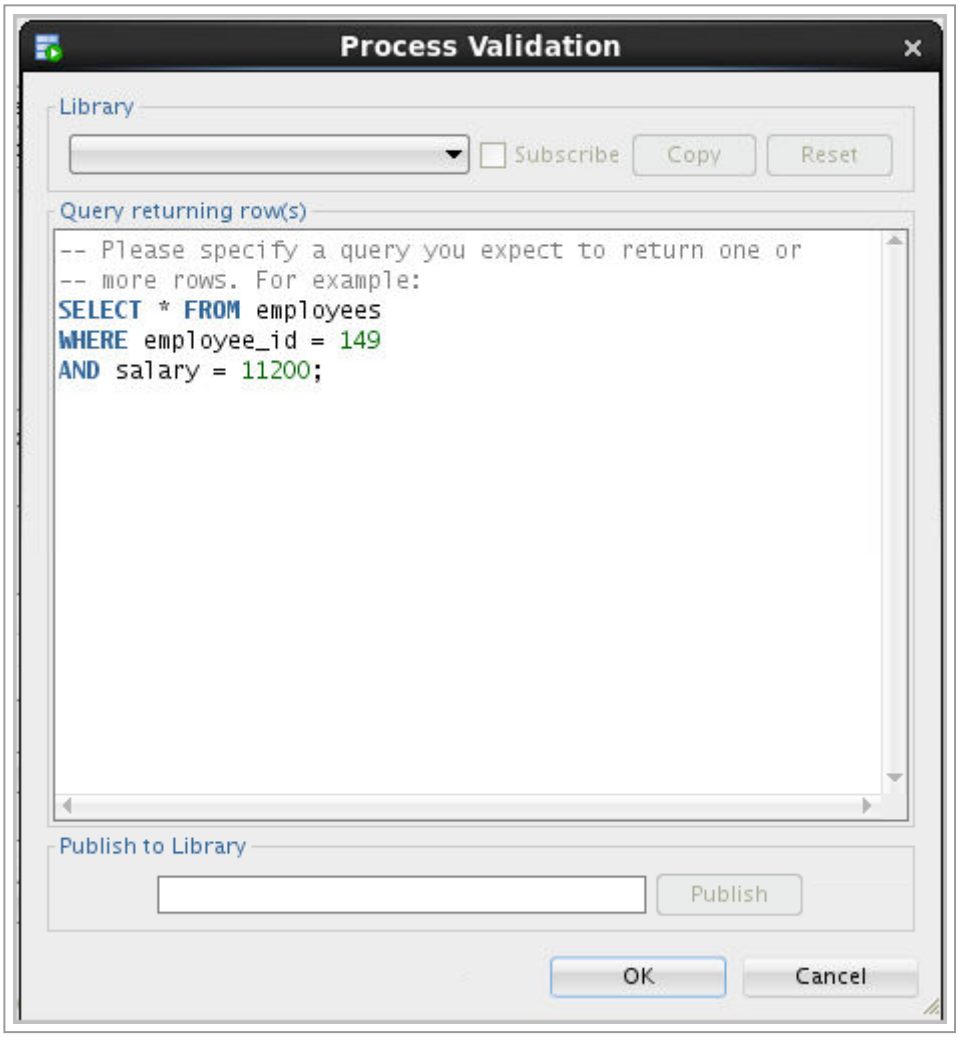
10. Select the '+' icon to add a validation and select **Query returning row(s)** from the drop down list.



Description of this image (files/tab07_10.txt)

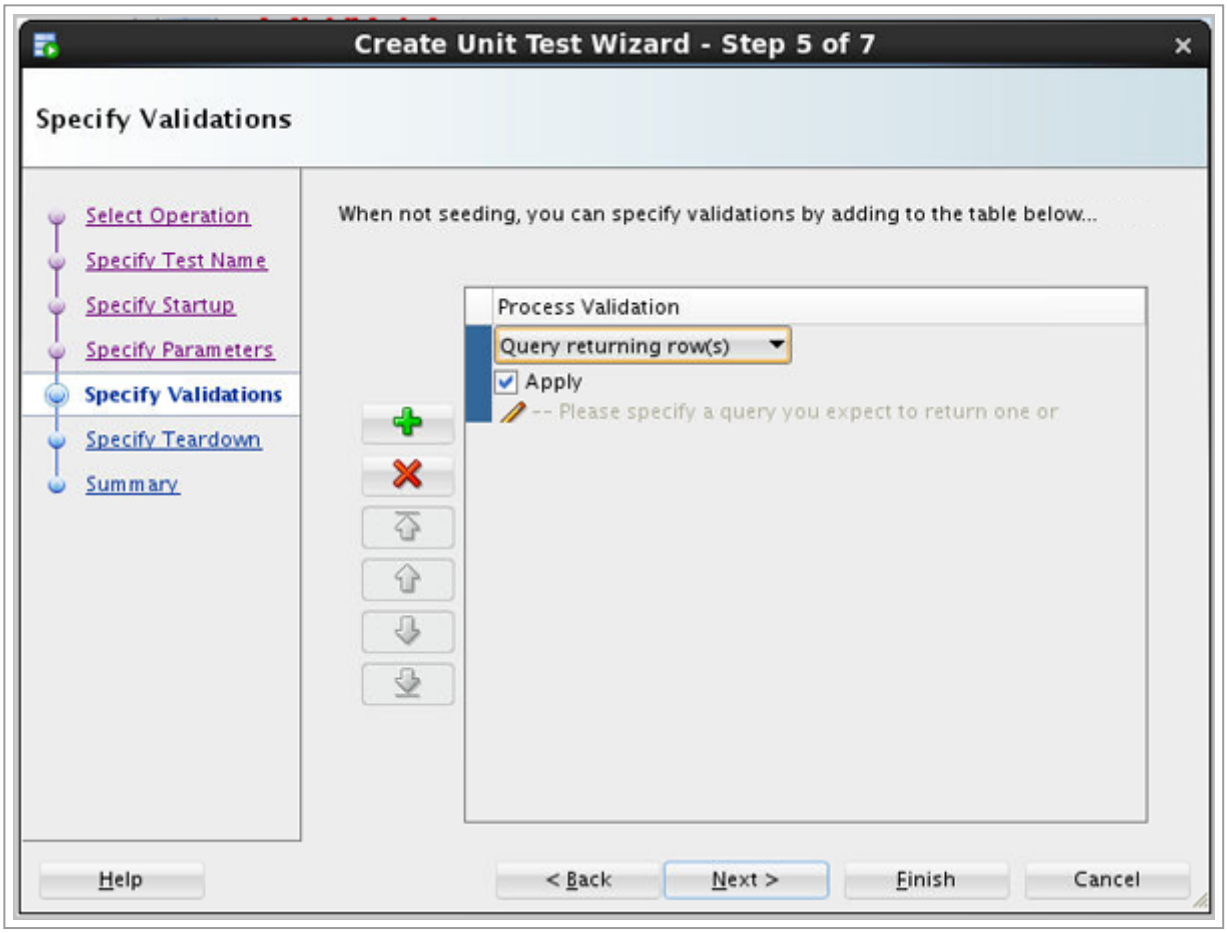
11. Specify the following query and click **OK**. This query will test the results of the change that the unit test performed.

```
SELECT * FROM employees
WHERE employee_id = 149
AND salary = 11200;
```



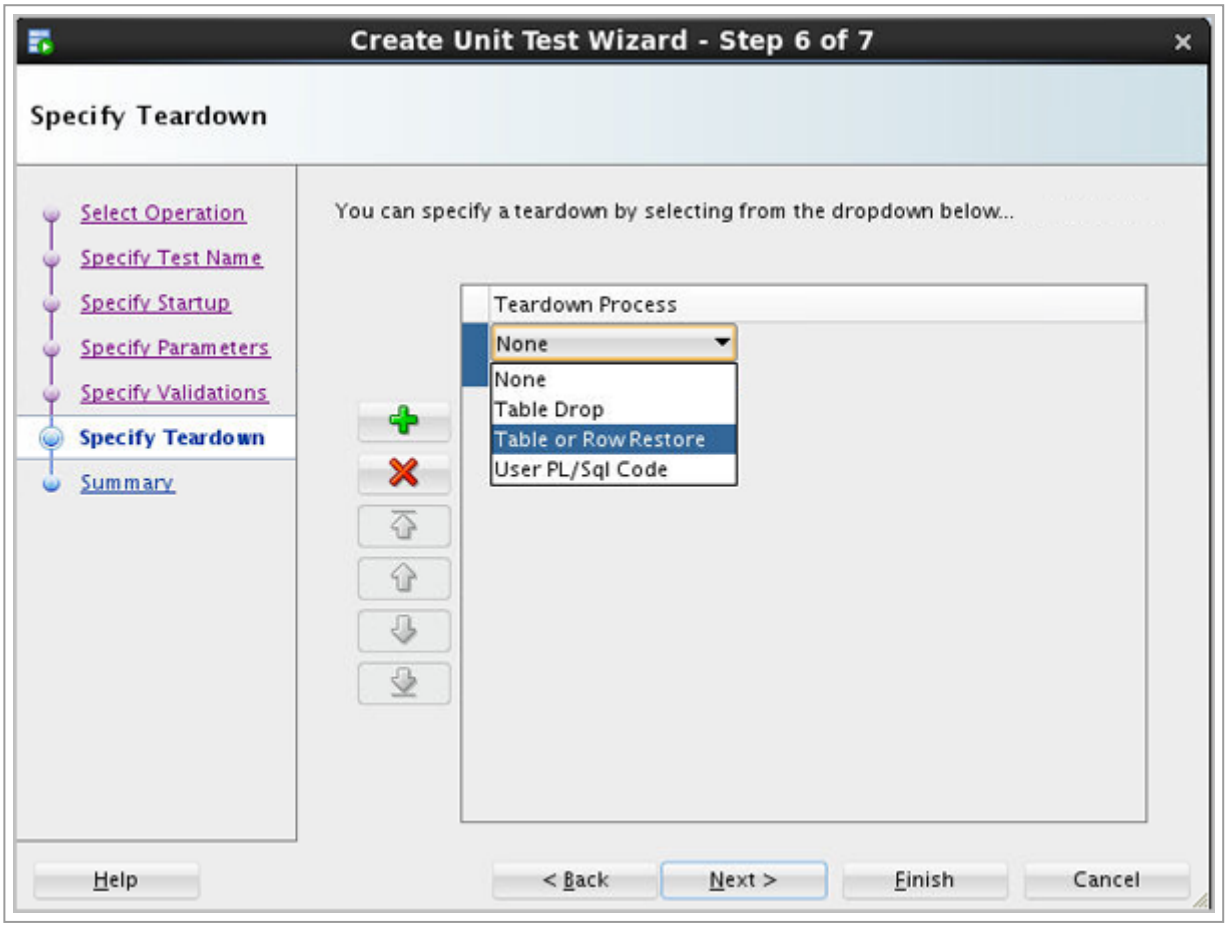
Description of this image (files/tab07_11.txt)

12. Click **Next**.



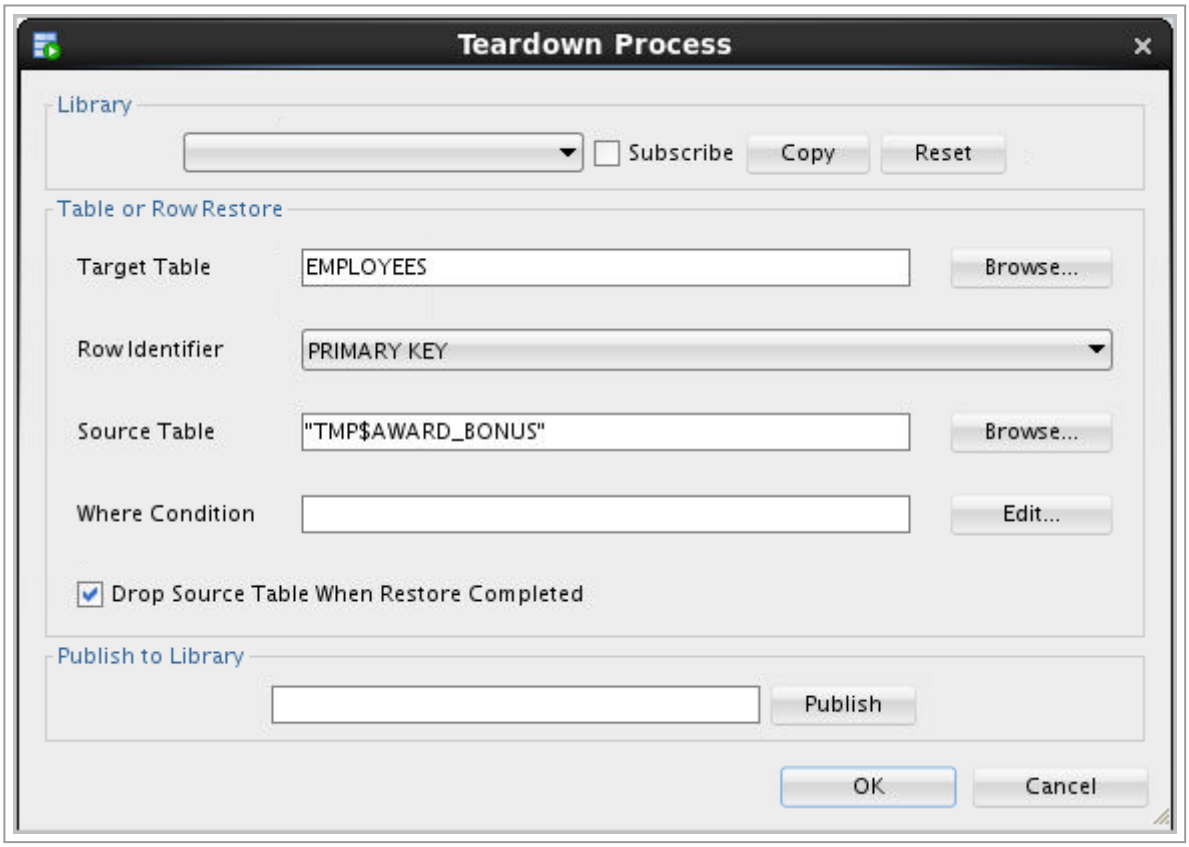
Description of this image (files/tab07_12.txt)

13. In the Specify Teardown window,click the '+' icon and select **Table or Row Restore** from the drop down list.



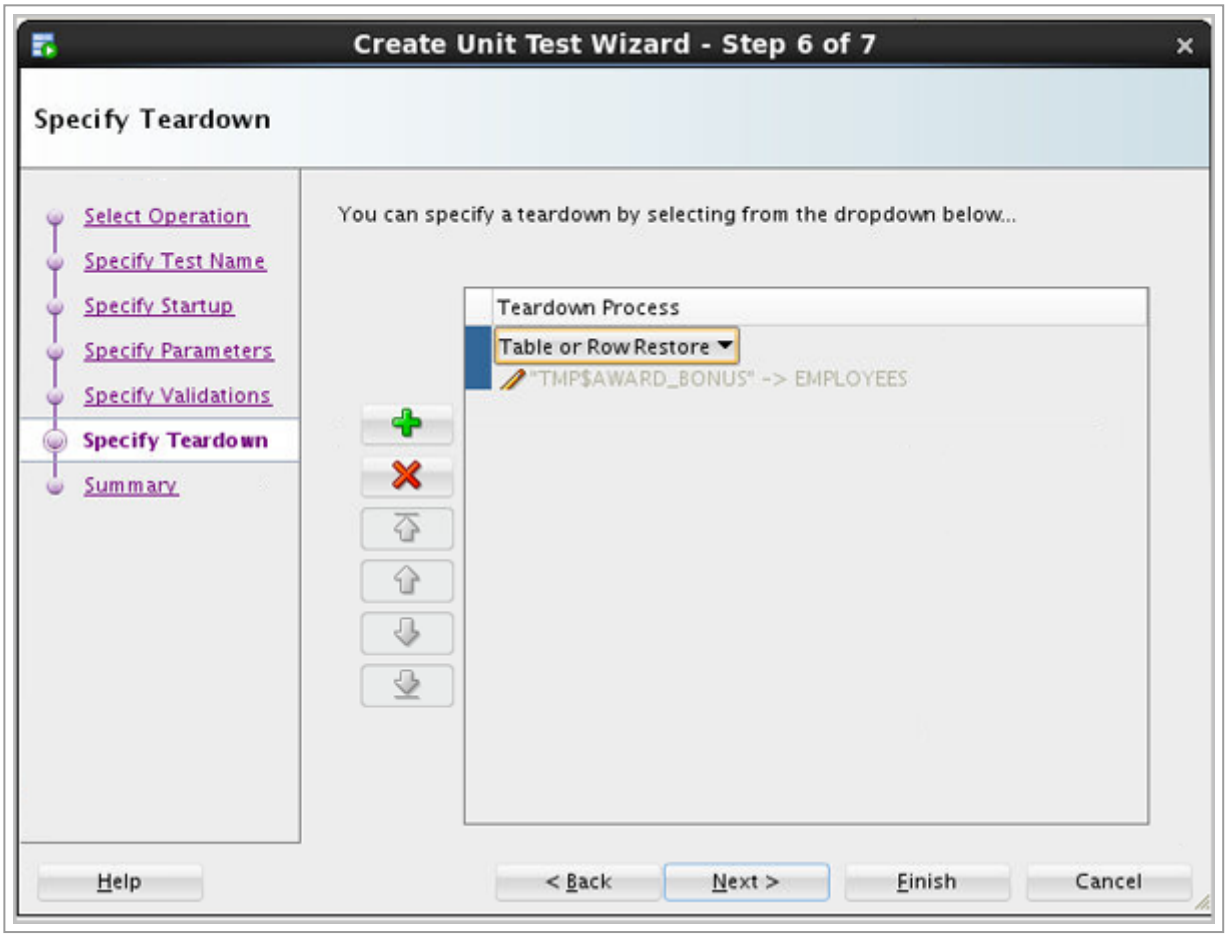
Description of this image (files/tab07_13.txt)

14. Leave the Row Identifier as Primary Key and click **OK**.



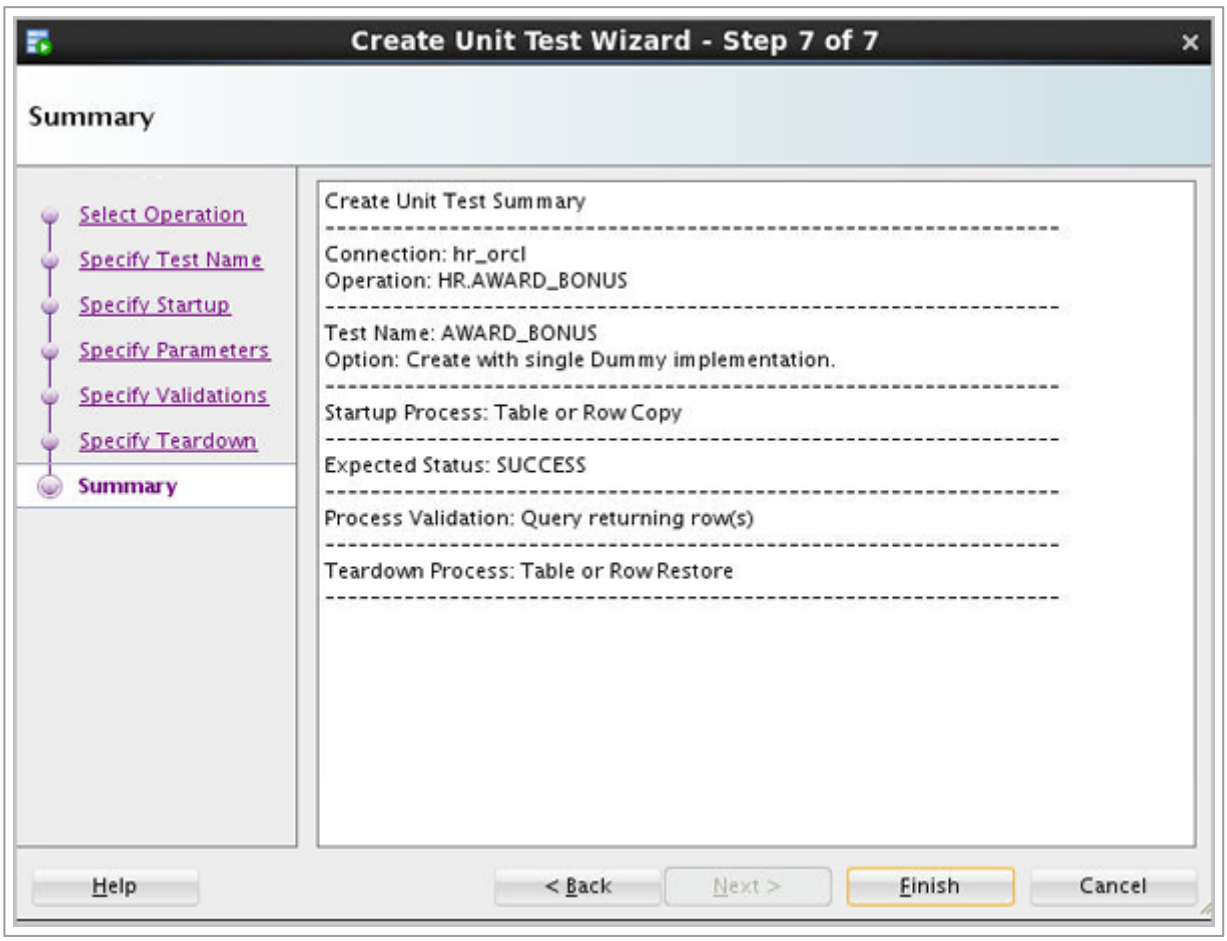
Description of this image (files/tab07_14.txt)

15. Click **Next**.



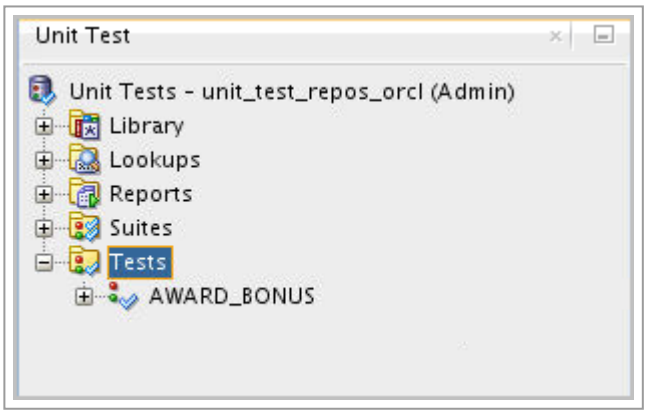
Description of this image (files/tab07_15.txt)

16. Click **Finish**.



Description of this image (files/tab07_16.txt)

17. Expand **Tests**. Your test appears in the list.

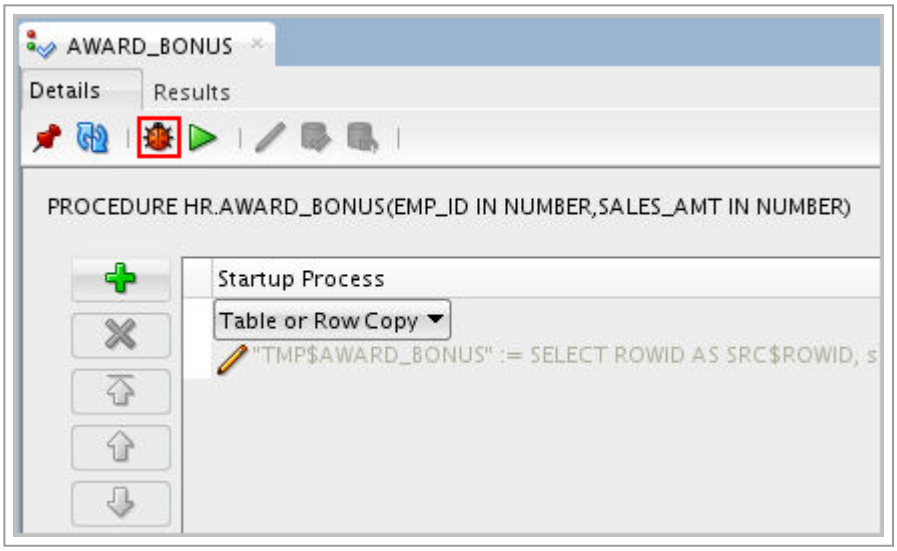


Description of this image (files/tab07_17.txt)

18. Select the **AWARD_BONUS** test in the left navigator. Notice that the test details are displayed on the right panel.

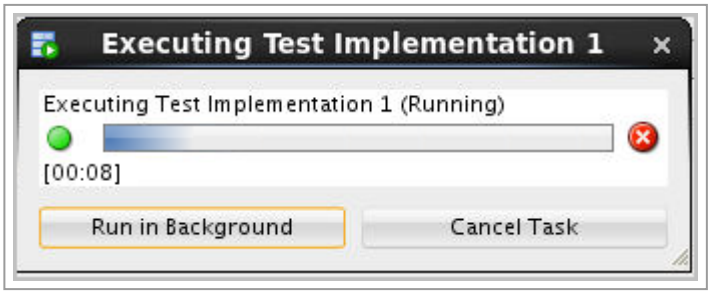
Description of this image (files/tab07_18.txt)

19. Run the test by clicking the **Debug Implementation** icon.



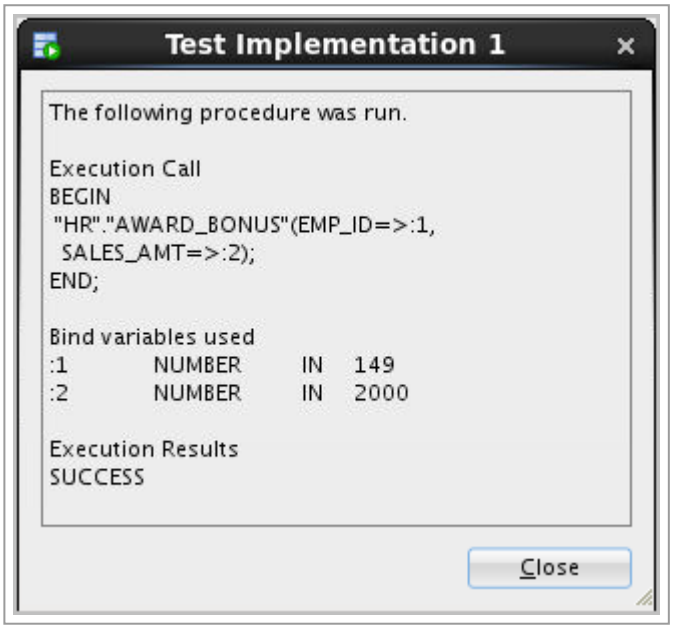
Description of this image (files/tab07_19.txt)

20. The Progress window appears.



Description of this image (files/tab07_20.txt)

21. When the test completes, the results are displayed. Click **Close**.



Description of this image (files/tab07_21.txt)

Want to Learn More?

- Oracle SQL Developer Data Modeler on OTN (http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html)
- Oracle Data Modeling and Relational Database Design course (http://education.oracle.com/pls/web_prod-plq-dad/db_pages.getCourseDesc?dc=D56497GC10&p_org_id=1080544&lang=US)
- Oracle Learning Library (http://www.oracle.com/oll)