

**ORACLE
OPEN
WORLD**

NOVEMBER 11–15, 2007 | MOSCONE CENTER | SAN FRANCISCO

ORACLE DEVELOP

The Premier Conference for Developers

DBA, Java, EJB 3.0, JPA, Forms, SOA, .NET, XML, Security, BPEL, JSF, SQL, PL/SQL, ASP, and AJA
DBA ADVISOR COMMANDS DBA AO AGENTS DBA ADVISOR DEFINITIONS DBA ADVISOR COMMANDS
DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS
DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS
DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS
DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS DBA ADVISOR COMMANDS
Forms, SOA, .NET, XML, Security, BPEL, JSF, SQL, PL/SQL, ASP, and AJA
DBA, Java, EJB 3.0, JPA, Forms, SOA, .NET, XML, Security, BPEL, JSF, SQL, PL/SQL, ASP, and AJA



E V E L O P

JPA
BPEL, JSF, S
Forms, S
SO
Form
JPA, FO
DBA, E
XML, Se
DBA, Java, EJB 3
E
BPEL, JSF, SOA, PL/SQL
DBA, Java, E
.NET, XML, Security, B
DBA, Jav
Forms,
SQL, PL/SC
BPEL, JSF, SQL, PL/SQL, ASP, and AJA
XML, Secu
BPEL, J
EJ
SOA, .NET, XML, Security, BPEL, JSF, SQL, PL/SQL, ASP, and AJA

HANDS-ON LAB

Oracle SQL Developer

SPONSORED BY

Dr.Dobb's

ORACLE®

Using Oracle SQL Developer

Introduction

This hands on lab is designed to allow you to pick and choose the content you are most interested in learning about. Lessons in this lab cover different levels of experience, including those who are new to Oracle SQL Developer, those who want to find out what's new, and those who are experienced users.

Please read through the summaries of the lessons below before choosing which one you would like to start with. An estimate of the time required to complete each lesson is provided.

List of Lessons	Page
Managing Your Database Objects Using Oracle SQL Developer 1.2	5
Developing and Debugging PL/SQL using Oracle SQL Developer	43
Creating XML Extensions using Oracle SQL Developer	67
Migrating a Microsoft Access Database to Oracle	93

Lesson Summaries

Managing Your Database Objects Using Oracle SQL Developer 1.2

Purpose

This tutorial introduces Oracle SQL Developer and shows you how to manage your database objects.

Time to Complete

Approximately 30 minutes

Developing and Debugging PL/SQL using Oracle SQL Developer

Purpose

This tutorial shows you how to create, run, and debug a PL/SQL procedure using Oracle SQL Developer.

Time to Complete

Approximately 30 minutes

Creating XML Extensions using Oracle SQL Developer

Purpose

This tutorial shows you how to add XML extensions to Oracle SQL Developer.

Time to Complete

Approximately 30 minutes

Migrating a Microsoft Access Database to Oracle

Purpose

This tutorial describes how to migrate a Microsoft Access Database to Oracle.

Time to Complete

Approximately 30 minutes

Managing Your Database Using Oracle SQL Developer

Purpose

This tutorial introduces Oracle SQL Developer and shows you how to manage your database objects.

Time to Complete

Approximately 30 minutes

Topics

This tutorial covers the following topics:

- [Overview](#)
- [Prerequisites](#)
- [Creating a Database Connection](#)
- [Adding a New Table Using the Create Table Dialog Box](#)
- [Changing a Table Definition](#)
- [Adding Table Constraints](#)
- [Adding Data to a Table](#)
- [Accessing Data](#)
- [Creating Reports](#)
- [Debugging and Executing PL/SQL](#)
- [Summary](#)

Viewing Screenshots

 Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)

Note: Alternatively, you can place the cursor over an individual icon in the following steps to load and view only the screenshot associated with that step. You can hide an individual screenshot by clicking it.

Overview

Oracle SQL Developer is a free graphical tool that enhances productivity and simplifies database development tasks. Using SQL Developer, users can browse database objects, run SQL statements, edit and debug PL/SQL statements and run reports, whether provided or created.

Developed in Java, SQL Developer runs on Windows, Linux and the Mac OS X. This is a great advantage to the increasing numbers of developers using alternative platforms. Multiple platform support also means that users can install SQL Developer on the Database Server and connect remotely from their desktops, thus avoiding client server network traffic.

Default connectivity to the database is through the JDBC Thin driver, so no Oracle Home is required. To install SQL Developer 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.

[Back to Topic List](#)

Prerequisites

For this Hands On Session, the following has already been installed for you::

1. Oracle Database 10g.
Note: To repeat this exercise later you can use any Oracle Database above 9.2.0.1
2. Oracle SQL Developer 1.2.1.
Note: Oracle SQL Developer is available for download for FREE from [OTN](#). To install Oracle SQL Developer, unzip it into any directory on your machine.

3. The files you use throughout the tutorial are located in the **\SQLDev_HOS\GettingStarted\files** directory. The files have also been included in the [sqldev.zip](#) file.
4. The shipped **HR** schema.
4. There should be no **DEPENDENTS** table. To delete the **DEPENDENTS** table open SQL*Plus and execute the following commands:

```
connect hr/hr  
drop table dependents;
```

[Back to Topic List](#)

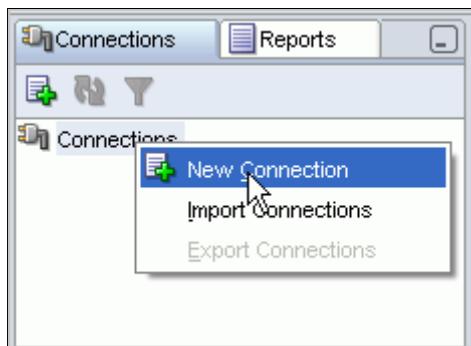
Creating a Database Connection

The first step to managing database objects using Oracle SQL Developer is to create a database connection. Perform the following steps:

1. Open Windows Explorer and double-click **<your_path>\sqldeveloper\sqldeveloper.exe**.

Note: you can also execute sqldeveloper in a non-windows environment.

2. In the Connections tab, right-click **Connections** and select **New Database Connection**.



3. Enter **HR_ORCL** for the Connection Name (or any other name that identifies your connection), **HR** for the Username and Password, specify your **<hostname>** for the Hostname and enter **ORCL** for the SID. Click **Test**.

Connection Name: HR_ORCL
Username: hr
Password: **
 Save Password

Oracle Access MySQL SQLServer

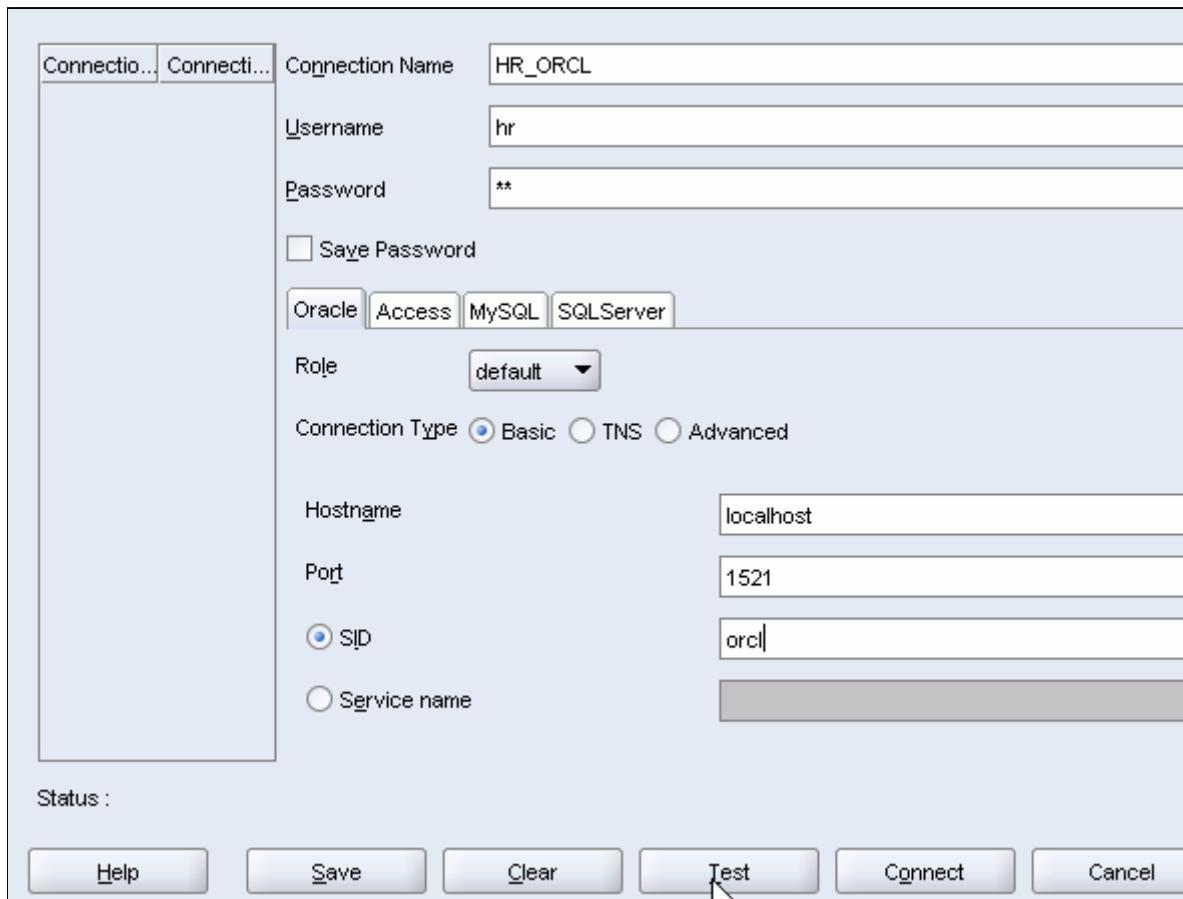
Role: default

Connection Type: Basic TNS Advanced

Hostname: localhost
Port: 1521
 SID: orcl
 Service name

Status:

Help Save Clear Test Connect Cancel



4. The status of the connection was tested successfully. The connection was not saved however. Click **Save** to save the connection, and then click **Connect**.

Connection Name: HR_ORCL
Username: hr
Password: **
 Save Password

Oracle Access MySQL SQLServer

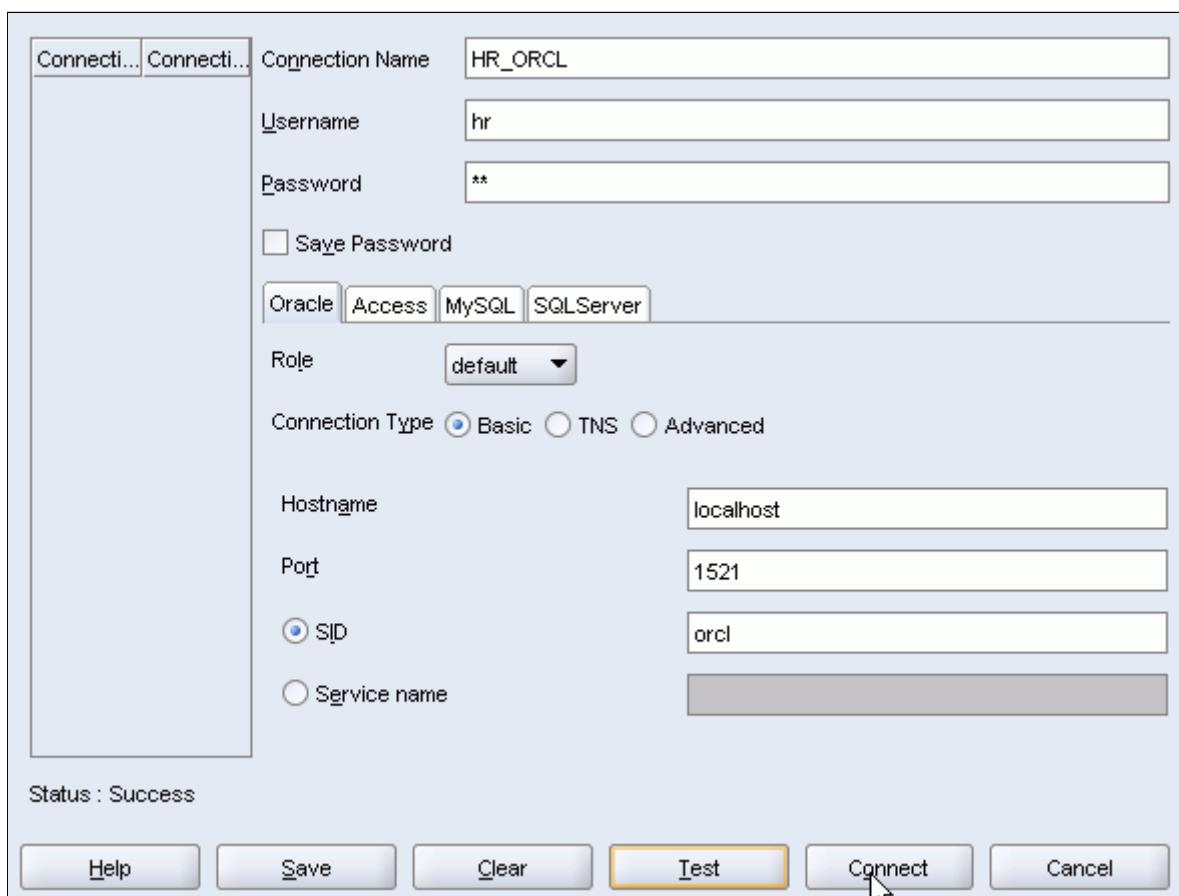
Role: default

Connection Type: Basic TNS Advanced

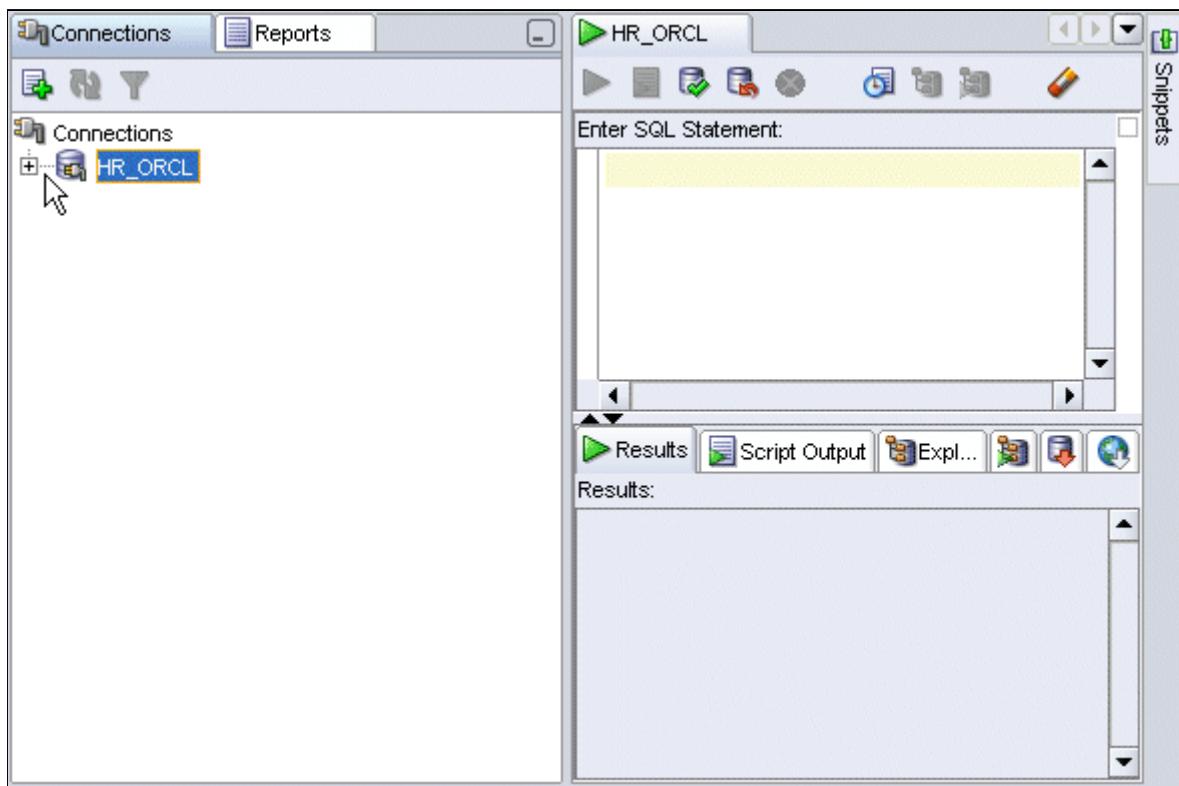
Hostname: localhost
Port: 1521
 SID: orcl
 Service name

Status: Success

Help Save Clear Test Connect Cancel

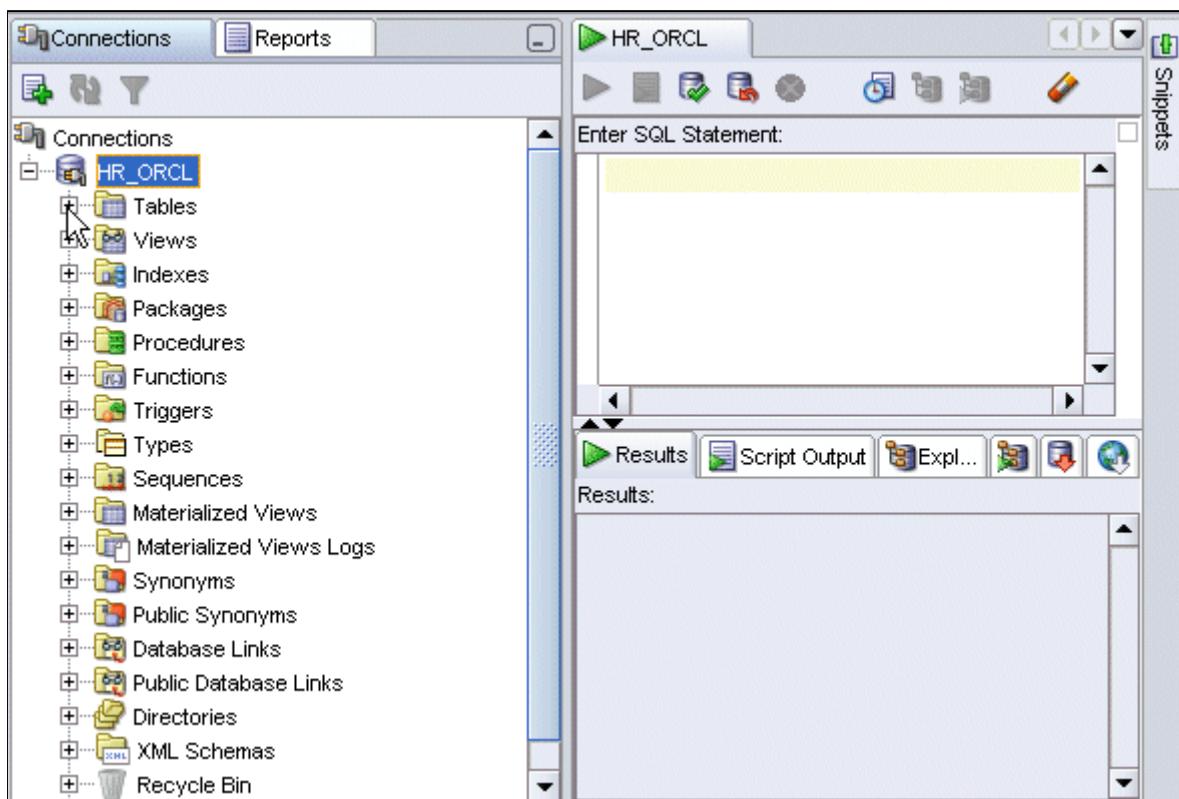


5. The connection was saved and you see the database in the list: **HR_ORCL**.

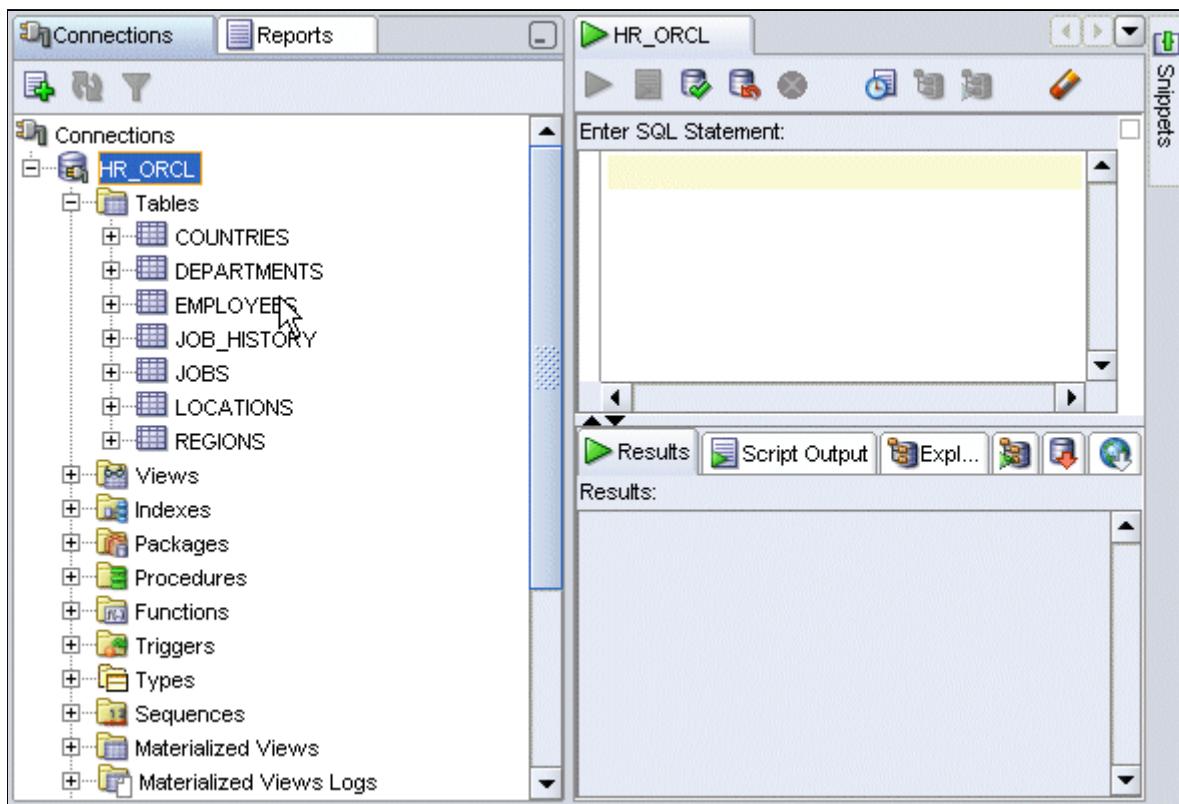


Note: When a connection is opened, a SQL Worksheet is opened automatically. The SQL Worksheet allows you to execute SQL against the connection you just created.

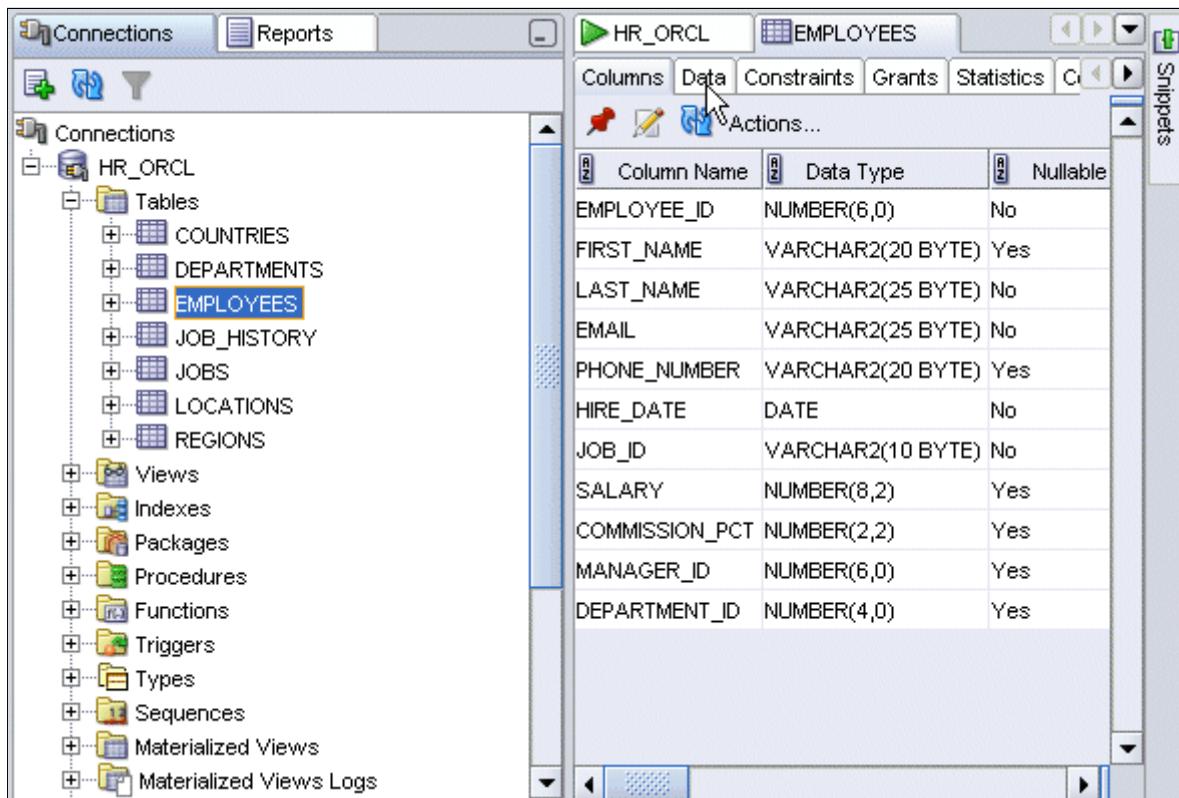
6. Expand **Tables**.



7. Select the Table **EMPLOYEES** to view the table definition.



8. To see the data, click the **Data** tab.



9. The data is shown. In the next topic, you create a new table and populate the table with data.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL'. Under 'Tables', several tables are listed: COUNTRIES, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS. The 'EMPLOYEES' table is selected. The main pane displays the data for the EMPLOYEES table. The first few rows are:

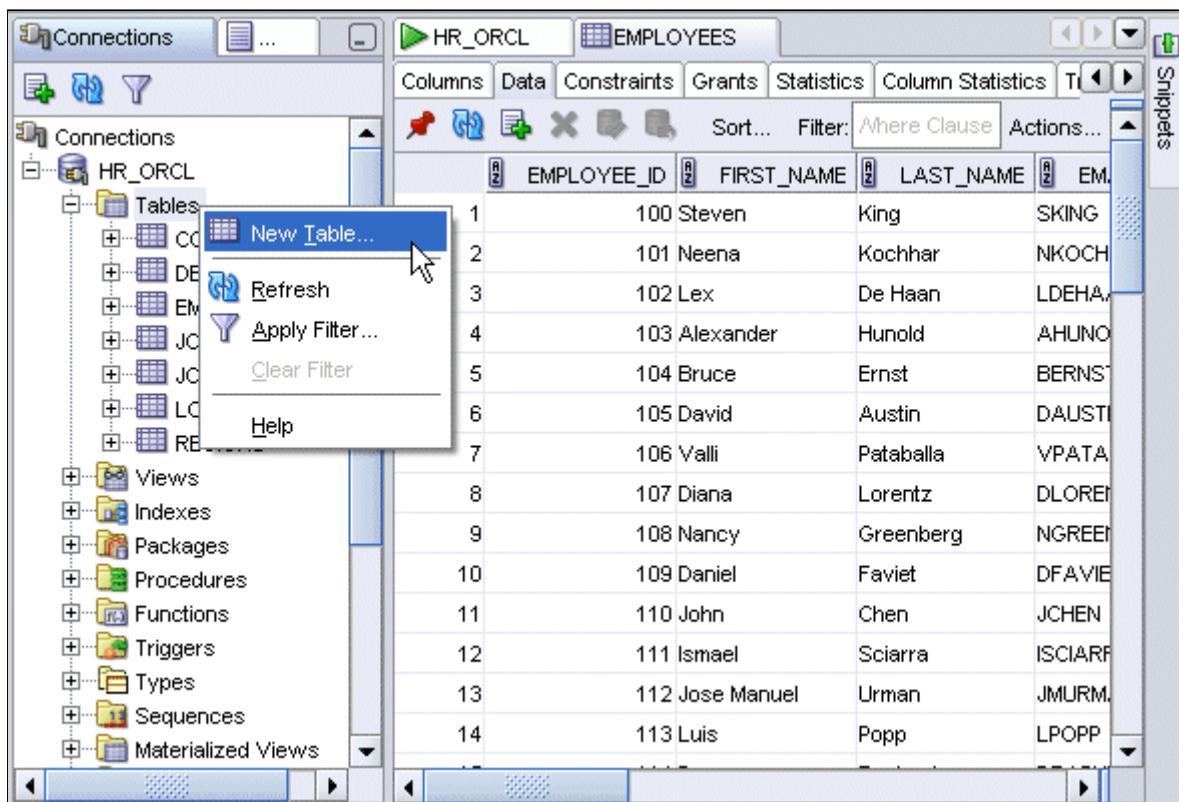
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...
1	Steven	King	SKING
2	Neena	Kochhar	NKOCH
3	Lex	De Haan	LDEHA
4	Alexander	Hunold	AHUNO
5	Bruce	Ernst	BERNST
6	David	Austin	DAUSTI
7	Valli	Pataballa	VPATA
8	Diana	Lorentz	DLORE
9	Nancy	Greenberg	NGREE
10	Daniel	Faviet	DFAVIE
11	John	Chen	JCHEN
12	Ismael	Sciarra	ISCIARR
13	Jose Manuel	Urman	JMURM
14	Luis	Popp	LPOPP

[Back to Topic List](#)

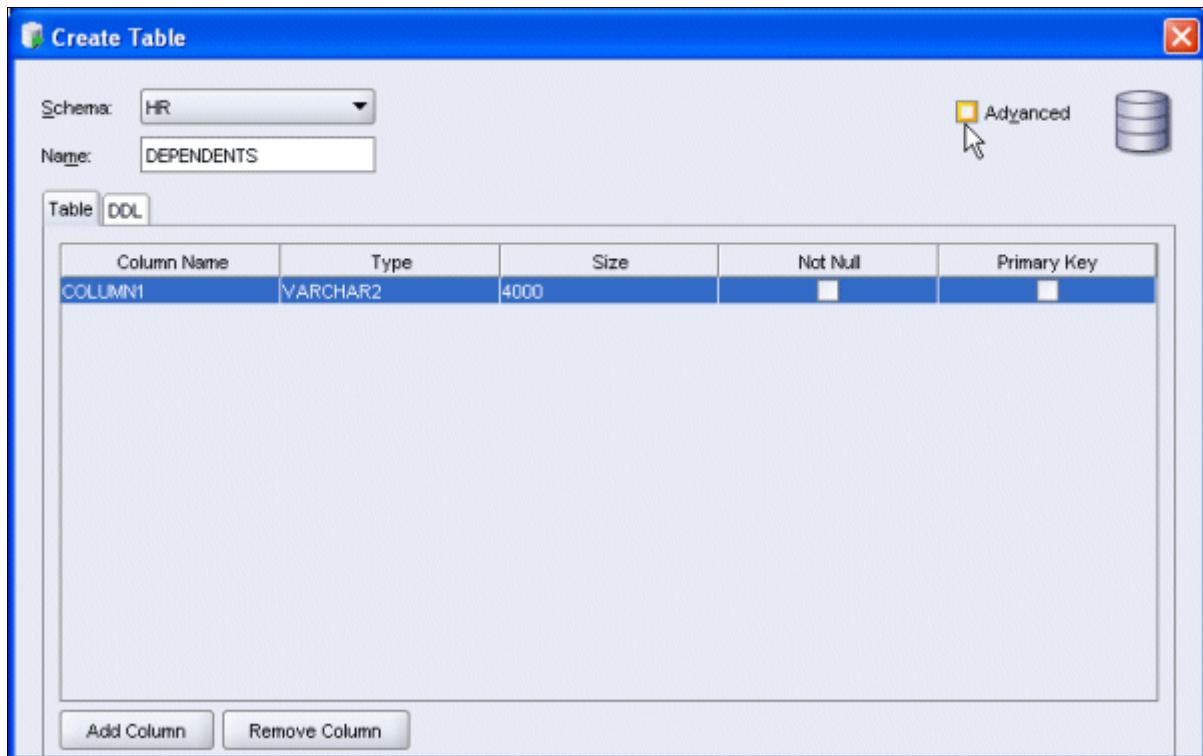
Adding a New Table Using the Create Table Dialog Box

You create a new table called DEPENDENTS which has a foreign key with the EMPLOYEES table. Perform the following steps:

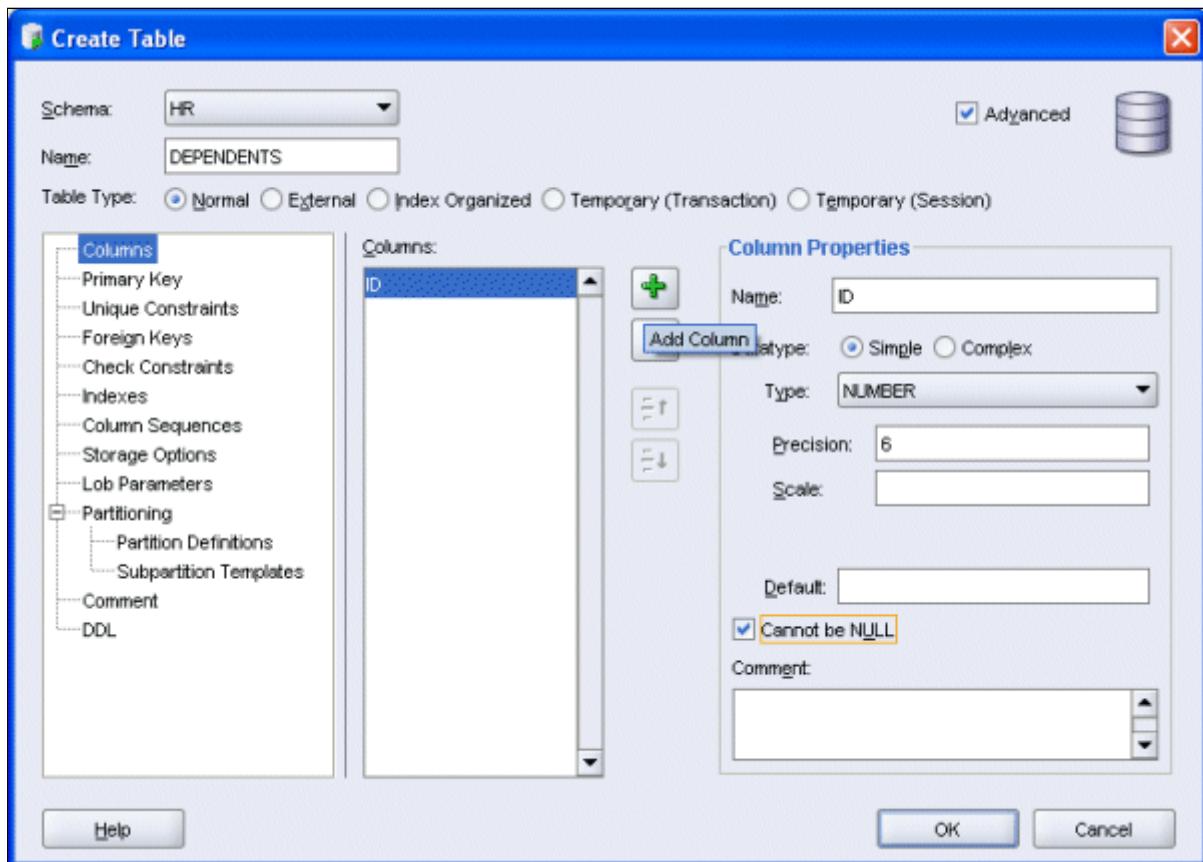
1. Right-click **Tables** and select **New TABLE...**



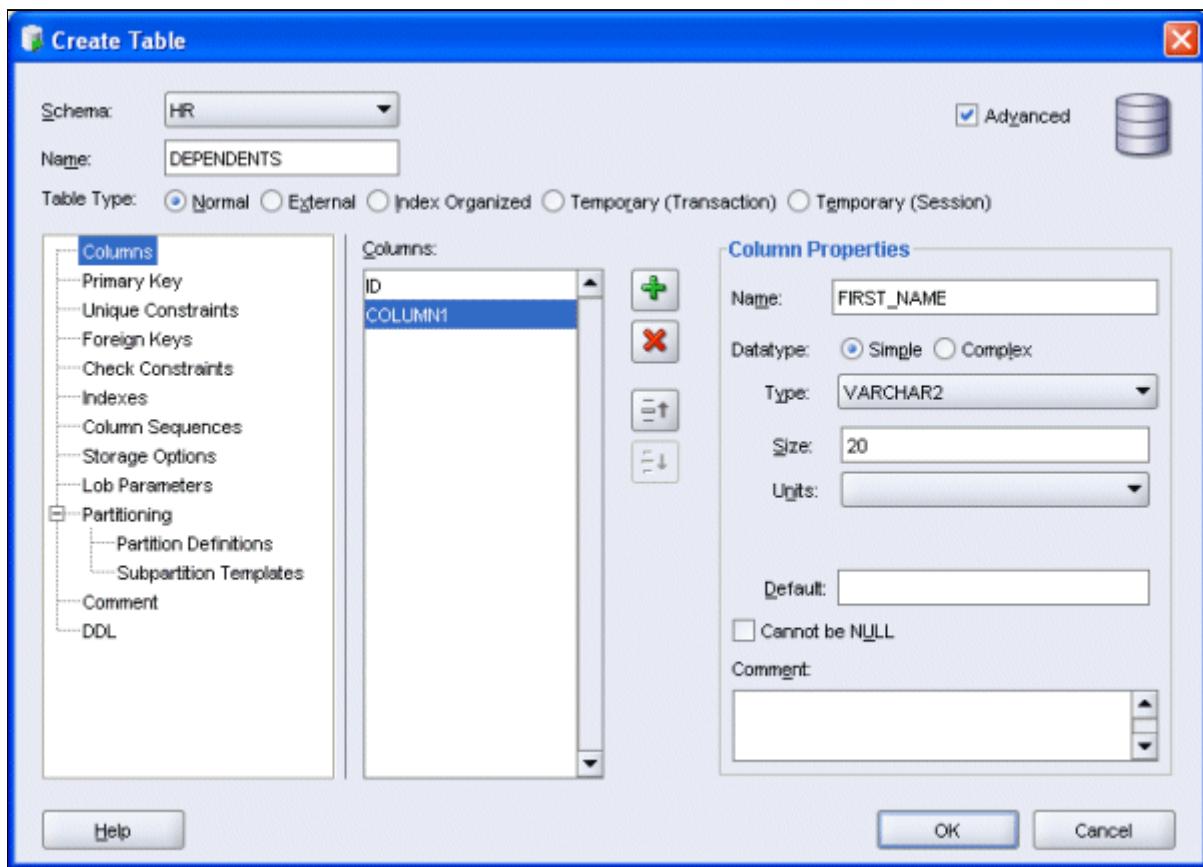
2. Enter **DEPENDENTS** for the Table Name and click the **Advanced** check box.



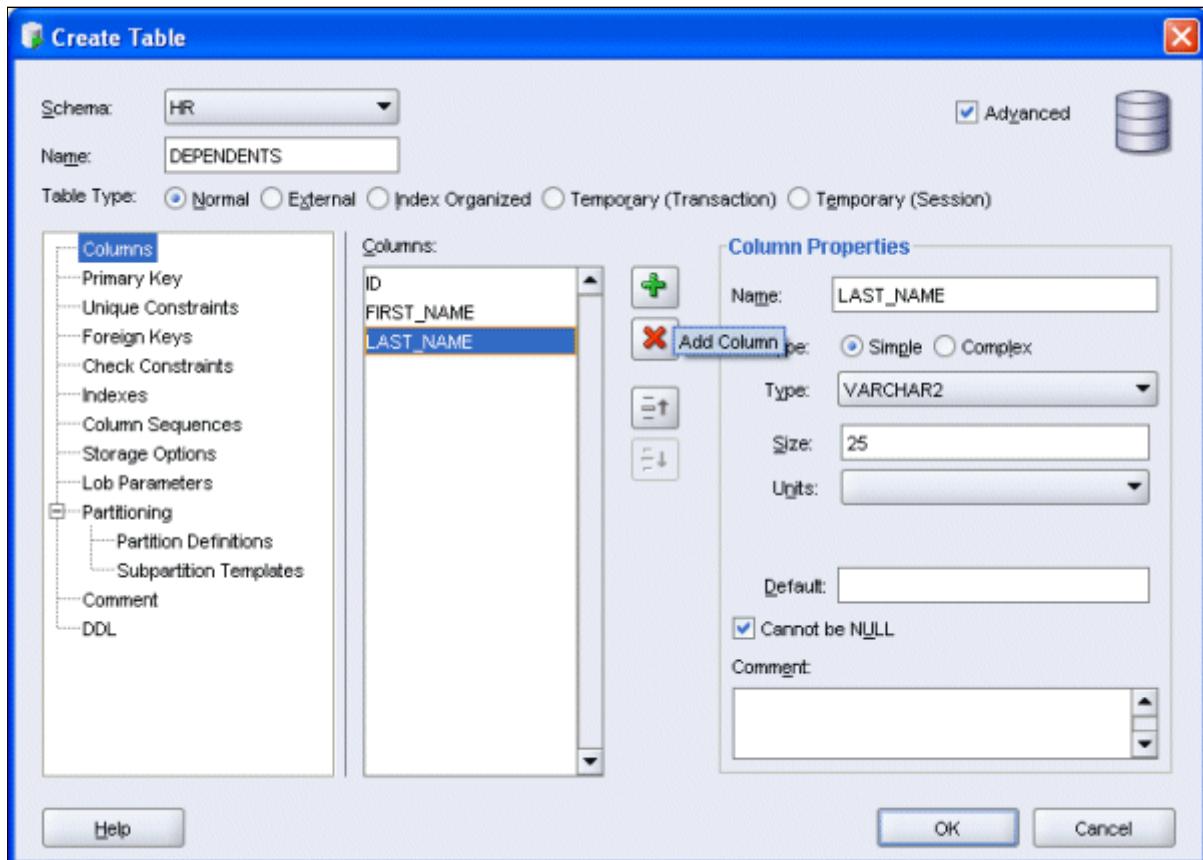
3. Enter **ID** for the Name, select **NUMBER** for the Data type and enter **6** for the Precision. Select the **Cannot be NULL** check box. Then click the **Add Column** icon.



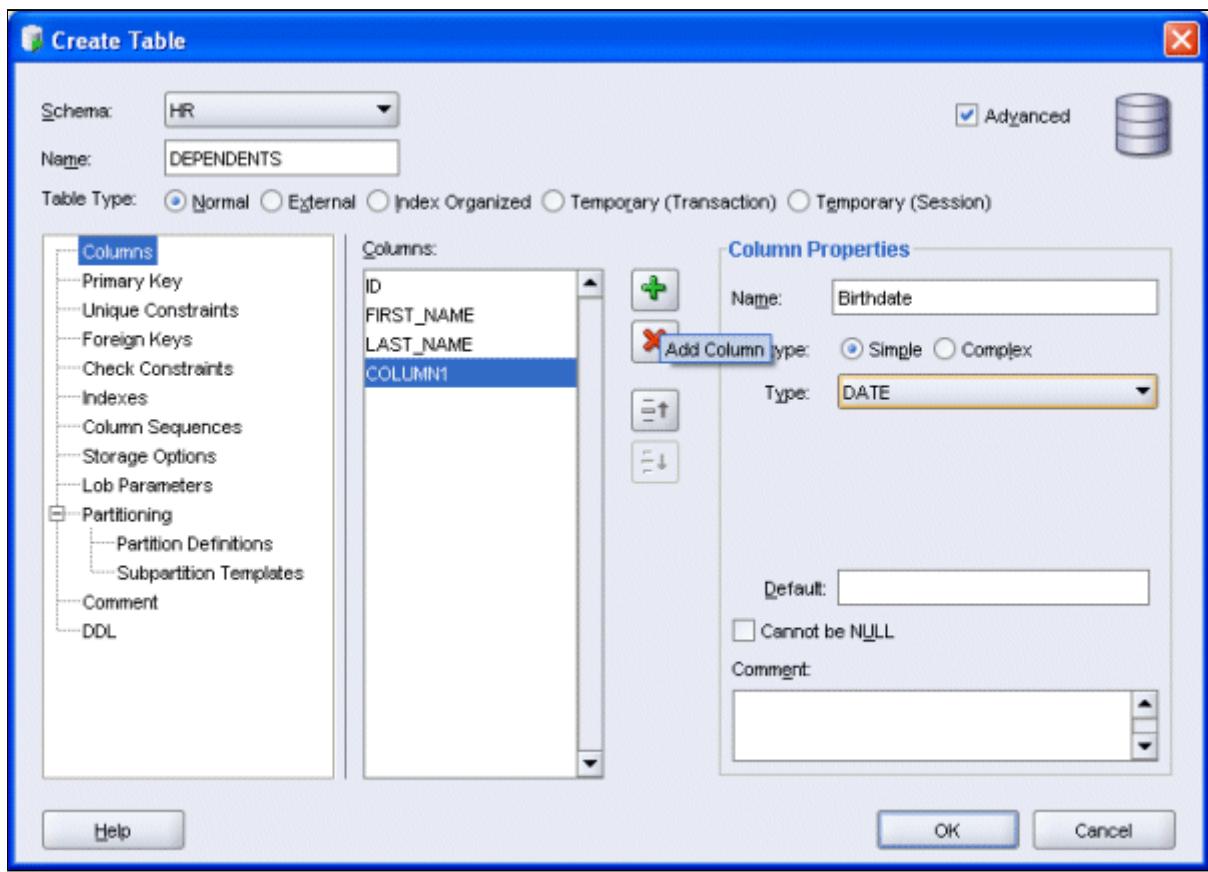
4. Enter **FIRST_NAME** for the Name, and enter **20** for the Size. Then click the **Add Column** icon.



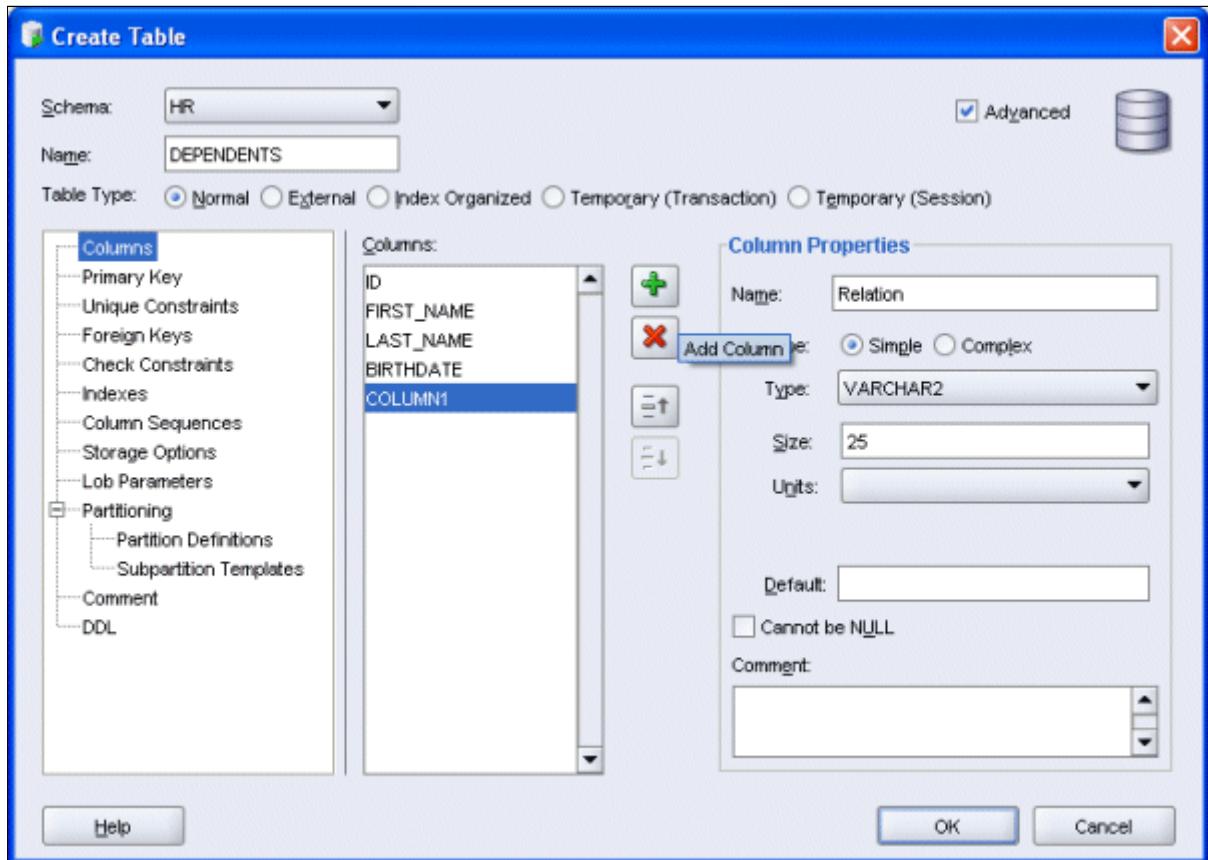
5. Enter **LAST_NAME** for the Name, enter **25** for the Size. Select the **Cannot be NULL** check box. Then click the **Add Column** icon.



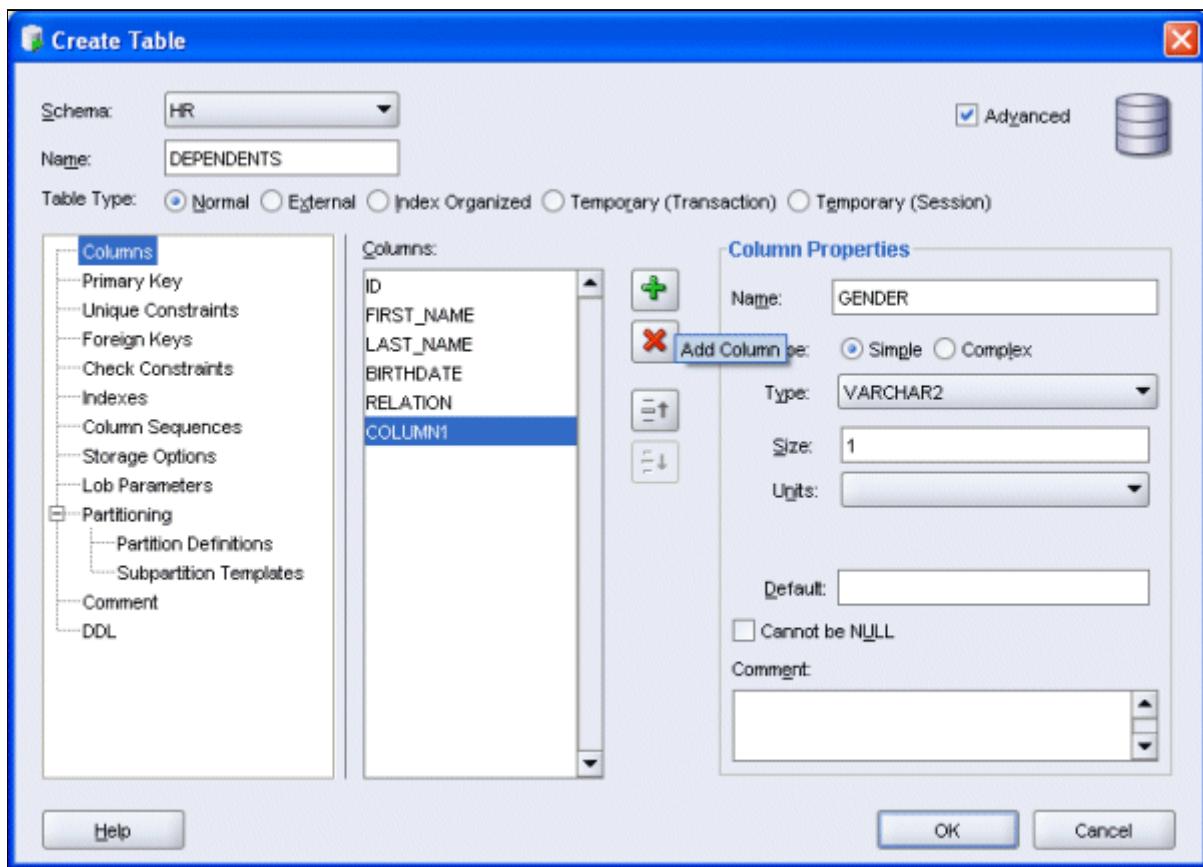
6. Enter **BIRTHDATE** for the Name, select **DATE** for the Data type. Then click the **Add Column** icon.



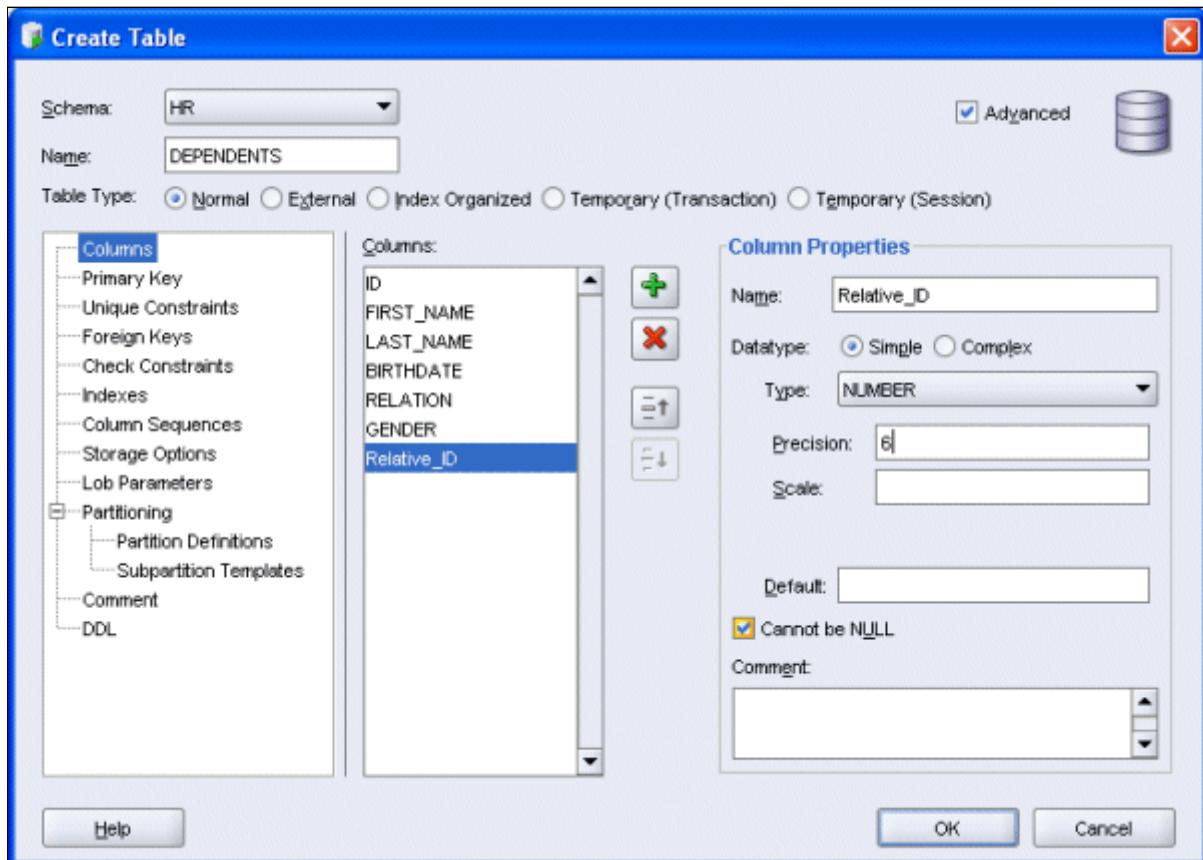
7. Enter **RELATION** for the Name, and enter **25** for the Size. Then click the **Add Column** icon.



8. Enter **GENDER** for the Name, and enter **1** for the Size. Then click the **Add Column** icon.



9. Enter **RELATIVE_ID** for the Name, select **NUMBER** for the Data type, and enter **6** for the Precision. Select the **Cannot be NULL** check box. Then click **OK** to create the table.



10. Your new table appears in the list of tables. Select **DEPENDENTS** from the list.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL' with the 'Tables' node selected. The main pane displays the 'EMPLOYEES' table with the following data:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	100	Steven	King	SKING
2	101	Neena	Kochhar	NKOCH
3	102	Lex	De Haan	LDEHAAN
4	103	Alexander	Hunold	AHUNOLD
5	104	Bruce	Ernst	BERNSTEIN
6	105	David	Austin	DAUSTIN
7	106	Valli	Pataballa	VPATABALLA
8	107	Diana	Lorentz	DLORENTZ
9	108	Nancy	Greenberg	NGREENBERG
10	109	Daniel	Faviet	DFAVIE
11	110	John	Chen	JCHEN
12	111	Ismael	Sciarra	ISCIARRA
13	112	Jose Manuel	Urman	JMURMAN
14	113	Luis	Popp	LPOPP

[Back to Topic List](#)

Changing a Table Definition

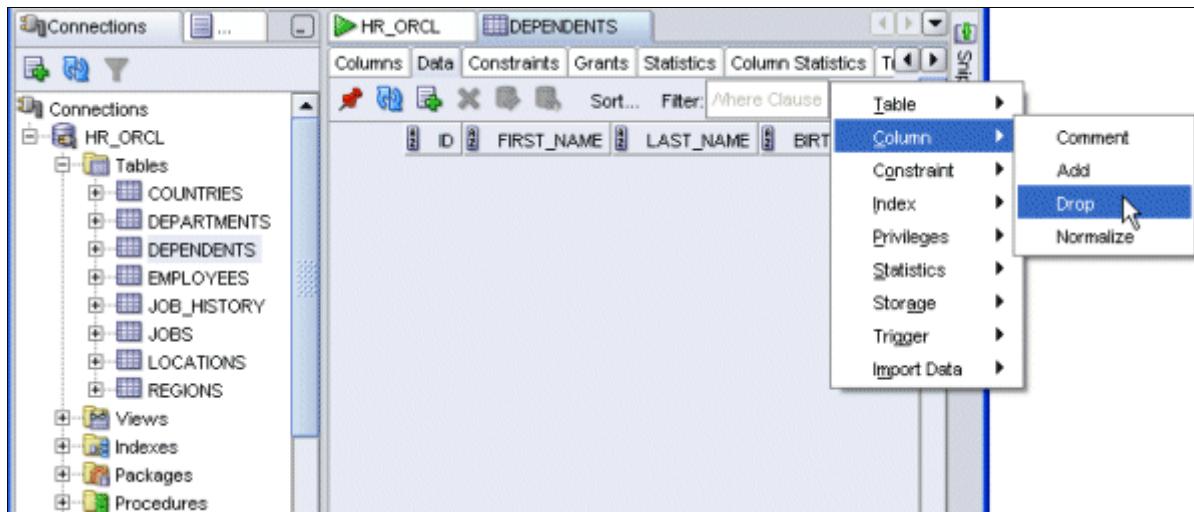
Oracle SQL Developer makes it very easy to make changes to database objects. In this topic, you delete a column in the DEPENDENTS table you just created. Perform the following steps:

1. Click the **Actions...** button.

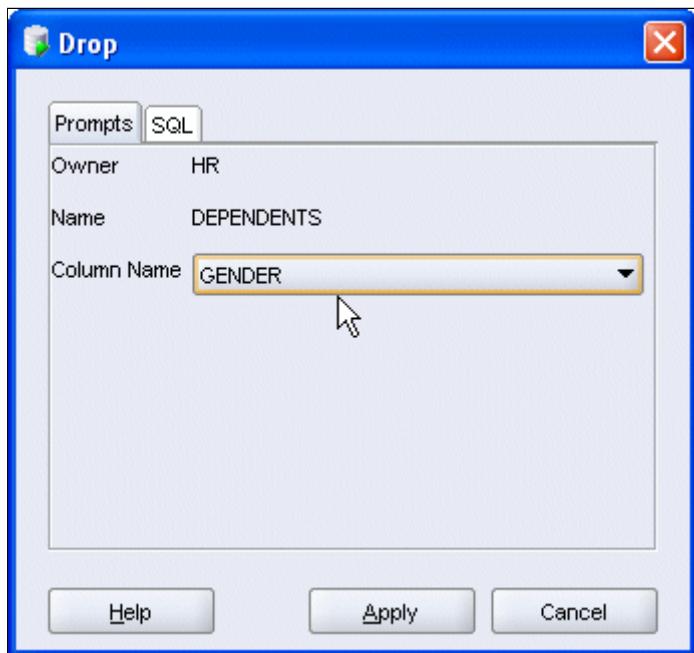
The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL' with the 'Tables' node selected. The main pane displays the 'DEPENDENTS' table with the following data:

	ID	FIRST_NAME	LAST_NAME	BIRTH_DATE
1	101	Patricia	Yap	1953-05-01
2	102	John	Smith	1960-06-20
3	103	Adam	Khalid	1964-05-03
4	104	Susan	Mazlan	1965-09-12
5	105	Mark	Frederick	1967-01-12
6	106	Julia	Frederick	1967-01-12
7	107	Michael	Frederick	1967-01-12
8	108	Howard	Frederick	1967-01-12
9	109	Robert	Frederick	1967-01-12
10	110	Patricia	Frederick	1967-01-12
11	111	John	Frederick	1967-01-12
12	112	Adam	Frederick	1967-01-12
13	113	Susan	Frederick	1967-01-12
14	114	Mark	Frederick	1967-01-12
15	115	Julia	Frederick	1967-01-12
16	116	Michael	Frederick	1967-01-12
17	117	Howard	Frederick	1967-01-12
18	118	Robert	Frederick	1967-01-12
19	119	Patricia	Frederick	1967-01-12
20	120	John	Frederick	1967-01-12
21	121	Adam	Frederick	1967-01-12
22	122	Susan	Frederick	1967-01-12
23	123	Mark	Frederick	1967-01-12
24	124	Julia	Frederick	1967-01-12
25	125	Michael	Frederick	1967-01-12
26	126	Howard	Frederick	1967-01-12
27	127	Robert	Frederick	1967-01-12
28	128	Patricia	Frederick	1967-01-12
29	129	John	Frederick	1967-01-12
30	130	Adam	Frederick	1967-01-12
31	131	Susan	Frederick	1967-01-12
32	132	Mark	Frederick	1967-01-12
33	133	Julia	Frederick	1967-01-12
34	134	Michael	Frederick	1967-01-12
35	135	Howard	Frederick	1967-01-12
36	136	Robert	Frederick	1967-01-12
37	137	Patricia	Frederick	1967-01-12
38	138	John	Frederick	1967-01-12
39	139	Adam	Frederick	1967-01-12
40	140	Susan	Frederick	1967-01-12
41	141	Mark	Frederick	1967-01-12
42	142	Julia	Frederick	1967-01-12
43	143	Michael	Frederick	1967-01-12
44	144	Howard	Frederick	1967-01-12
45	145	Robert	Frederick	1967-01-12
46	146	Patricia	Frederick	1967-01-12
47	147	John	Frederick	1967-01-12
48	148	Adam	Frederick	1967-01-12
49	149	Susan	Frederick	1967-01-12
50	150	Mark	Frederick	1967-01-12
51	151	Julia	Frederick	1967-01-12
52	152	Michael	Frederick	1967-01-12
53	153	Howard	Frederick	1967-01-12
54	154	Robert	Frederick	1967-01-12
55	155	Patricia	Frederick	1967-01-12
56	156	John	Frederick	1967-01-12
57	157	Adam	Frederick	1967-01-12
58	158	Susan	Frederick	1967-01-12
59	159	Mark	Frederick	1967-01-12
60	160	Julia	Frederick	1967-01-12
61	161	Michael	Frederick	1967-01-12
62	162	Howard	Frederick	1967-01-12
63	163	Robert	Frederick	1967-01-12
64	164	Patricia	Frederick	1967-01-12
65	165	John	Frederick	1967-01-12
66	166	Adam	Frederick	1967-01-12
67	167	Susan	Frederick	1967-01-12
68	168	Mark	Frederick	1967-01-12
69	169	Julia	Frederick	1967-01-12
70	170	Michael	Frederick	1967-01-12
71	171	Howard	Frederick	1967-01-12
72	172	Robert	Frederick	1967-01-12
73	173	Patricia	Frederick	1967-01-12
74	174	John	Frederick	1967-01-12
75	175	Adam	Frederick	1967-01-12
76	176	Susan	Frederick	1967-01-12
77	177	Mark	Frederick	1967-01-12
78	178	Julia	Frederick	1967-01-12
79	179	Michael	Frederick	1967-01-12
80	180	Howard	Frederick	1967-01-12
81	181	Robert	Frederick	1967-01-12
82	182	Patricia	Frederick	1967-01-12
83	183	John	Frederick	1967-01-12
84	184	Adam	Frederick	1967-01-12
85	185	Susan	Frederick	1967-01-12
86	186	Mark	Frederick	1967-01-12
87	187	Julia	Frederick	1967-01-12
88	188	Michael	Frederick	1967-01-12
89	189	Howard	Frederick	1967-01-12
90	190	Robert	Frederick	1967-01-12
91	191	Patricia	Frederick	1967-01-12
92	192	John	Frederick	1967-01-12
93	193	Adam	Frederick	1967-01-12
94	194	Susan	Frederick	1967-01-12
95	195	Mark	Frederick	1967-01-12
96	196	Julia	Frederick	1967-01-12
97	197	Michael	Frederick	1967-01-12
98	198	Howard	Frederick	1967-01-12
99	199	Robert	Frederick	1967-01-12
100	200	Patricia	Frederick	1967-01-12
101	201	John	Frederick	1967-01-12
102	202	Adam	Frederick	1967-01-12
103	203	Susan	Frederick	1967-01-12
104	204	Mark	Frederick	1967-01-12
105	205	Julia	Frederick	1967-01-12
106	206	Michael	Frederick	1967-01-12
107	207	Howard	Frederick	1967-01-12
108	208	Robert	Frederick	1967-01-12
109	209	Patricia	Frederick	1967-01-12
110	210	John	Frederick	1967-01-12
111	211	Adam	Frederick	1967-01-12
112	212	Susan	Frederick	1967-01-12
113	213	Mark	Frederick	1967-01-12
114	214	Julia	Frederick	1967-01-12
115	215	Michael	Frederick	1967-01-12
116	216	Howard	Frederick	1967-01-12
117	217	Robert	Frederick	1967-01-12
118	218	Patricia	Frederick	1967-01-12
119	219	John	Frederick	1967-01-12
120	220	Adam	Frederick	1967-01-12
121	221	Susan	Frederick	1967-01-12
122	222	Mark	Frederick	1967-01-12
123	223	Julia	Frederick	1967-01-12
124	224	Michael	Frederick	1967-01-12
125	225	Howard	Frederick	1967-01-12
126	226	Robert	Frederick	1967-01-12
127	227	Patricia	Frederick	1967-01-12
128	228	John	Frederick	1967-01-12
129	229	Adam	Frederick	1967-01-12
130	230	Susan	Frederick	1967-01-12
131	231	Mark	Frederick	1967-01-12
132	232	Julia	Frederick	1967-01-12
133	233	Michael	Frederick	1967-01-12
134	234	Howard	Frederick	1967-01-12
135	235	Robert	Frederick	1967-01-12
136	236	Patricia	Frederick	1967-01-12
137	237	John	Frederick	1967-01-12
138	238	Adam	Frederick	1967-01-12
139	239	Susan	Frederick	1967-01-12
140	240	Mark	Frederick	1967-01-12
141	241	Julia	Frederick	1967-01-12
142	242	Michael	Frederick	1967-01-12
143	243	Howard	Frederick	1967-01-12
144	244	Robert	Frederick	1967-01-12
145	245	Patricia	Frederick	1967-01-12
146	246	John	Frederick	1967-01-12
147	247	Adam	Frederick	1967-01-12
148	248	Susan	Frederick	1967-01-12
149	249	Mark	Frederick	1967-01-12
150	250	Julia	Frederick	1967-01-12
151	251	Michael	Frederick	1967-01-12
152	252	Howard	Frederick	1967-01-12
153	253	Robert	Frederick	1967-01-12
154	254	Patricia	Frederick	1967-01-12
155	255	John	Frederick	1967-01-12
156	256	Adam	Frederick	1967-01-12
157	257	Susan	Frederick	1967-01-12
158	258	Mark	Frederick	1967-01-12
159	259	Julia	Frederick	1967-01-12
160	260	Michael	Frederick	1967-01-12
161	261	Howard	Frederick	1967-01-12
162	262	Robert	Frederick	1967-01-12
163	263	Patricia	Frederick	1967-01-12
164	264	John	Frederick	1967-01-12
165	265	Adam	Frederick	1967-01-12
166	266	Susan	Frederick	1967-01-12
167	267	Mark	Frederick	1967-01-12
168	268	Julia	Frederick	1967-01-12
169	269	Michael	Frederick	1967-01-12
170	270	Howard	Frederick	1967-01-12
171	271	Robert	Frederick	1967-01-12
172	272	Patricia	Frederick	1967-01-12
173	273	John	Frederick	1967-01-12
174	274	Adam	Frederick	1967-01-12
175	275	Susan	Frederick	1967-01-12
176	276	Mark	Frederick	1967-01-12
177	277	Julia	Frederick	1967-01-12
178	278	Michael	Frederick	1967-01-12
179	279	Howard	Frederick	1967-01-12
180	280	Robert	Frederick	1967-01-12
181	281	Patricia	Frederick	1967-01-12
182	282	John	Frederick	1967-01-12
183	283	Adam	Frederick	1967-01-12
184	284	Susan	Frederick	1967-01-12
185	285	Mark	Frederick	1967-01-12
186	286	Julia	Frederick	1967-01-12
187	287	Michael	Frederick	1967-01-12
188	288	Howard	Frederick	1967-01-12
189	289	Robert	Frederick	1967-01-12
190	290	Patricia	Frederick	1967-01-12
191	291	John	Frederick	1967-01-12
192	292	Adam	Frederick	1967-01-12
193	293	Susan	Frederick	1967-01-12
194	294	Mark	Frederick	1967-01-12
195	295	Julia	Frederick	1967-01-12
196	296	Michael	Frederick	1967-01-12
197	297	Howard	Frederick	1967-01-12
198	298	Robert	Frederick	1967-01-12
199	299	Patricia	Frederick	1967-01-12
200	300	John	Frederick	1967-01-12

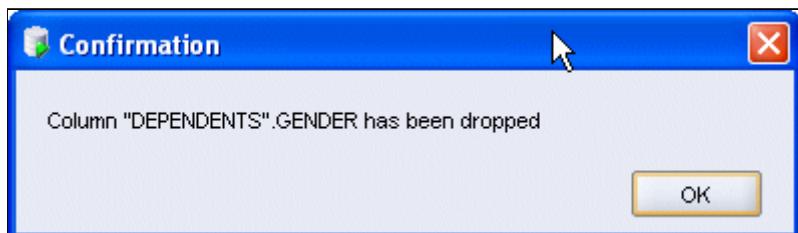
2. Select **Column** then **Drop**.



3. Select the Column Name **GENDER** and click **Apply**.



4. The column has been dropped. Click **OK**.



5. Right-click the **tables** node for the **HR_ORCL** Database Connection and click **Refresh** to refresh the Database Connection.

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'Connections' is selected. Below it, the 'HR_ORCL' connection is chosen. The main area displays the 'DEPENDENTS' table's columns: ID (NUMBER(6,0)), FIRST_NAME (VARCHAR2(20 BYTE)), LAST_NAME (VARCHAR2(25 BYTE)), BIRTHDATE (DATE), RELATION (VARCHAR2(25 BYTE)), and RELATIVE_ID (NUMBER(6,0)). A context menu is open over the 'Tables' node in the Connections pane, with 'Refresh' highlighted.

6. Expand **HR_ORCL > Tables**, and select the DEPENDENTS table.

The screenshot shows the Oracle SQL Developer interface with the 'Tables' node expanded under 'HR_ORCL'. The 'DEPENDENTS' table is selected and highlighted with a blue border. The table's columns are listed in the main pane: ID (NUMBER(6,0)), FIRST_NAME (VARCHAR2(20 BYTE)), LAST_NAME (VARCHAR2(25 BYTE)), BIRTHDATE (DATE), RELATION (VARCHAR2(25 BYTE)), and RELATIVE_ID (NUMBER(6,0)).

[Back to Topic List](#)

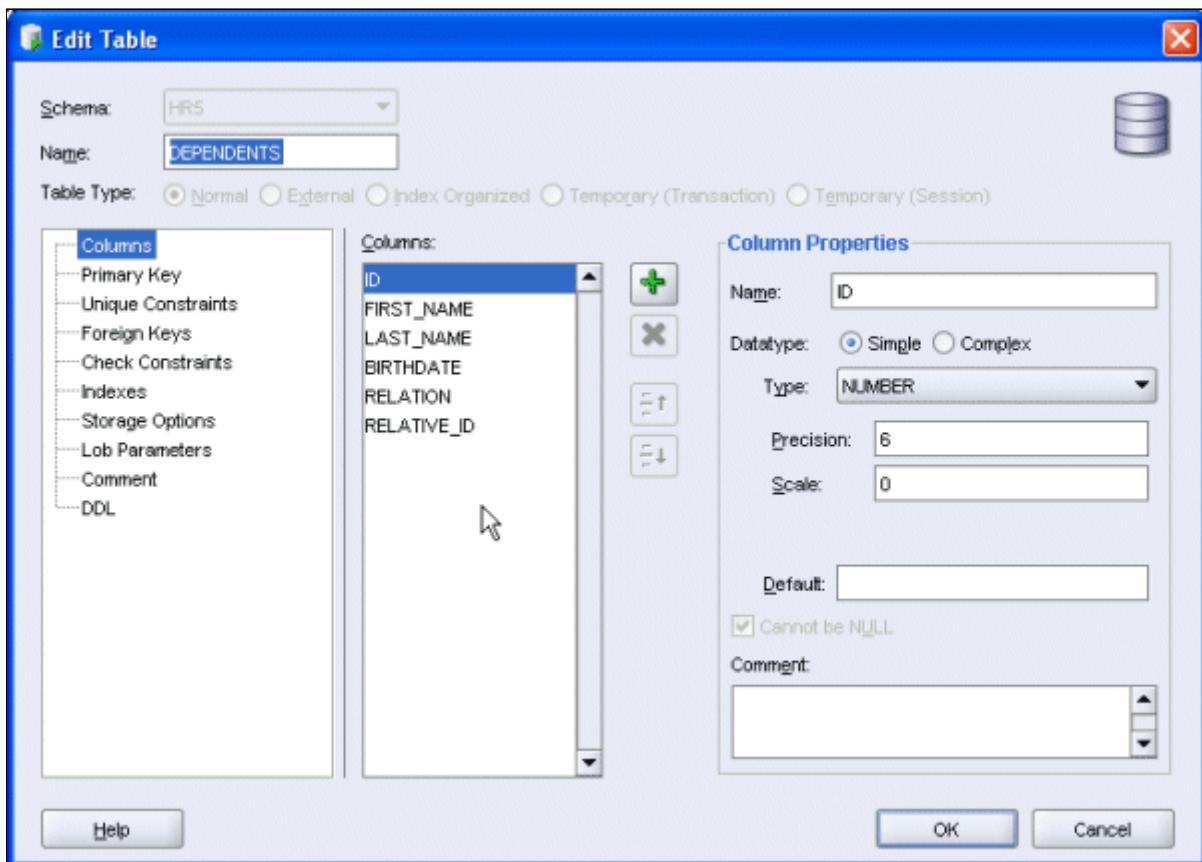
Adding Table Constraints

In this topic, you create the Primary and Foreign Key Constraints for the DEPENDENTS table. Perform the following steps:

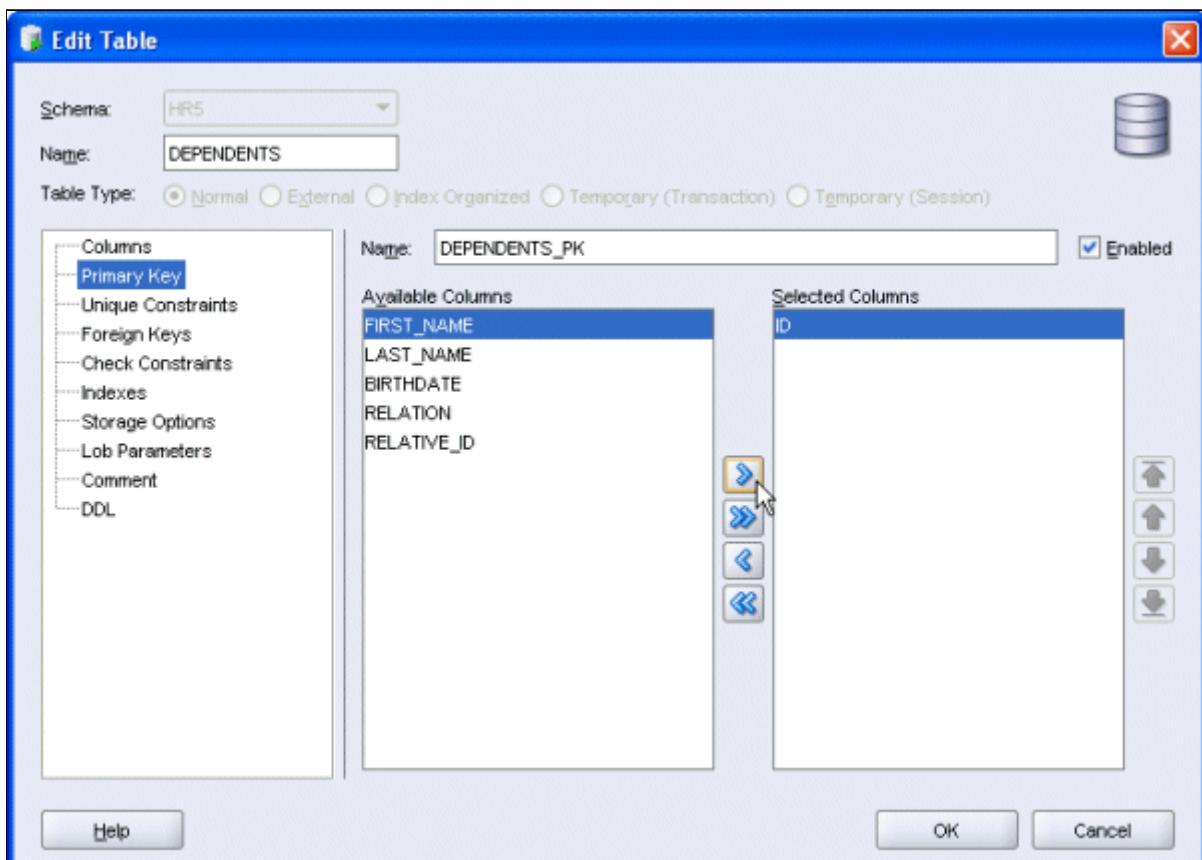
1. Right-click **DEPENDENTS** table and select **Edit...**

The screenshot shows the Oracle SQL Developer interface with the 'DEPENDENTS' table selected. A context menu is open, and 'Edit...' is highlighted. Other options in the menu include 'Table', 'Column', 'Constraint', and 'Index'.

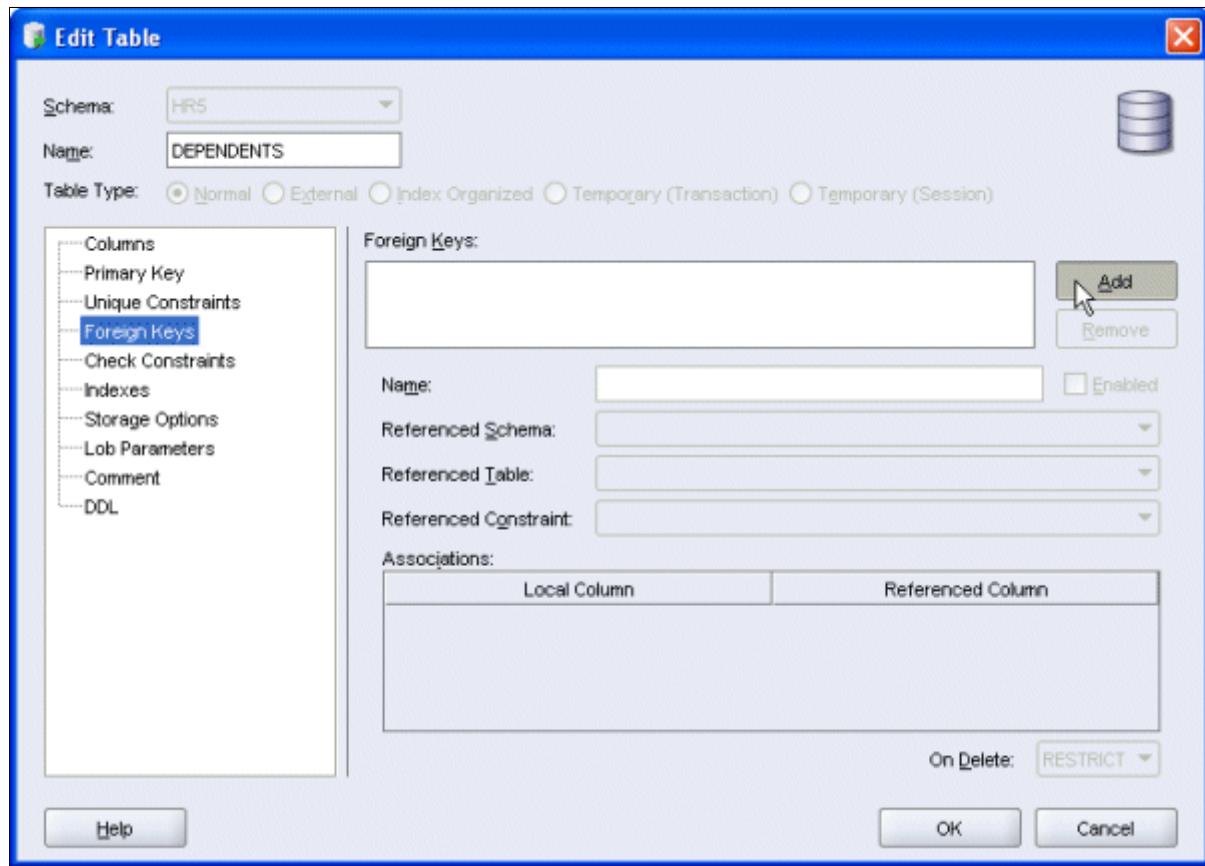
2. Click the **Primary Key** node in the tree.



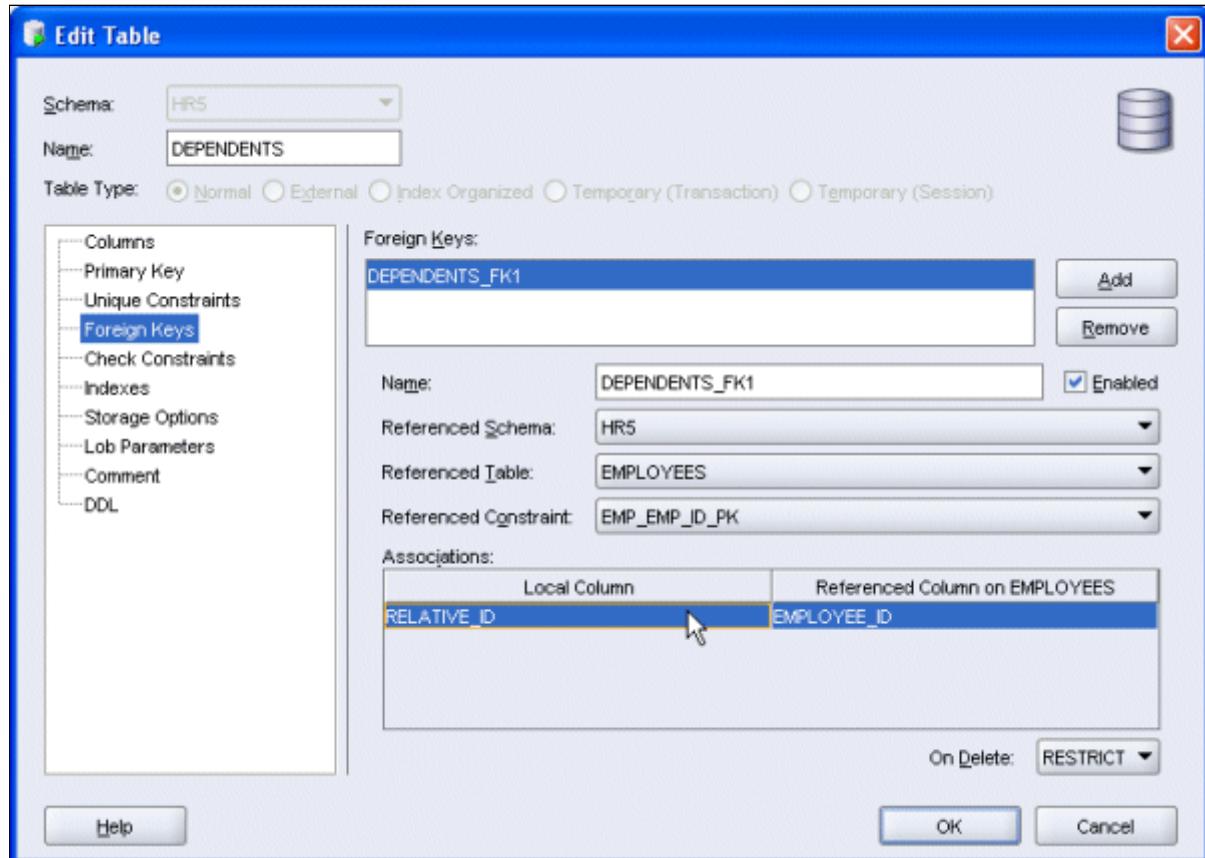
3. Select the **ID** column and click > to shuttle the value to the Selected Columns window. Then click the **Foreign Key** node in the tree.



4. Click Add.



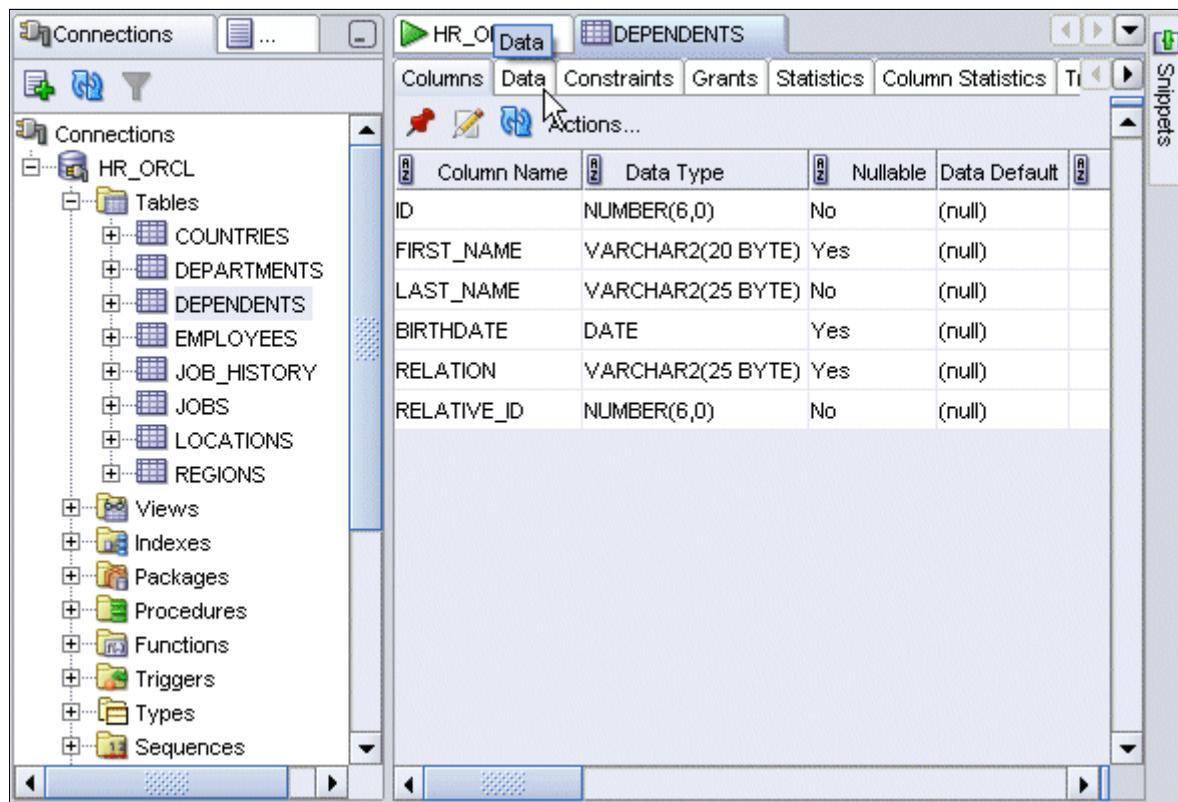
5. Select EMPLOYEES for the Referenced Table and select RELATIVE_ID for the Local Column and click OK.



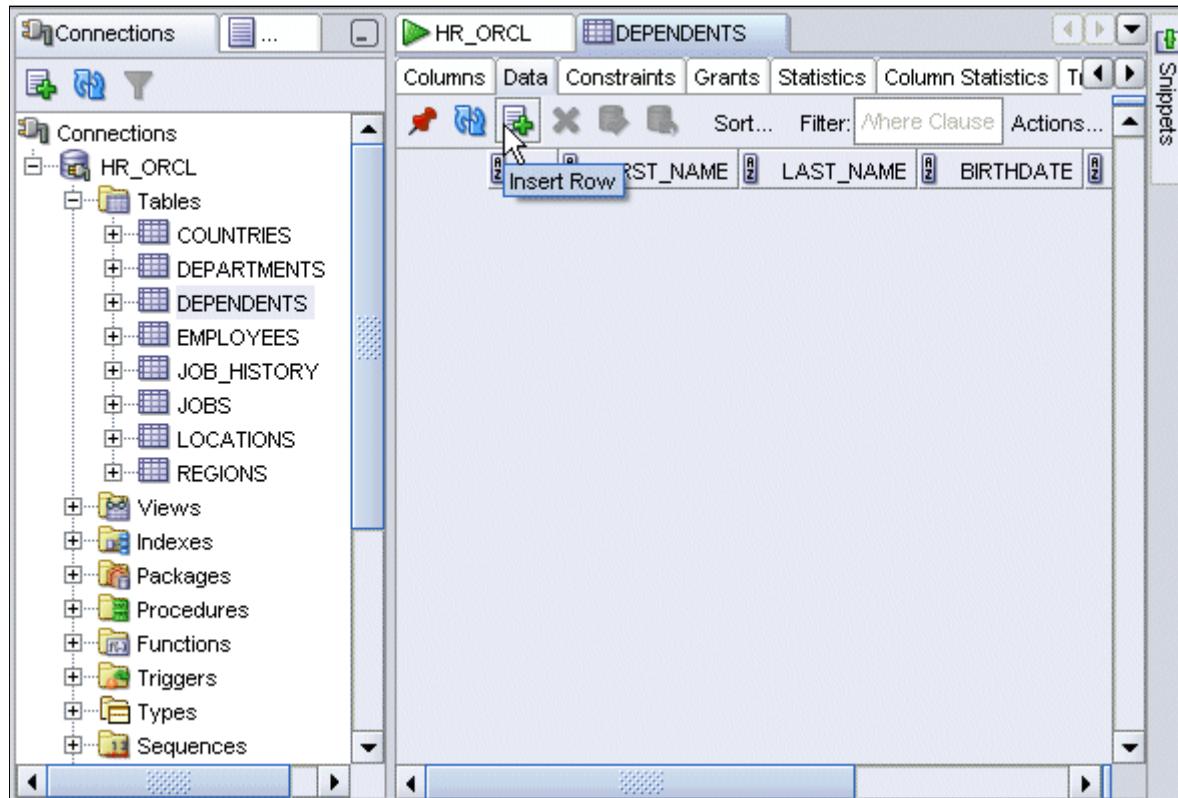
Adding Data to a Table

You can add data to a table by performing the following steps:

1. Click the **Data** tab.



2. Then click the **Insert Row** icon.



3. Enter the following data and click the **Commit Changes** icon.

ID	209
FIRST_NAME	Sue
LAST_NAME	Littlefield
BIRTHDATE	01-JAN-97
RELATION	Daughter
RELATIVE_ID	110

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL' containing various schema objects like COUNTRIES, DEPARTMENTS, and EMPLOYEES. The main window displays the 'DEPENDENTS' table with columns AST_NAME, BIRTHDATE, RELATION, and RELATIVE_ID. A new row is being added with the following data:

AST_NAME	BIRTHDATE	RELATION	RELATIVE_ID
ield	01-JAN-97	Daughter	110

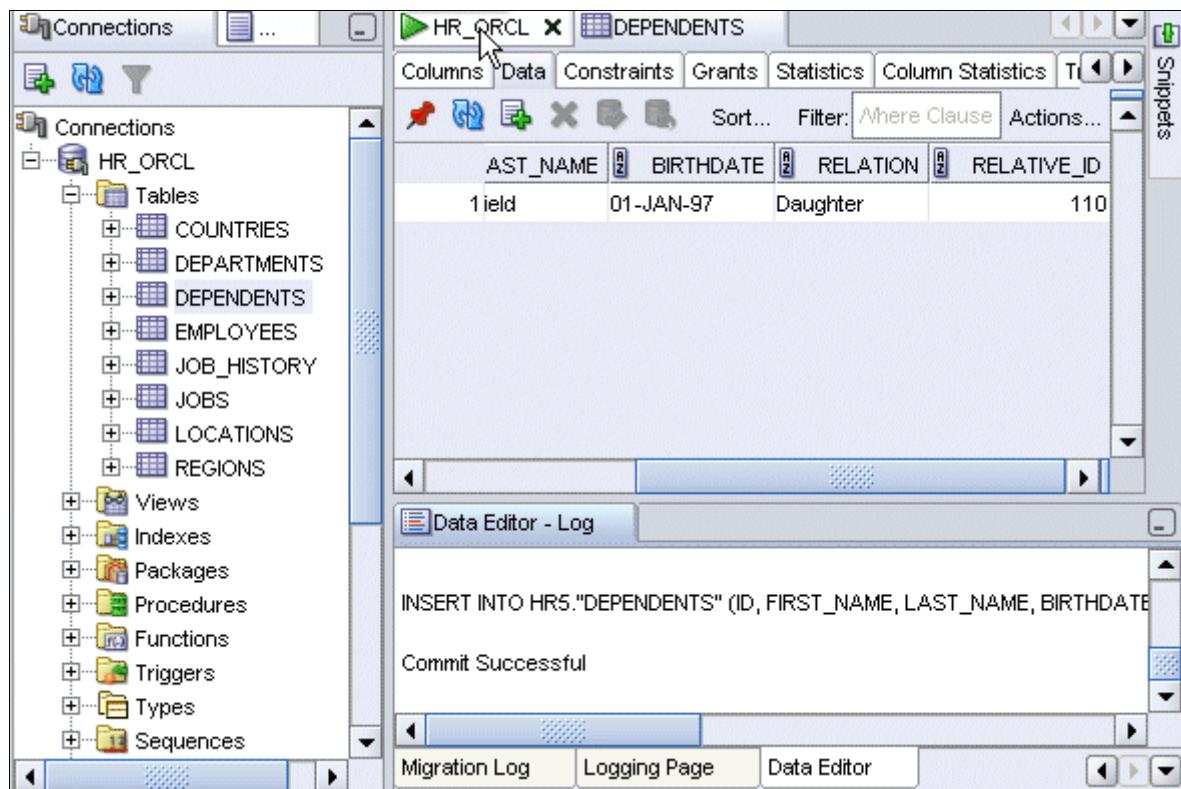
4. The row you just added was committed to the database.

The screenshot shows the Oracle SQL Developer interface with the 'Data Editor - Log' pane open. The log contains the following entries:

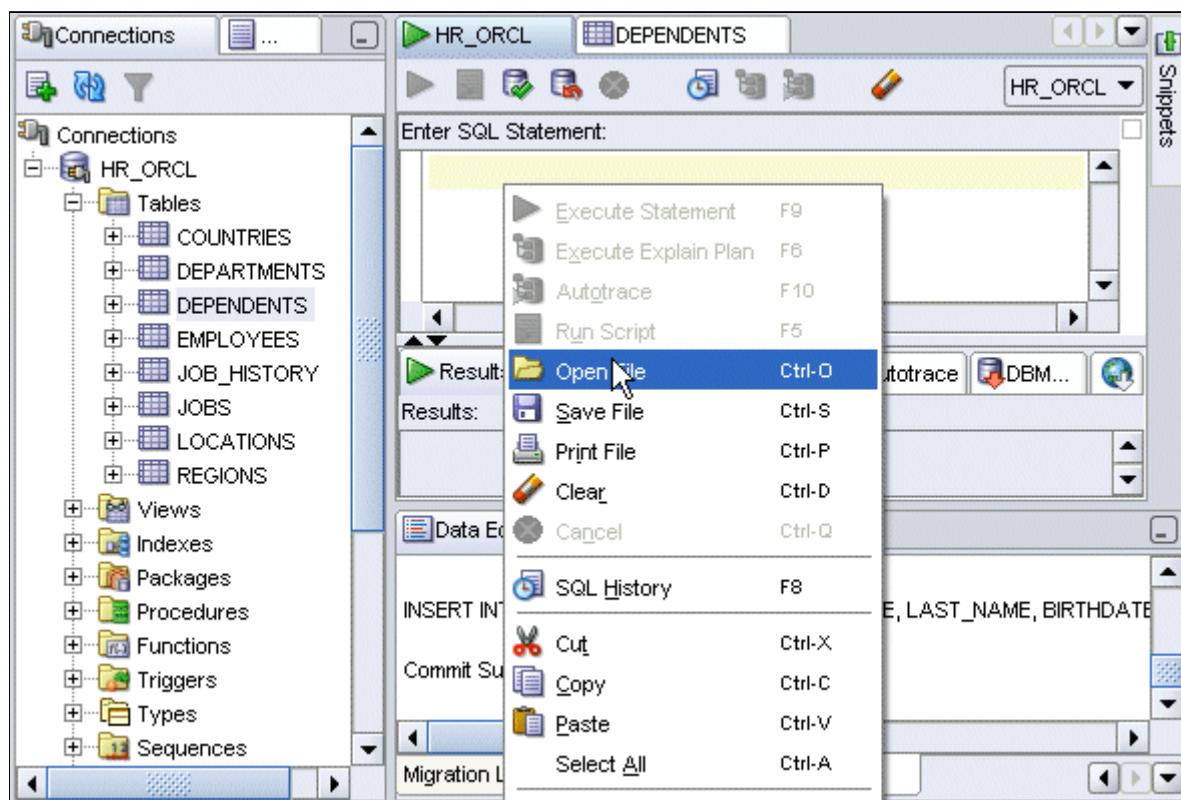
```

INSERT INTO HR5."DEPENDENTS" (ID, FIRST_NAME, LAST_NAME, BIRTHDATE)
Commit Successful
  
```

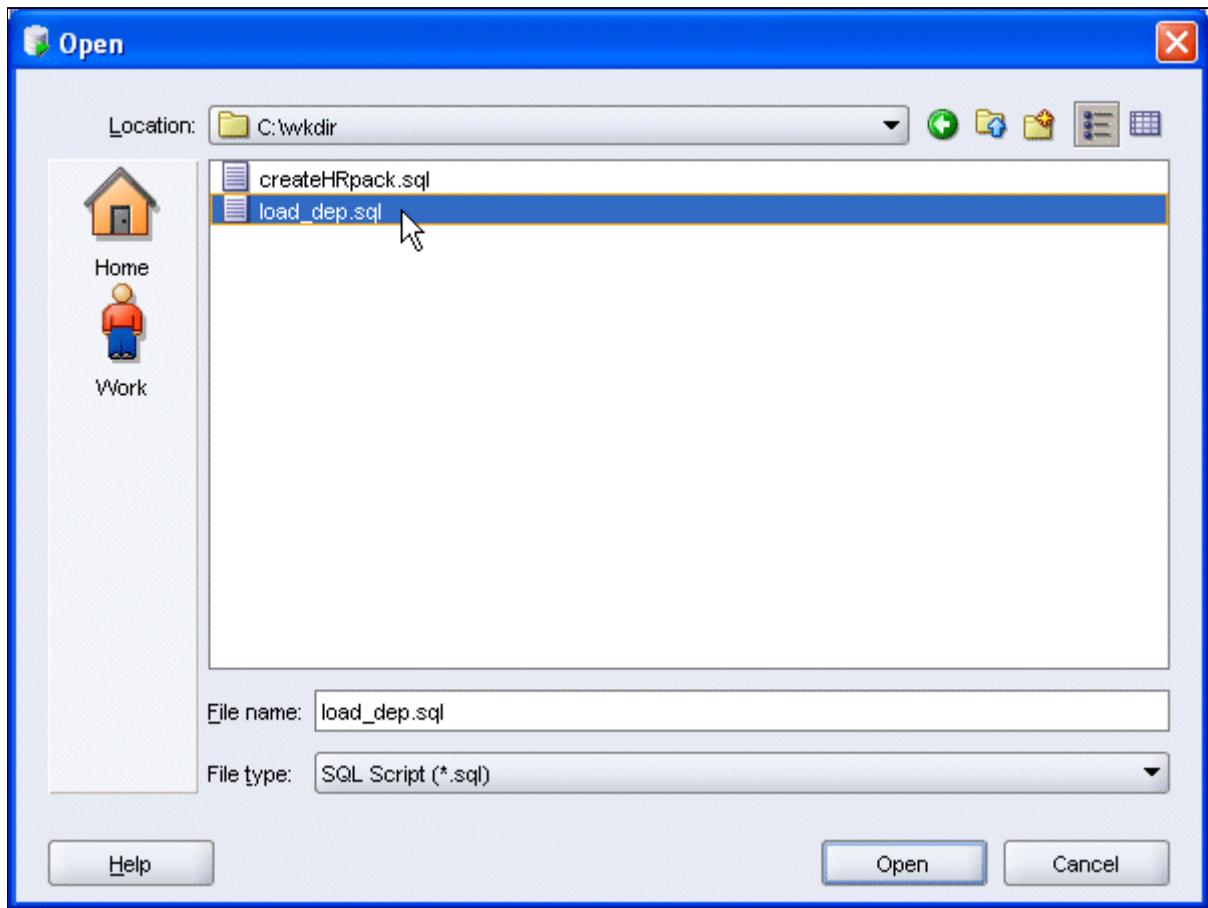
5. You can also load multiple rows at one time using a script. Click the SQL Worksheet HR_ORCL tab.



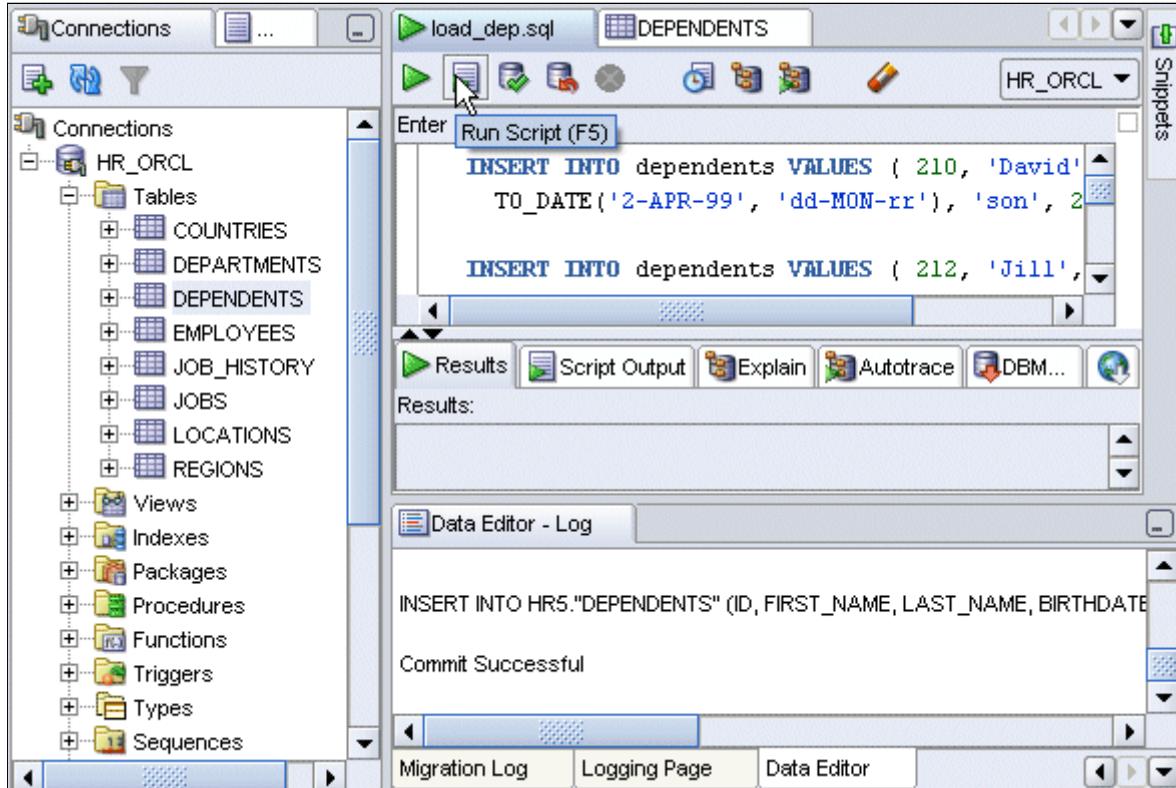
6. Right-click in the SQL Statement area and select Open File.



7. Navigate to the \SQLDev_HOS\GettingStarted\files directory and select the load_dep.sql file and click Open.



8. The SQL from the script is shown. Click the **Run Script** icon.



9. The data was inserted. To view the data, click the **DEPENDENTS** tab.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to HR_ORCL. The central workspace contains three SQL statements in a script named 'load_dep.sql':

```
INSERT INTO dependents VALUES ( 210, 'David', TO_DATE('2-APR-99', 'dd-MON-rr'), 'son', 210, 'Littlefield', TO_DATE('01-JAN-97', 'dd-MON-rr'), 'Daughter' );
INSERT INTO dependents VALUES ( 212, 'Jill', TO_DATE('10-FEB-92', 'dd-MON-rr'), 'daughter', 212, 'Littlefield', TO_DATE('19-AUG-01', 'dd-MON-rr'), 'daughter' );
INSERT INTO dependents VALUES ( 214, 'Vicki', TO_DATE('14-MAR-94', 'dd-MON-rr'), 'daughter', 214, 'Littlefield', TO_DATE('01-JAN-97', 'dd-MON-rr'), 'daughter' );
```

The results pane at the bottom shows four rows inserted for each statement.

10. Click Refresh to show all the data.

The screenshot shows the Oracle SQL Developer interface. The central workspace displays the DEPENDENTS table data after the refresh:

LAST_NAME	BIRTHDATE	RELATION
Littlefield	01-JAN-97	Daughter

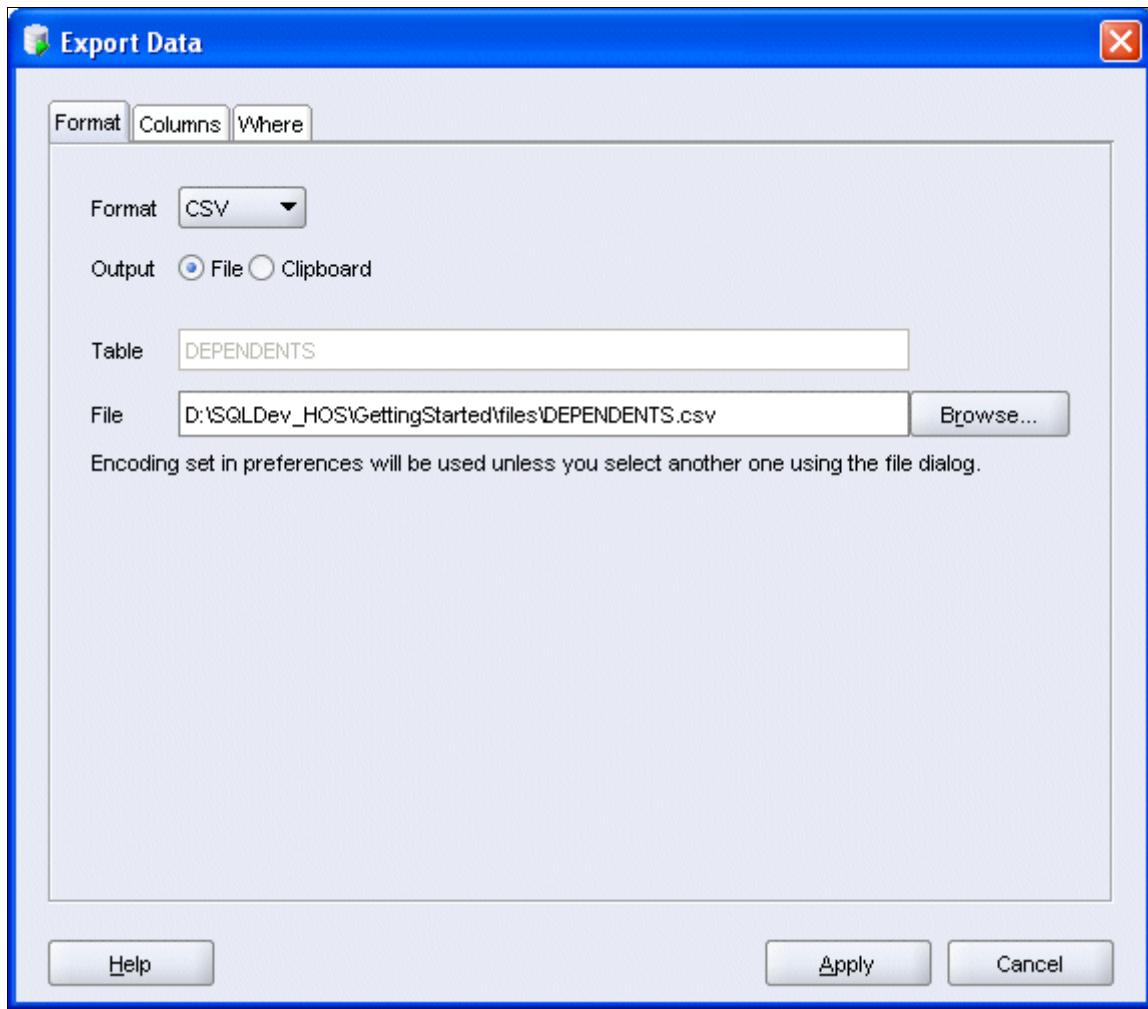
11. All the data is displayed

ID	FIRST_NAME	LAST_NAME	BIRTHDATE
1	Jill	Reed	10-FEB-92
2	Sue	Littlefield	01-JAN-97
3	David	Griffiths	02-APR-99
4	Vicki	Dean	19-AUG-01
5	Don	King	24-OCT-89

12. You can export the data so it can be used in another tool, for example, Excel. Right-click on one of the values in any column, select **Export** and then one of the file types.

ID	FIRST_NAME	LAST_NAME	BIRTHDATE
1	Jill	Reed	10-FEB-92
2	Sue	Littlefield	01-JAN-97
3	210	David	Griffiths
4		Single Record View	
5		Auto-fit All Columns	
		Auto-fit Selected Column	
		Duplicate Row	
		Count Rows	
		Export Data	
		TEXT	
		CSV	
		INSERT	
		LOADER	
		XML	
		HTML	
		XLS	

13. Specify the directory and name of the file and click **Apply**.



14. If you review the DEPENDENTS.CSV file, you see the following.

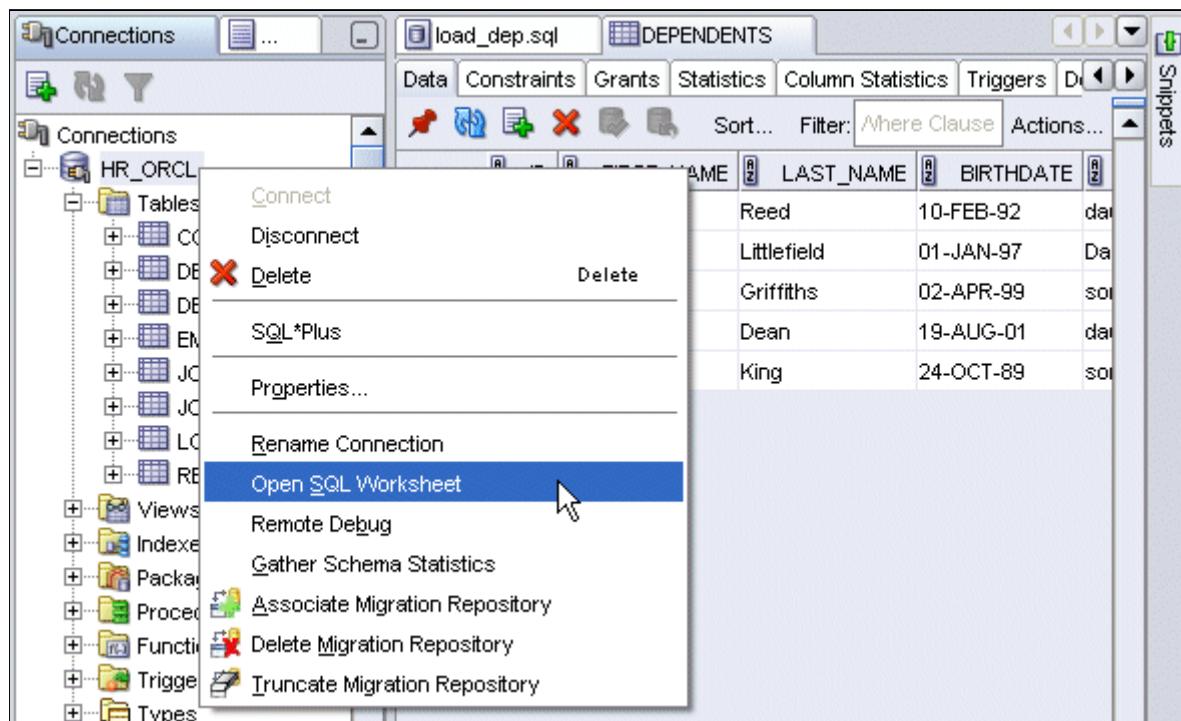
The screenshot shows a Microsoft Excel spreadsheet titled 'Microsoft Excel - dependents.csv'. The data is organized into columns labeled A through H. Column A is labeled 'ID', column B is 'FIRST_NA', column C is 'LAST_NAM', column D is 'BIRTHDAT', column E is 'RELATION', and column F is 'RELATIVE_ID'. The data consists of six rows of information:

	A	B	C	D	E	F	G	H
1	ID	FIRST_NA	LAST_NAM	BIRTHDAT	RELATION	RELATIVE_ID		
2	212	Jill	Reed	10-Feb-92	daughter	110		
3	207	Sue	Littlefield	01-Jan-97	Daughter	110		
4	210	David	Griffiths	02-Apr-99	son	201		
5	214	Vicki	Dean	19-Aug-01	daughter	201		
6	215	Don	King	24-Oct-89	son	120		
7								
8								
9								
10								
11								

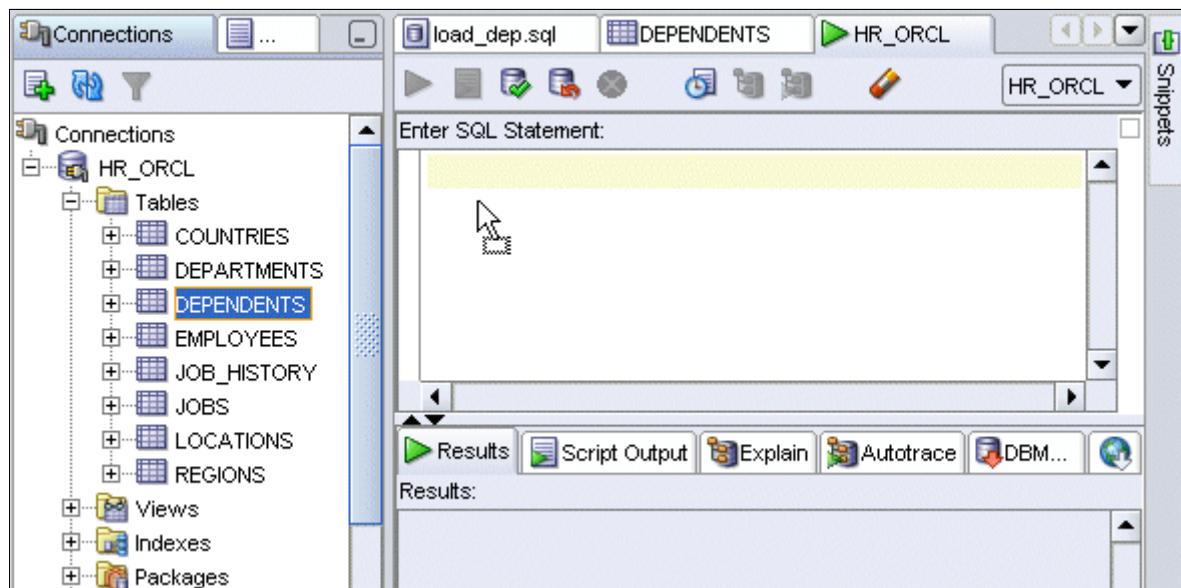
[Back to Topic List](#)

One way to access DEPENDENTS data is to generate a SELECT statement on the DEPENDENTS table and add a WHERE clause. Perform the following steps:

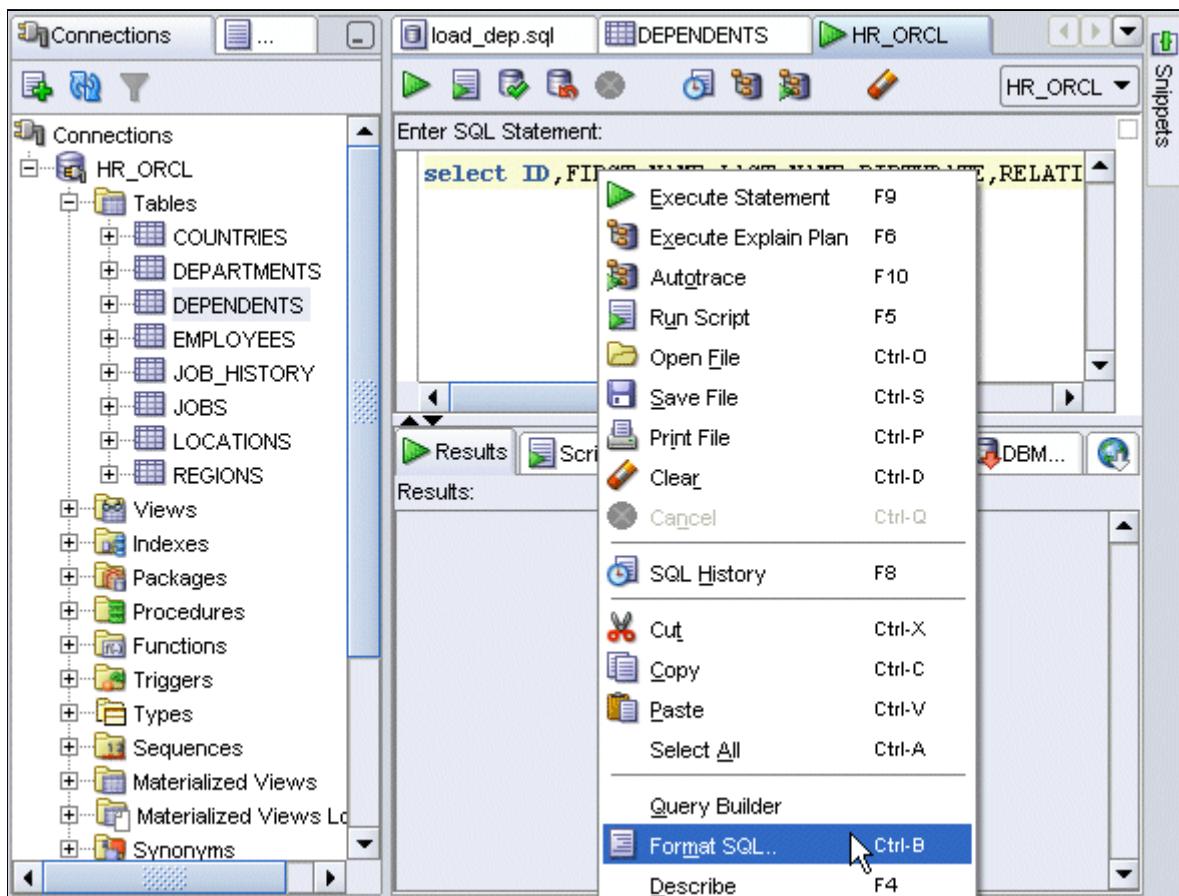
1. Select the **HR_ORCL** Database Connection, right-click and select Open SQL Worksheet.



2. Drag and Drop the **DEPENDENTS** table from the list of database objects to the SQL statement area.

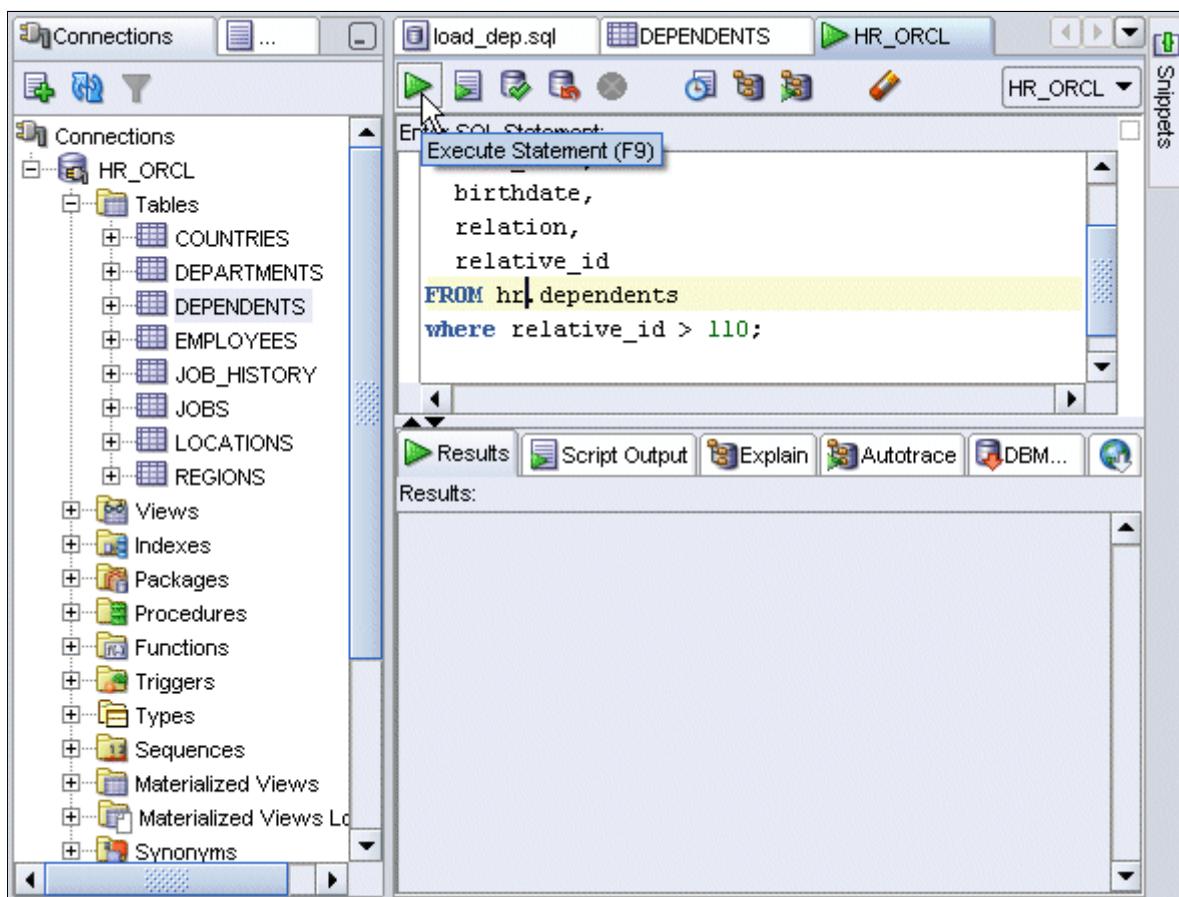


3. A SELECT statement is shown with all the columns contained in the DEPENDENTS table. Right-click and select **Format SQL...**



4. Add the WHERE clause *where relative_id > 110* to the end of the SELECT statement BEFORE the ';'.

Then click the **Execute Statement** icon.



5. The results are shown.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows the HR_ORCL connection selected. The central workspace displays a SQL worksheet titled 'load_dep.sql' with the following SQL statement:

```
SELECT id,
       first_name,
       last_name,
       birthdate,
       relation,
       relative_id
```

The 'Results' tab is active, showing the output of the query:

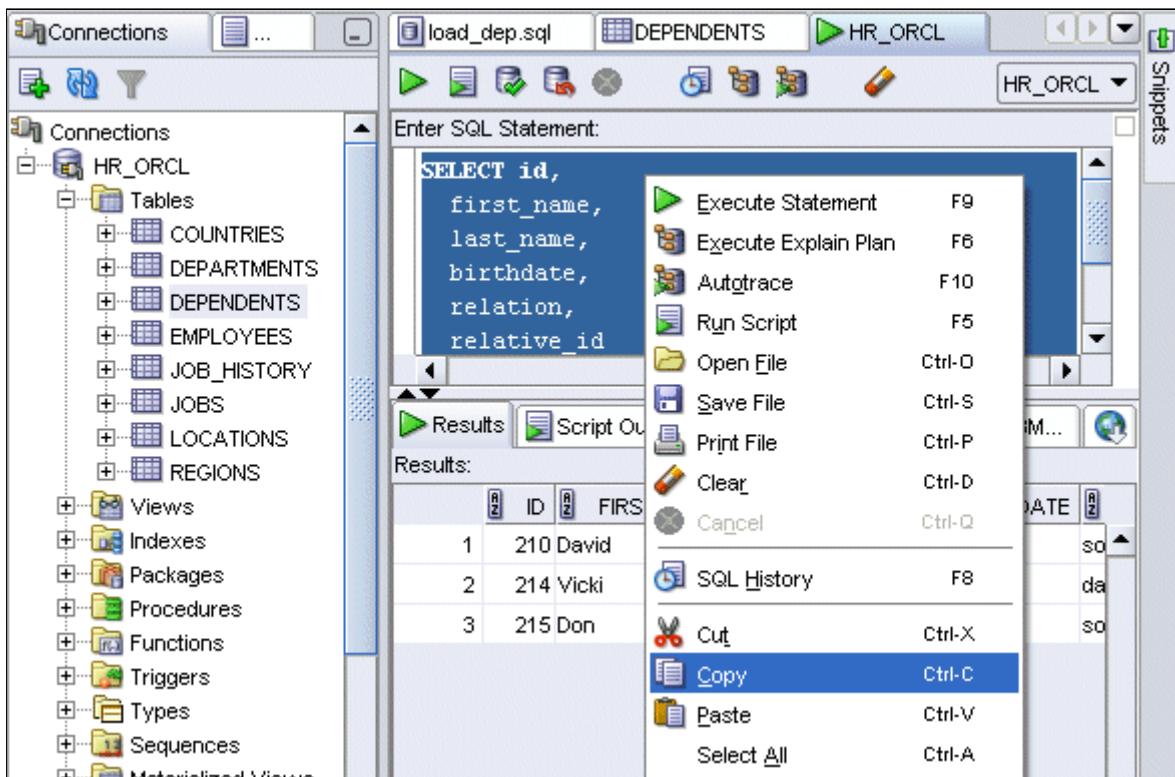
ID	FIRST_NAME	LAST_NAME	BIRTHDATE
1	David	Griffiths	02-APR-99
2	Vicki	Dean	19-AUG-01
3	Don	King	24-OCT-89

[Back to Topic List](#)

Creating Reports

Since the SQL you just ran in the previous topic needs to be executed frequently, you can create a custom report based on the SQL. In addition, you can run a report of your database data dictionary using bind variables. Perform the following steps:

1. Select the SQL in the HR_ORCL SQL Worksheet that you executed, right-click and select **Copy**.



2. Click the **Reports** tab.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL' containing tables like COUNTRIES, DEPARTMENTS, DEPENDENTS, EMPLOYEES, etc. The central workspace displays a query editor with the following SQL statement:

```
SELECT id,
       first_name,
       last_name,
       birthdate,
       relation,
       relative_id
```

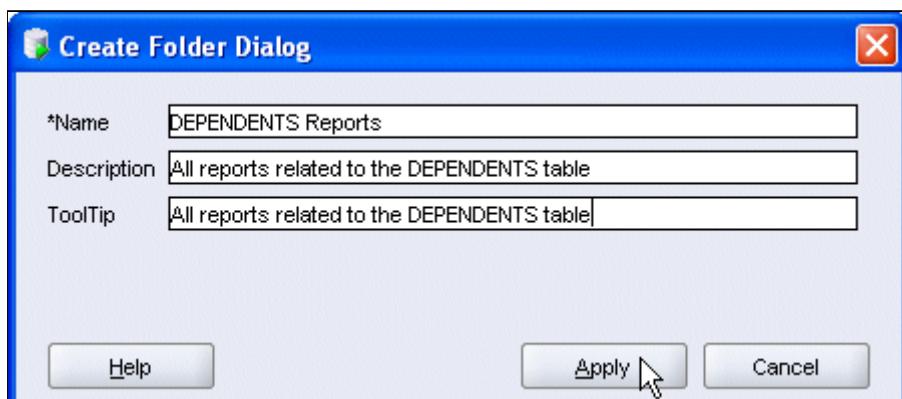
Below the query editor is a results grid showing three rows of data from the DEPENDENTS table:

ID	FIRST_NAME	LAST_NAME	BIRTHDATE
1	David	Griffiths	02-APR-81
2	Vicki	Dean	19-AUG-83
3	Don	King	24-OCT-53

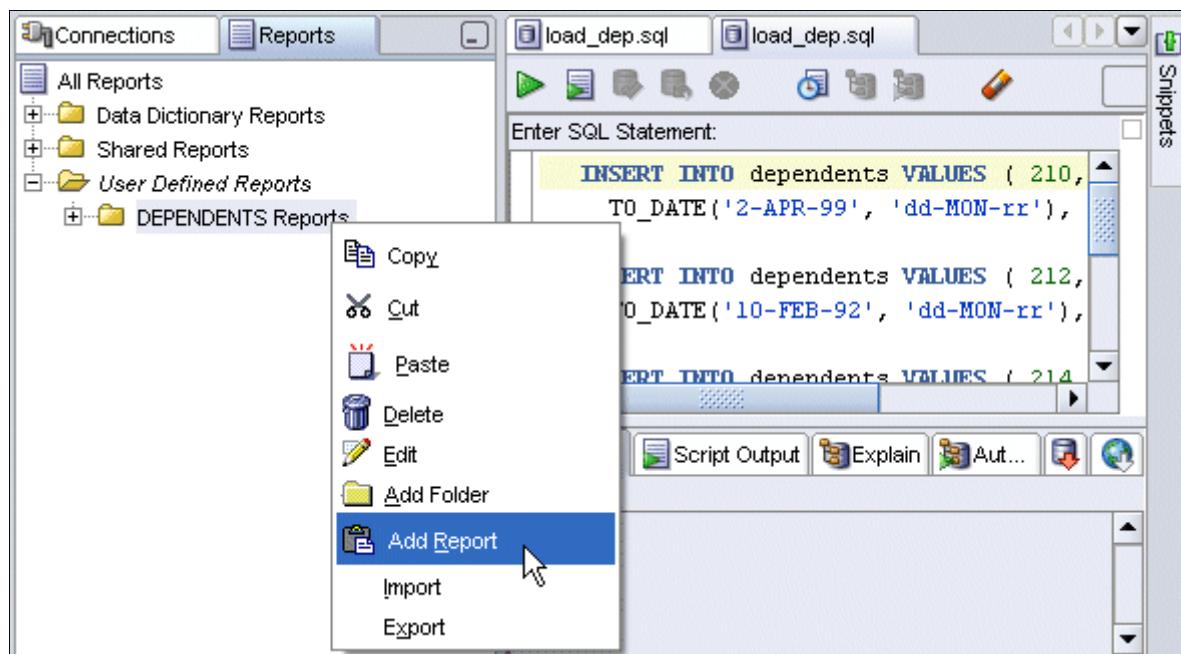
3. Right-click **User Defined Reports** and select **Add Folder**.

The screenshot shows the Oracle SQL Developer interface. The 'Reports' tab is selected in the top bar. In the center, there is a context menu open over the 'User Defined Reports' folder. The 'Add Folder' option is highlighted with a blue selection bar.

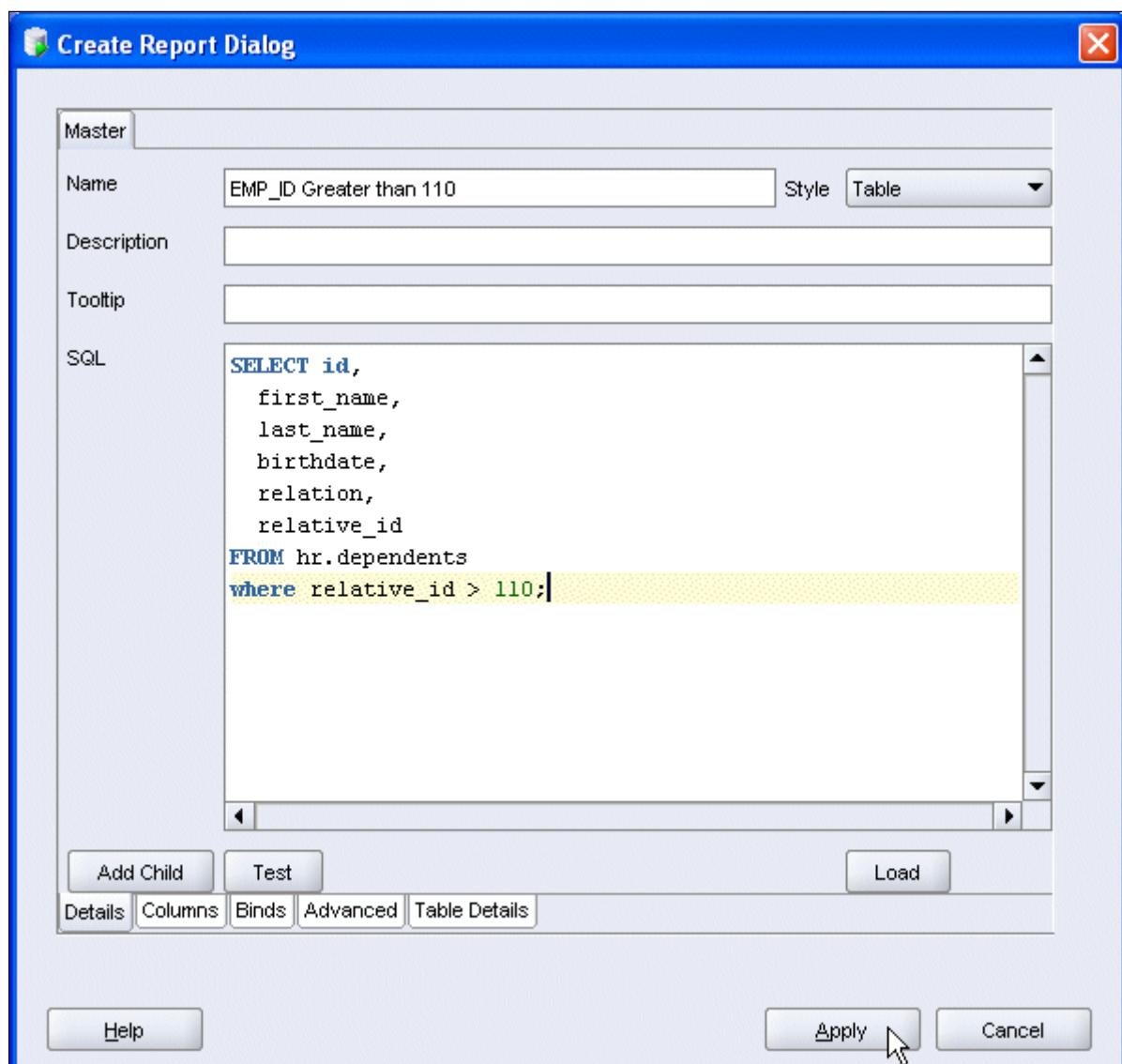
4. Enter the Folder Name **DEPENDENTS Reports**. You can add a Description and Tooltip of your choice. Click **Apply**.



5. Right-click **DEPENDENT Reports** and select **Add Report**.



6. Enter a Name for the report and click **ctrl+v** in the SQL area to paste the SQL you copied from the HR_ORCL SQL Worksheet. Then click **Apply**.



7. Select the Report you just created.

The screenshot shows the Oracle SQL Developer interface. The left pane displays a tree view of reports under 'User Defined Reports'. A report titled 'EMP_ID Greater than 110' is selected. The main pane shows the SQL query:

```
SELECT id,
       first_name,
       last_name,
       birthdate,
       relative_id
  FROM hr5.dependents
 WHERE relative_id > 110;
```

The results pane below shows the output:

ID	FIRST_NAME
210	David
214	Vicki

8. Select HR_ORCL from the drop list and click **OK** to connect to your database.

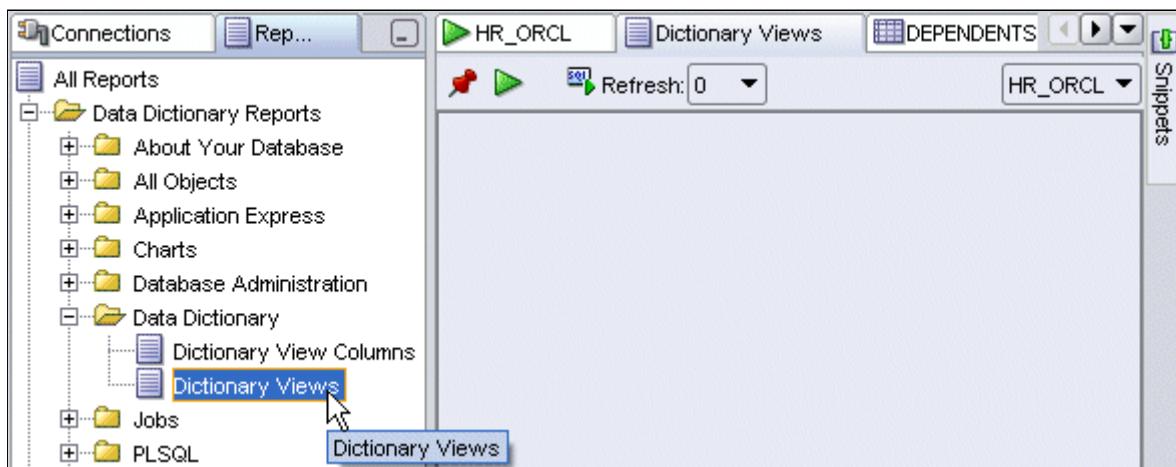


9. The results of your report are shown.

The screenshot shows the Oracle SQL Developer interface with the report results grid. The columns are labeled: ID, FIRST NAME, LAST NAME, BIRTHDATE, and RELAT. The data is as follows:

ID	FIRST NAME	LAST NAME	BIRTHDATE	RELAT
210	David	Griffiths	02-APR-99	son
214	Vicki	Dean	19-AUG-01	daughter
215	Don	King	24-OCT-89	son

10. You can also run a Data Dictionary report. Expand **Data Dictionary Reports > Data Dictionary**. Then select **Dictionary Views**.



11. Enter **col** for the Value and click **Apply**.



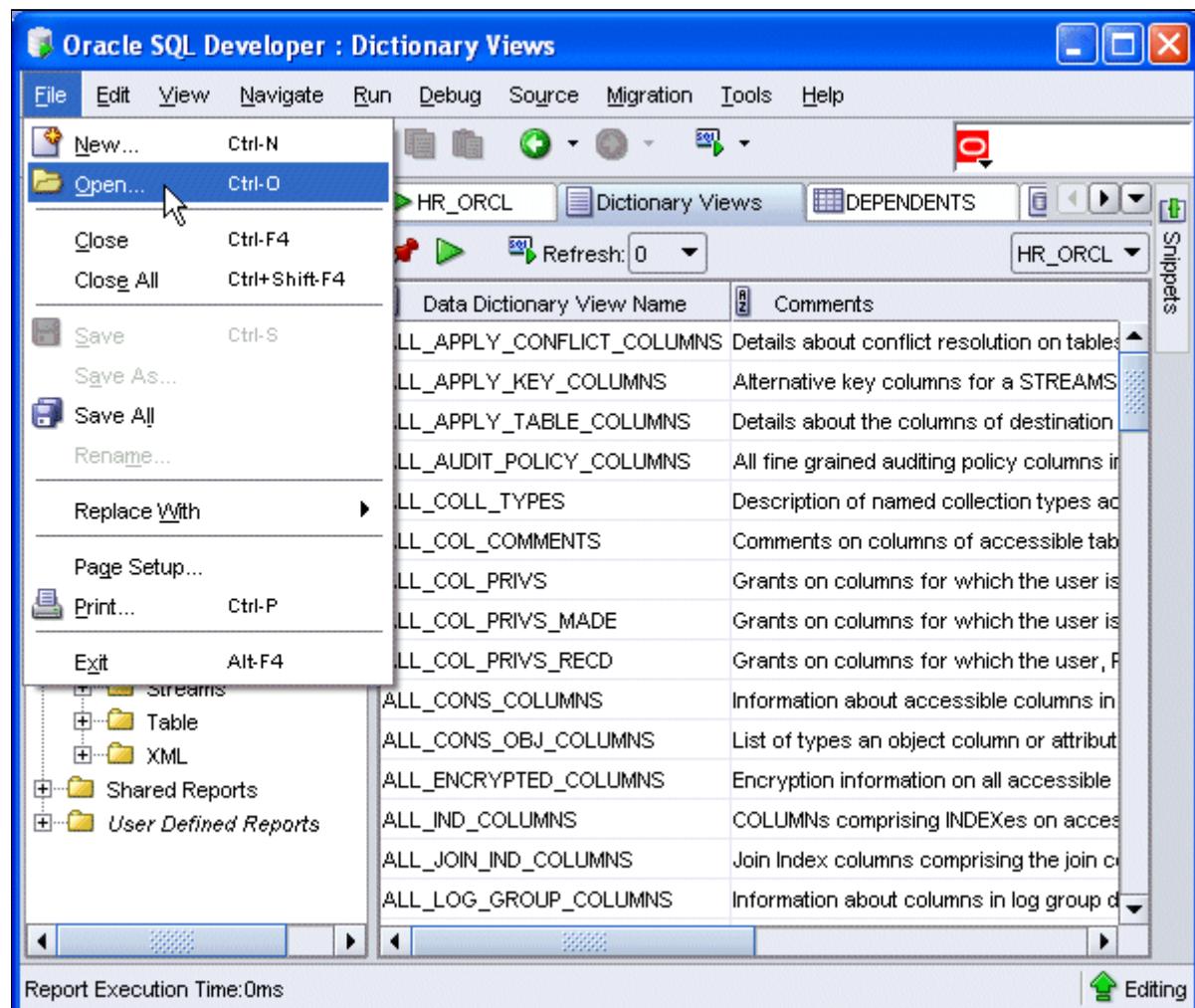
12. All the Data Dictionary views that contain 'col' in its name are displayed.

View Name	Comments
ALL_APPLY_CONFLICT_COLUMNS	Details about conflict resolution on tables
ALL_APPLY_KEY_COLUMNS	Alternative key columns for a STREAMS
ALL_APPLY_TABLE_COLUMNS	Details about the columns of destination
ALL_AUDIT_POLICY_COLUMNS	All fine grained auditing policy columns in
ALL_COLL_TYPES	Description of named collection types ac
ALL_COL_COMMENTS	Comments on columns of accessible tab
ALL_COL_PRIVS	Grants on columns for which the user is
ALL_COL_PRIVS_MADE	Grants on columns for which the user is

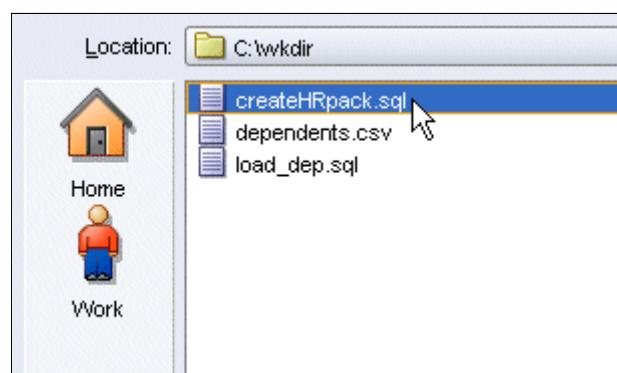
[Back to Topic List](#)

Oracle SQL Developer contains extensive PL/SQL debugging capabilities. In this topic, you create a Package Spec and Package Body that adjusts an employee's salary. Perform the following steps:

1. Select **File > Open** using the main menu.



2. Browse to the **\SQLDev_HOS\GettingStarted\files** directory and select **createHRpack.sql**. Click **Open**.



3. Select the HR_ORCL database connection from the the drop list on the right.

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for 'HR_ORCL', 'Dictionary Views', and 'createHRpack.sql'. On the far right of the top bar, there is a dropdown menu labeled 'Snippets' with a small green icon. A mouse cursor is hovering over this dropdown. The main workspace contains a tree view on the left under 'All Reports' and a SQL editor on the right. The SQL editor has a toolbar at the top with icons for running scripts, saving, and so on. Below the toolbar is a text input field labeled 'Enter SQL Statement:' containing the following code:

```
DROP PROCEDURE raise_amount;
DROP FUNCTION get_sal;

CREATE OR REPLACE PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (in_id IN employees.employee_id%TYPE)
```

4. Click the **Run Script** icon.

This screenshot is identical to the one above it, showing the Oracle SQL Developer interface with the 'Run Script' icon highlighted in the toolbar. The SQL editor window contains the same code as before:

```
DROP PROCEDURE raise_amount;
DROP FUNCTION get_sal;

CREATE OR REPLACE PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (in_id IN employees.employee_id%TYPE)
```

5. Click the **Connections** tab.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows HR_ORCL selected. The central workspace has a toolbar with a play button, a magnifying glass, and a pencil. Below the toolbar is a text input field labeled "Enter SQL Statement:". The SQL code entered is:

```
DROP PROCEDURE raise_amount;
DROP FUNCTION get_sal;

CREATE OR REPLACE PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (in_id IN employees.employee_id%TYPE,
        p_raise_amt IN NUMBER default 0);
    FUNCTION get_sal
        (p_id IN employees.employee_id%TYPE,
        p_increment IN NUMBER := 1)
        RETURN NUMBER;
END hr_pack;
```

6. Expand **HR_ORCL > Packages > HR_PACK** and select **HR_PACK** to view the package definition.

The screenshot shows the Oracle SQL Developer interface with the connections tree on the left. Under HR_ORCL, the Packages node is expanded, and the HR_PACK package is selected. The central workspace shows the package definition:

```
create or replace PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (p_id IN employees.employee_id%TYPE,
        p_raise_amt IN NUMBER default 0);
    FUNCTION get_sal
        (p_id IN employees.employee_id%TYPE,
        p_increment IN NUMBER := 1)
        RETURN NUMBER;
END hr_pack;
```

7. Select **HR_PACK BODY** to view the package body definition.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'HR_ORCL' containing tables like COUNTRIES, DEPARTMENTS, and EMPLOYEES, and packages like HR_PACK. The HR_PACK package is expanded, showing its body. On the right, the main window displays the PL/SQL code for the package body:

```
create or replace PACKAGE BODY hr_pack
IS
  PROCEDURE raise_amount
    (p_id IN employees.employee_id%TYPE,
     p_raise_amt IN NUMBER default 0)
  IS
  BEGIN
    UPDATE employees
      SET salary = salary * (1+p_raise_amt)
      WHERE employee_id = p_id;
  END raise_amount;

  FUNCTION get_sal
    (p_id IN employees.employee_id%TYPE,
     p_increment IN NUMBER := 1 )
  RETURN NUMBER
  IS
```

8. To make any changes to the Package Body, click the **Edit** icon.

The screenshot shows the Oracle SQL Developer interface. The 'Actions...' toolbar at the top has the 'Edit' icon highlighted with a cursor. The code editor on the right contains the same PL/SQL code as the previous screenshot. The 'Edit' icon is located in the toolbar area.

9. Click on any one of the - to collapse the code or press + to expand the code.

The screenshot shows the Oracle SQL Developer interface. The left pane is the Connections Navigator, displaying a connection to 'HR_ORCL' which contains tables like COUNTRIES, DEPARTMENTS, and EMPLOYEES, and packages like HR_PACK. The right pane is a code editor with the following PL/SQL code:

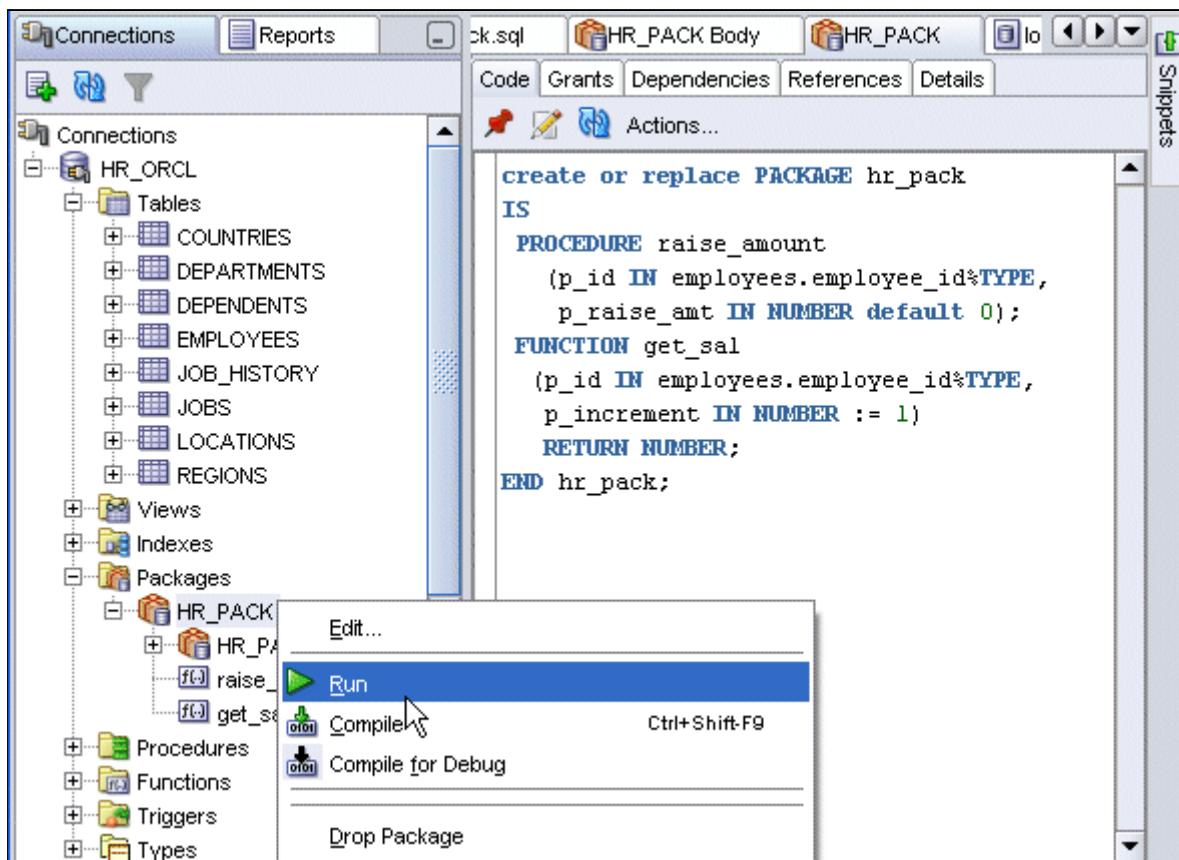
```
create or replace
PACKAGE BODY hr_pack
IS
  PROCEDURE raise_amount
    (p_id IN employees.employee_id%TYPE,
     p_raise_amt IN NUMBER default 0)
  IS
  BEGIN
    UPDATE employees
      SET salary = salary * (1+p_raise_
      WHERE employee_id = p_id;
  END raise_amount;

  FUNCTION get_sal
    (p_id IN employees.employee_id%TYPE,
     p_increment IN NUMBER := 1 )
  RETURN NUMBER
  IS
```

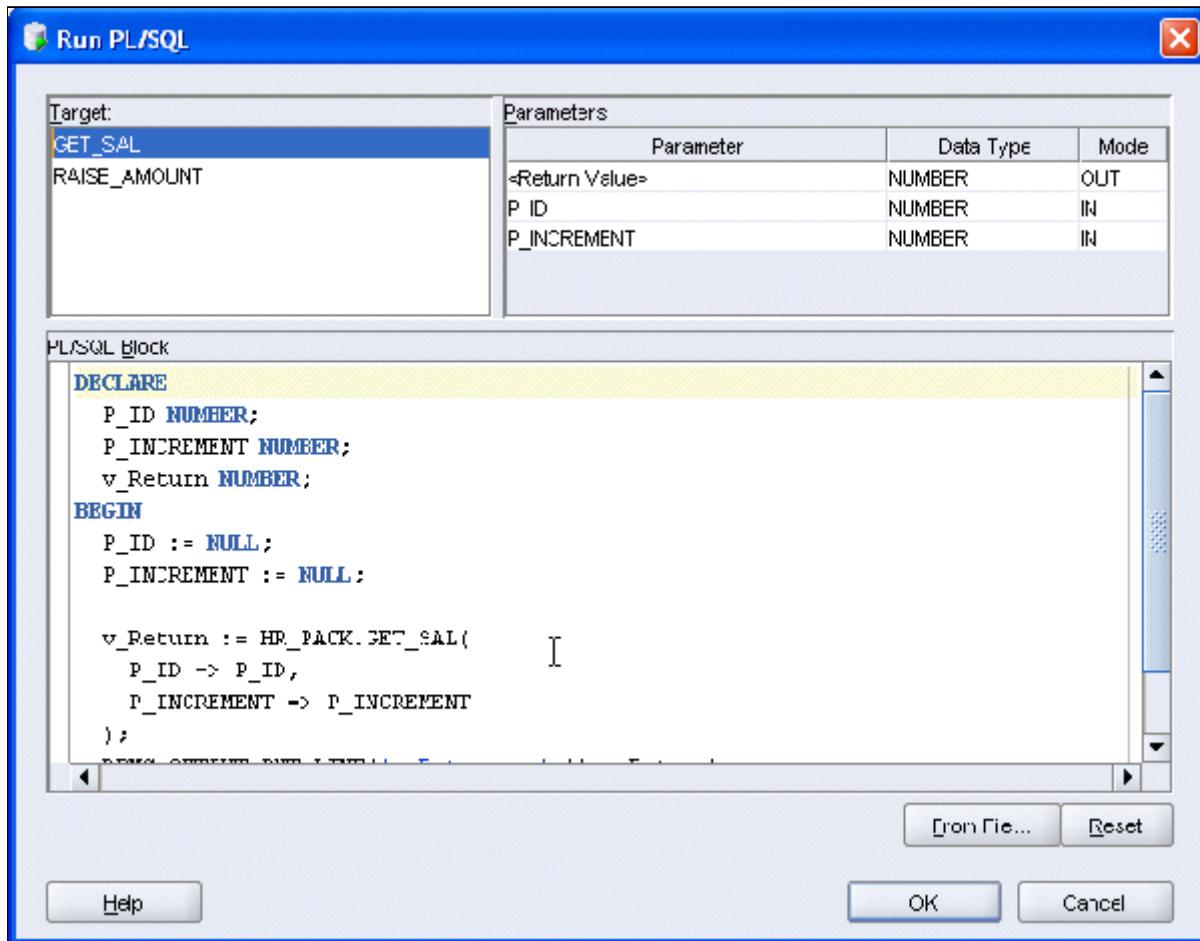
10. If your line numbers do not appear, you can right-click in the line number area and click **Toggle Line Numbers** to turn them on. This is useful for debugging purposes.

The screenshot shows the same Oracle SQL Developer interface as above, but with a context menu open over the code editor. The menu items include 'Toggle Bookmark' (Ctrl-K), 'Toggle Breakpoint' (F5), 'Toggle Line Numbers' (which is highlighted in blue), and 'Preferences...'. The code in the editor remains the same as in the previous screenshot.

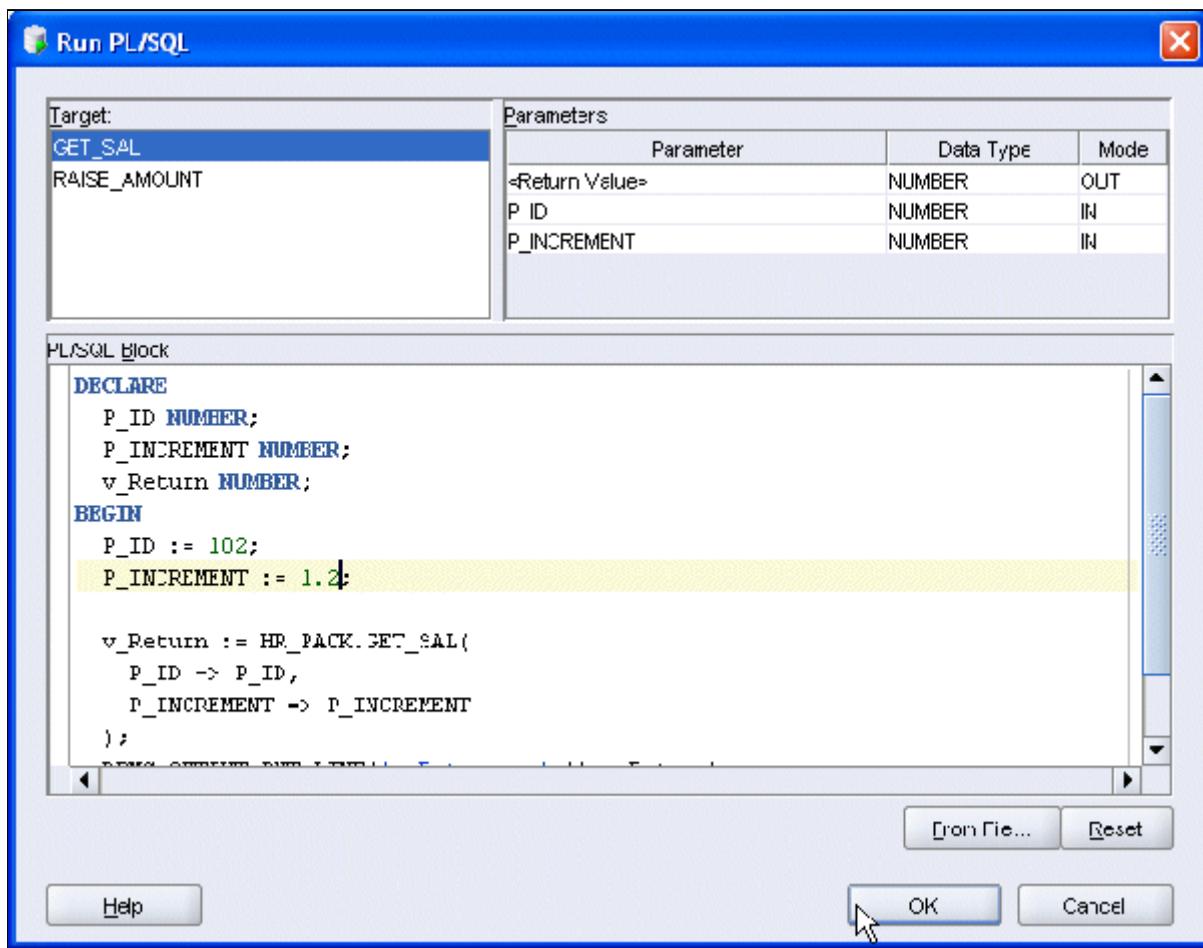
11. In the Connections Navigator, select Packages > **HR_PACK**, right-click and select **Run**.



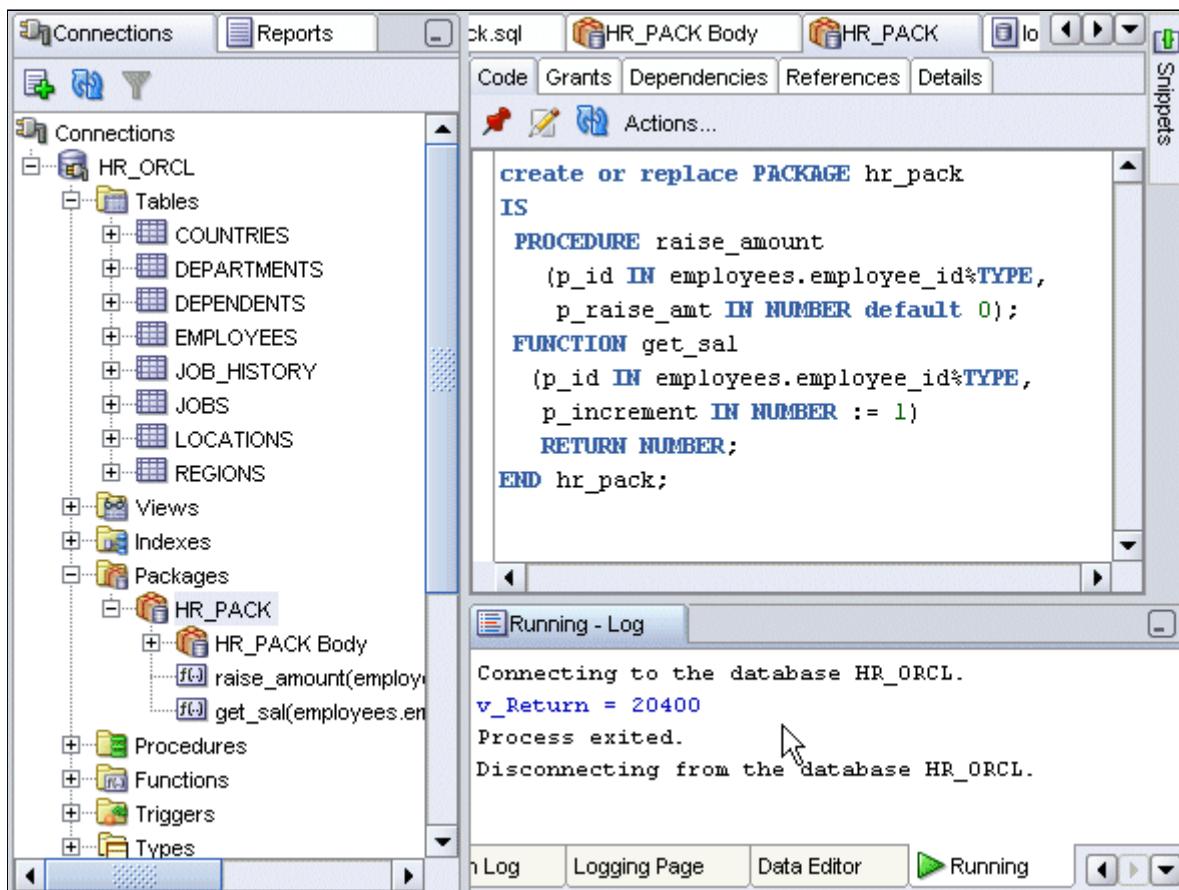
12. A parameter window appears. You need to set the input parameters here.



Set the P_ID to **102** and P_INCREMENT to **1.2**. What this means is that the Employee who has the ID 102, their salary is increased by 20%. The current SALARY for EMPLOYEE_ID 102 is 17000. Click **OK**.



13. The value returned is 20400.



14. To test the Exception Handling, right-click on HR_PACK in the navigator and select Run.

```

create or replace PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (p_id IN employees.employee_id%TYPE,
         p_raise_amt IN NUMBER default 0);
    FUNCTION get_sal
        (p_id IN employees.employee_id%TYPE,
         p_increment IN NUMBER := 1)
        RETURN NUMBER;
END hr_pack;

```

15. This time, change the P_INCREMENT value to 5 and click OK.

Parameter	Data Type	Mode
<Return Value>	NUMBER	OUT
P_ID	NUMBER	IN
P_INCREMENT	NUMBER	IN

```

DECLARE
    P_ID NUMBER;
    P_INCREMENT NUMBER;
    v_Return NUMBER;
BEGIN
    P_ID := 102;
    P_INCREMENT := 5; [highlighted]
    v_Return := HR_PACK.GET_SAL(
        P_ID => P_ID,
        P_INCREMENT => P_INCREMENT
    );

```

16. In this case, an exception was raised with "Invalid increment value" because the P_INCREMENT value was greater than 1.5.

```
create or replace PACKAGE hr_pack
IS
    PROCEDURE raise_amount
        (p_id IN employees.employee_id%TYPE,
         p_raise_amt IN NUMBER default 0);
    FUNCTION get_sal
        (p_id IN employees.employee_id%TYPE,
         p_increment IN NUMBER := 1)
        RETURN NUMBER;
END hr_pack;
```

Running - Log

Connecting to the database HR_ORCL.
Invalid increment amount
v_Return =
Process exited.
Disconnecting from the database HR_ORCL.

[Back to Topic List](#)

Summary

In this tutorial, you learned how to:

- Create a database connection
- Add a new table using the Table Dialog Box
- Change a table definition
- Add data to a table
- Access data
- Generate a report
- Debug and execute PL/SQL

[Back to Topic List](#)

 Move your mouse over this icon to hide all screenshots.

Developing and Debugging PL/SQL using Oracle SQL Developer

Purpose

This tutorial shows you how to create, run, and debug a PL/SQL procedure using Oracle SQL Developer.

Time to Complete

Approximately 30 minutes

Topics

This tutorial covers the following topics:

- [Overview](#)
- [Prerequisites](#)
- [Creating a Database Connection](#)
- [Browsing Your Database](#)
- [Creating and Compiling a PL/SQL Procedure](#)
- [Running a PL/SQL Procedure](#)
- [Debugging a PL/SQL Procedure](#)
- [Summary](#)

Viewing Screenshots

 Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)

Note: Alternatively, you can place the cursor over each individual icon in the following steps to load and view only the screenshot associated with that step.

Overview

Oracle SQL Developer is a free graphical tool that enhances productivity and simplifies database development tasks. With Oracle SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, as well as create and save your own. This tutorial focuses on creating, compiling, running and debugging PL/SQL.

[Back to Topic List](#)

Prerequisites

For this Hands On Session, the following has already been installed for you:

1. Oracle Database 10g.

Note: To repeat this exercise later you can use any Oracle Database above 9.2.0.1

2. Oracle SQL Developer 1.2.1.

Note: Oracle SQL Developer is available for download for FREE from [OTN](#). To install unzip it into any directory on your machine.

3. The files you use throughout the tutorial are located in the **\SQLDev_HOS\PLSQL\files** directory. The files have also been included in the [plsql.zip](#) file.

4. The shipped HR schema.

Note: For the PL/SQL debugging portion of this tutorial, the HR user needs a few additional privileges. These have

been granted using the following commands:

GRANT debug any procedure, debug connect session TO hr;

[Back to Topic List](#)

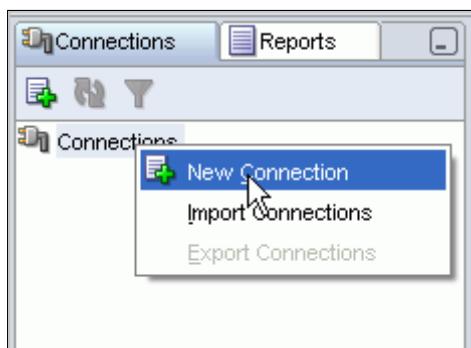
Creating a Database Connection

To create a database connection, perform the following steps:

1. Open Windows Explorer and double-click <your_path>\sqldeveloper\sqldeveloper.exe.

Note: If you receive a dialog window asking whether you want to migrate settings from a previous release, click **No**.

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

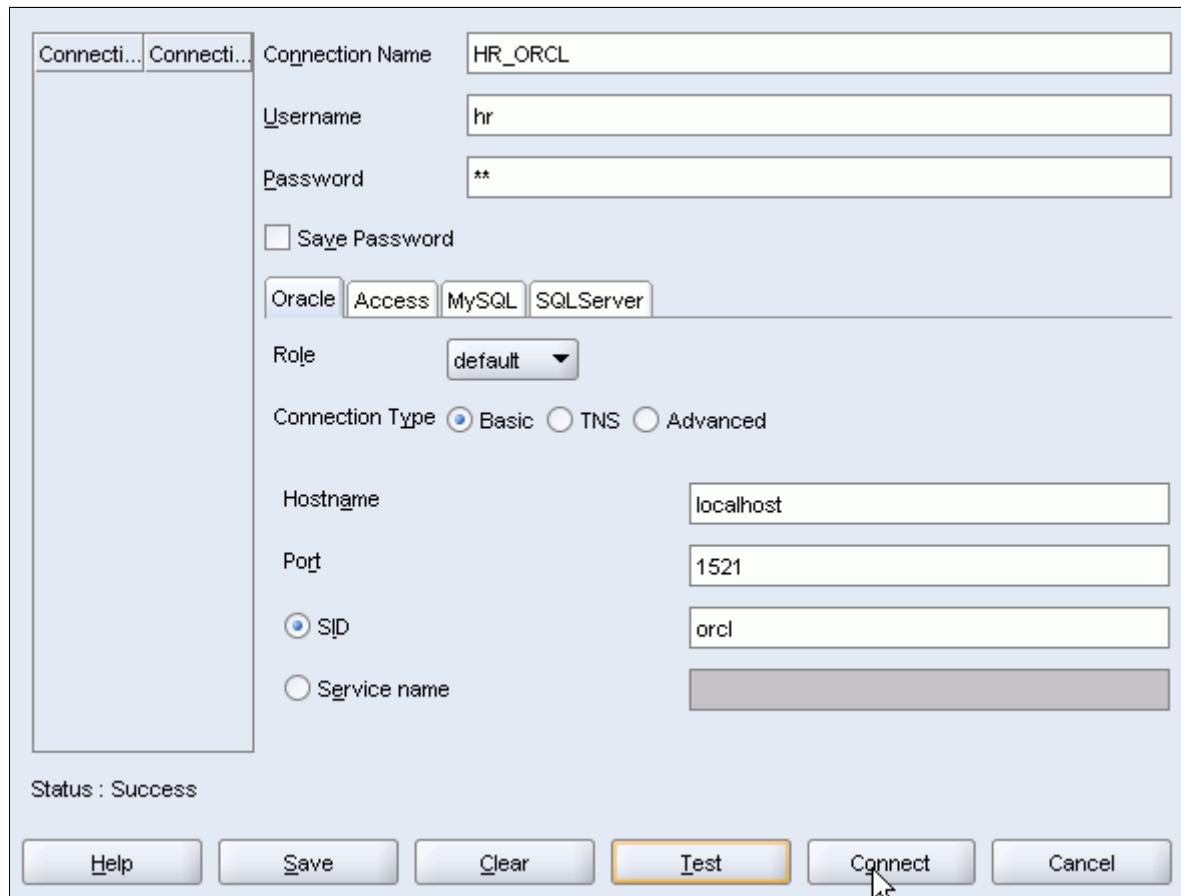


3. Enter **HR_ORCL** for the Connection Name (or any other name that identifies your connection), **HR** for the Username and Password, specify your <hostname> for the Hostname and enter **ORCL** for the SID. Click **Test**.

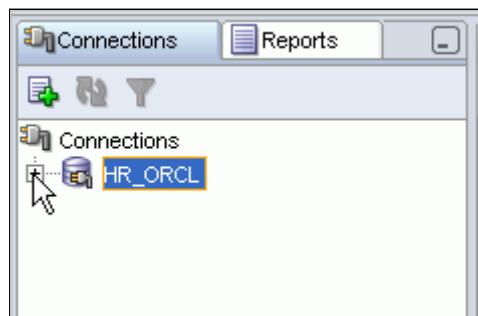
The screenshot shows the 'New Connection' dialog box. The 'Connection Name' field is set to 'HR_ORCL'. The 'Username' field contains 'hr' and the 'Password' field contains '**'. The 'Save Password' checkbox is unchecked. The 'Role' dropdown is set to 'default'. Under 'Connection Type', the 'Basic' radio button is selected. The 'Hostname' field is 'localhost', 'Port' is '1521', and 'SID' is 'orcl'. The 'Service name' radio button is unselected. At the bottom, there are buttons for 'Help', 'Save', 'Clear', 'Test' (with a cursor over it), 'Connect', and 'Cancel'.

Connection Name	HR_ORCL
Username	hr
Password	**
<input type="checkbox"/> Save Password	
Oracle Access MySQL SQLServer	
Role	default
Connection Type	<input checked="" type="radio"/> Basic <input type="radio"/> TNS <input type="radio"/> Advanced
Hostname	localhost
Port	1521
SID	orcl
Service name	

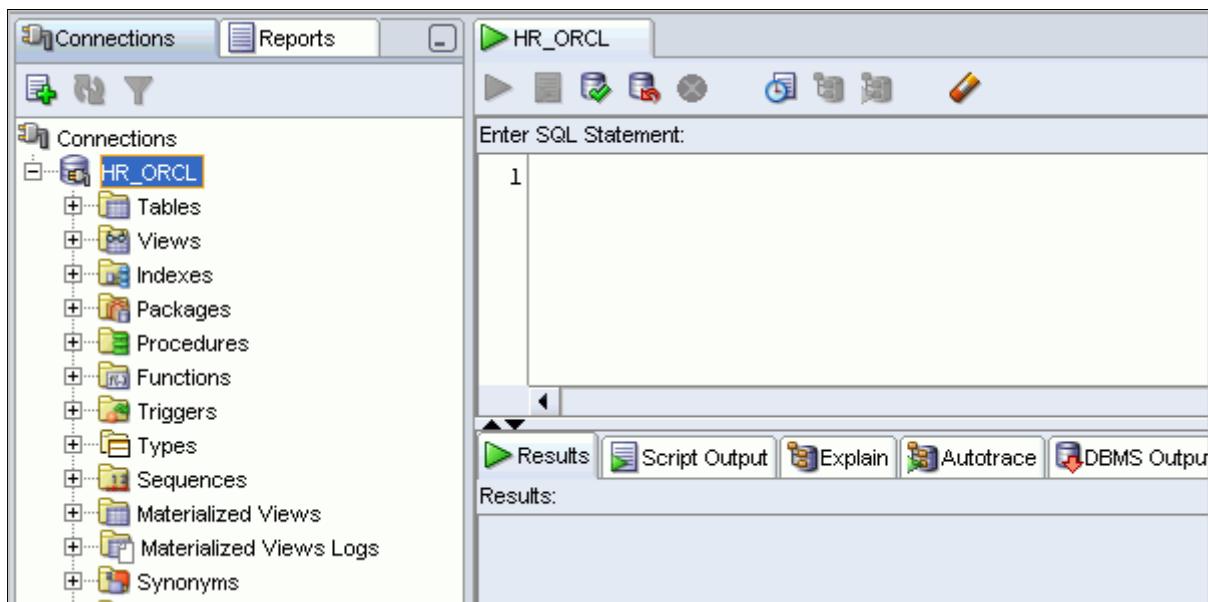
4. The status of the connection was tested successfully. The connection was not saved however. To save the connection, click **Connect**.



5. The connection was saved and you see the database in the list. Expand **HR_ORCL**.



6. When a connection is opened, a SQL Worksheet is opened automatically. The SQL Worksheet allows you to execute SQL against the connection you just created.

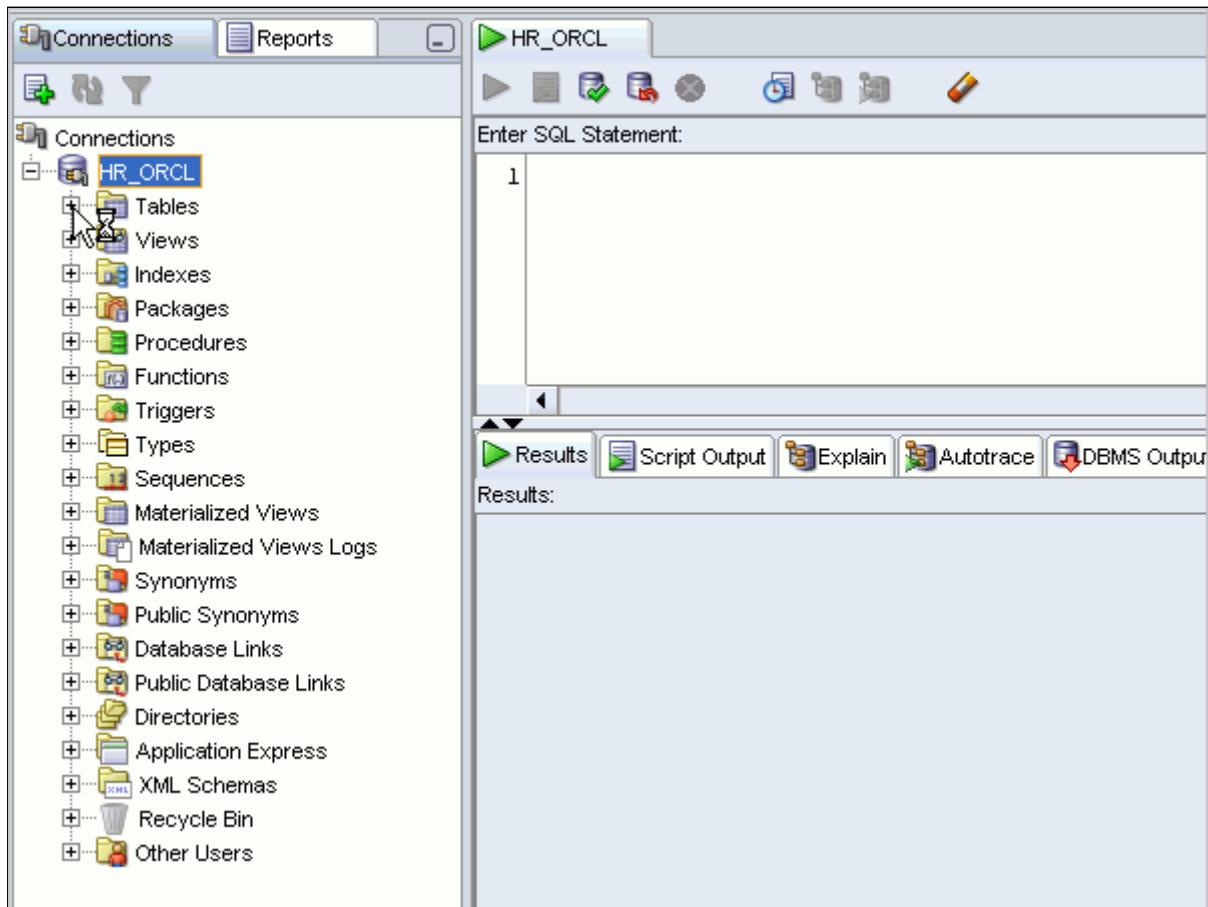


[Back to Topic List](#)

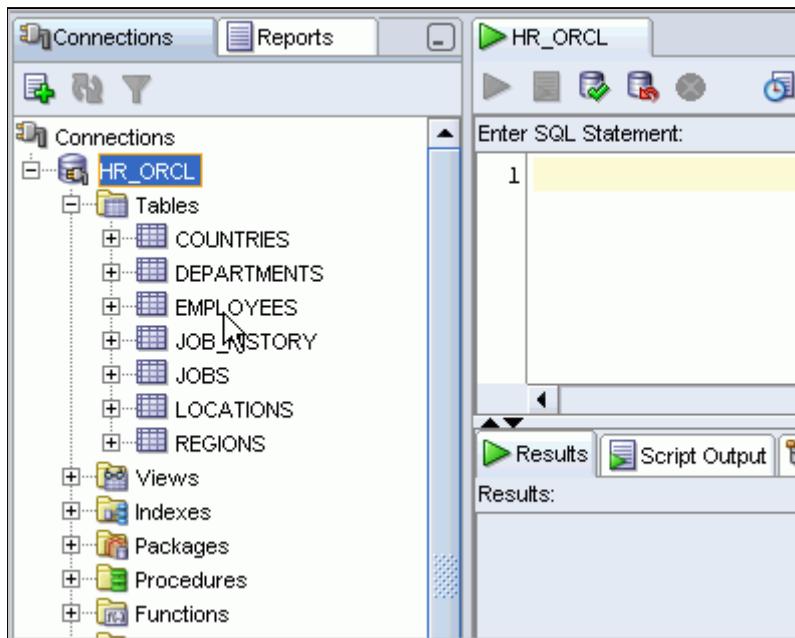
Browsing Your Database

The Connections Navigator in Oracle SQL Developer allows you to browse and edit database objects. This tutorial creates and debugs PL/SQL and uses a selection of tables from the HR schema. In this topic, you review the tables you will use later in the tutorial. Perform the following steps:

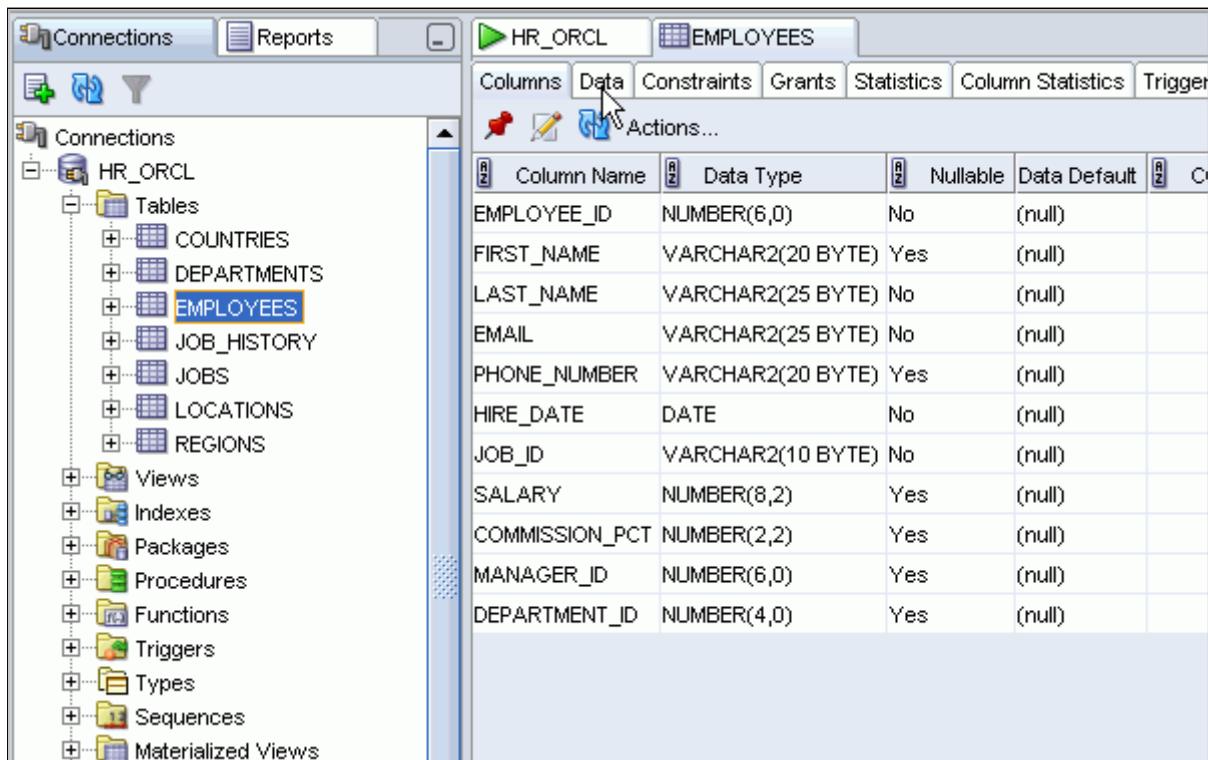
1. Expand the **Tables** node.



2. Click the **EMPLOYEES** table to view the table definition.



3. To see the data, click the **Data** tab.



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

Connections Reports

HR_ORCL EMPLOYEES

Columns Data Constraints Grants Statistics Column Statistics Trigger

EMPLOYEE_ID FIRST_NAME LAST_NAME EMAIL

1	198 Donald	OConnell	DOCONN.
2	199 Douglas	Grant	DGRANT
3	200 Jennifer	Whalen	JWHALEN
4	201 Michael	Hartstein	MHARTS.
5	202 Pat	Fay	PFAY
6	203 Susan	Mavris	SMAVRIS
7	204 Hermann	Baer	HBAER
8	205 Shelley	Higgins	SHIGGINS
9	206 William	Gietz	WGIETZ
10	100 Steven	King	SKING
11	101 Neena	Kochhar	NKOCHH.
12	102 Lex	De Haan	LDEHAAN
13	103 Alexander	Hunold	AHUNOLD
14	104 Bruce	Ernst	BERNST

5. There are a number of constraints for the DEPARTMENTS table. When the table definition displays, select the **Constraints** tab.

Connections Reports

HR_ORCL DEPARTMENTS

Columns Data Constraints Grants Statistics Column Statistics Trigger

DEPARTMENT_ID DEPARTMENT_NAME MANAGER_ID

1	10 Administration	2
2	20 Marketing	2
3	30 Purchasing	1
4	40 Human Resources	2
5	50 Shipping	1
6	60 IT	1
7	70 Public Relations	2
8	80 Sales	1
9	90 Executive	1
10	100 Finance	1
11	110 Accounting	2
12	120 Treasury	(null)
13	130 Corporate Tax	(null)

6. Click the **Edit** icon.

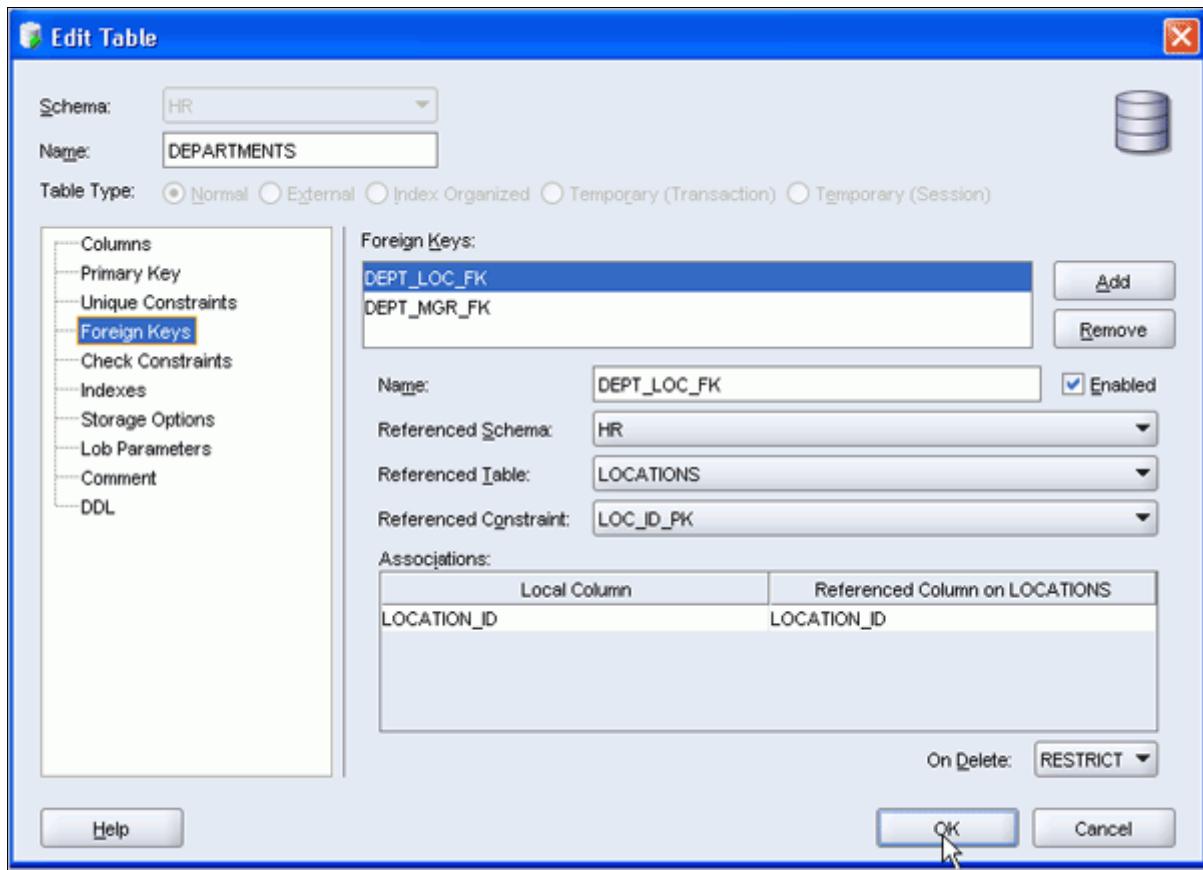
The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows 'HR_ORCL' selected. Under 'Tables', 'DEPARTMENTS' is selected. The main panel displays the 'DEPARTMENTS' table with its constraints. The 'Actions...' button, which typically provides context-specific options like 'Edit', is highlighted with a cursor.

Constraint Name	Constraint Type	Search Condition
DEPT_ID_PK	Primary_Key	(null)
DEPT_LOC_FK	Foreign_Key	(null)
DEPT_MGR_FK	Foreign_Key	(null)
DEPT_NAME_NN	Check	"DEPARTMENT_NAME" IS NOT NULL

7. The dialog has a number of tabs, select the **Foreign Keys** tab.

The screenshot shows the 'Edit Table' dialog for the 'DEPARTMENTS' table. The 'Schema' is set to 'HR'. The 'Name' is 'DEPARTMENTS'. The 'Table Type' is 'Normal'. The 'Columns' tab is selected, showing the columns: 'DEPARTMENT_ID', 'DEPARTMENT_NAME', 'MANAGER_ID', and 'LOCATION_ID'. The 'Column Properties' section is configured for 'DEPARTMENT_ID': Name is 'DEPARTMENT_ID', Datatype is 'Simple' (selected), Type is 'NUMBER', Precision is 4, Scale is 0, and 'Cannot be NULL' is checked. The 'Comment' field contains 'Primary key column of departments table.'.

Review the Foreign Keys. Then click **OK**.



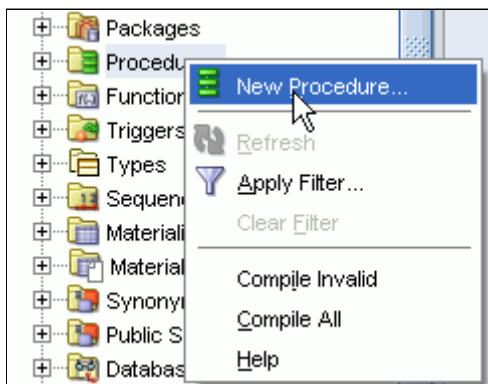
8. Verify that the JOBS and LOCATIONS tables exist, and have data, by selecting each in the Navigator in turn and reviewing the definitions and data.

[Back to Topic List](#)

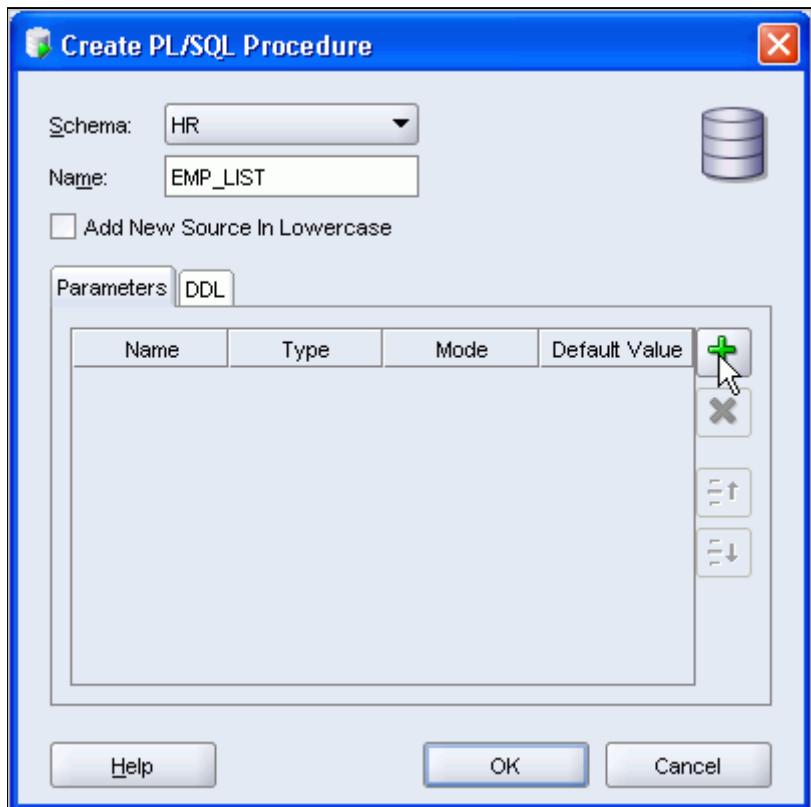
Creating and Compiling a PL/SQL Procedure

In this topic you create, edit and compile a PL/SQL procedure. Perform the following steps:

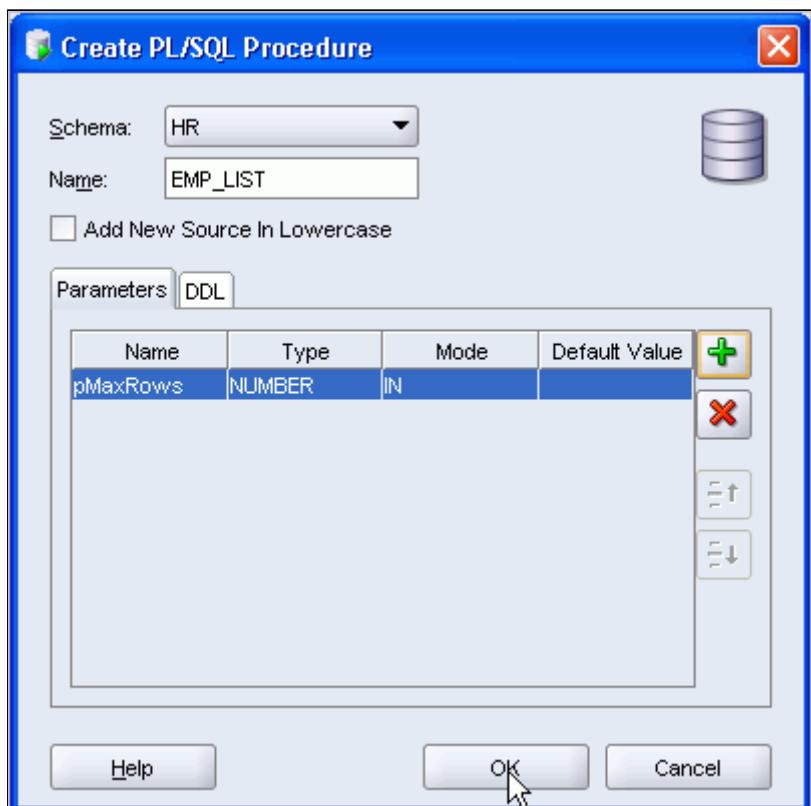
1. Right-click on the **Procedures** node in the Connections Navigator, to invoke the context menu, and select **Create PROCEDURE**.



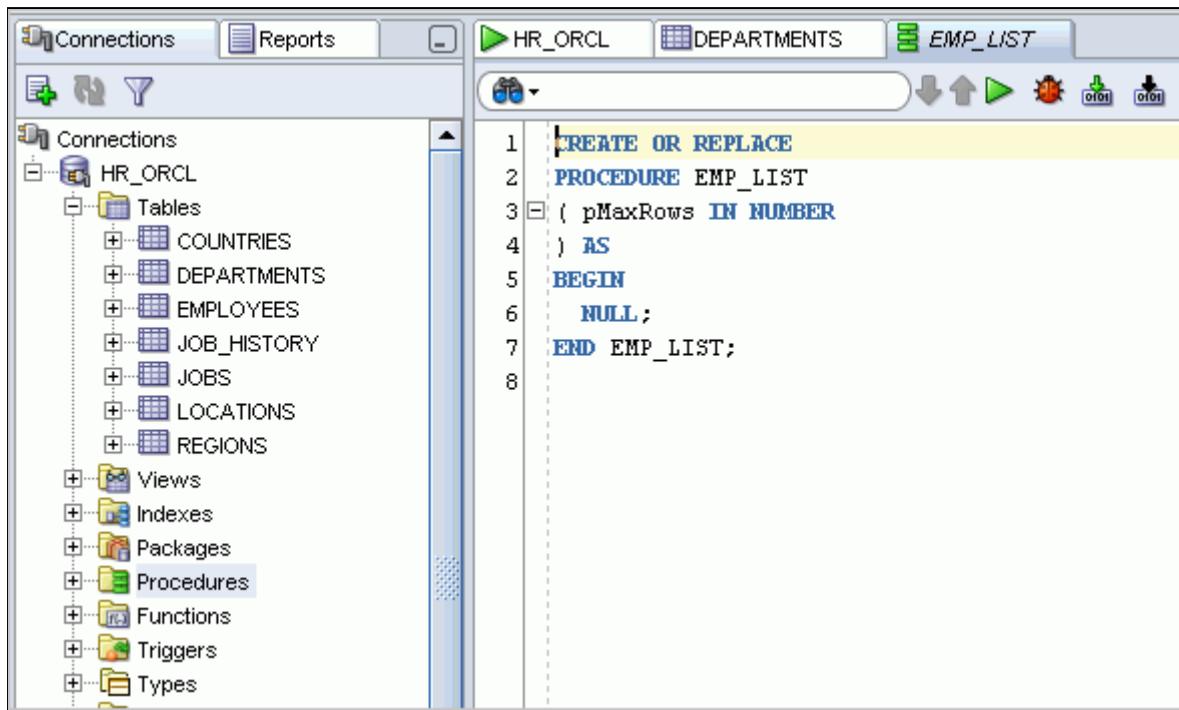
2. Enter **EMP_LIST** as the procedure name. Then click the + to add a Parameter.



3. Double-click on param to allow you to change the value to **pMaxRows** and then change VARCHAR2 to **NUMBER**. Make sure you press enter before you click **OK**.



4. The skeleton of the procedure with the parameter specified is displayed.



5. Replace the following PL/SQL:

```

BEGIN
NULL;
END;

```

With the following code:

(You will find the code in the file **emp_cursor.sql** in the directory **\PLSQL\files**)

```

CURSOR emp_cursor IS
  SELECT l.state_province, l.country_id, d.department_name, e.last_name,
         j.job_title, e.salary, e.commission_pct
    FROM locations l, departments d, employees e, jobs j
   WHERE l.location_id = d.location_id
     AND d.department_id = e.department_id
     AND e.job_id = j.job_id;
  emp_record emp_cursor%ROWTYPE;
  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
  emp_tab emp_tab_type;
i NUMBER := 1;
BEGIN
  OPEN emp_cursor;
  FETCH emp_cursor INTO emp_record;
  emp_tab(i) := emp_record;
  WHILE ((emp_cursor%FOUND) AND (i <= pMaxRows)) LOOP
    i := i + 1;
    FETCH emp_cursor INTO emp_record;
    emp_tab(i) := emp_record;
  END LOOP;
  CLOSE emp_cursor;
  FOR j IN REVERSE 1..i LOOP
    DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
  END LOOP;
END;

```

Notice how the reserved words are formatted by Oracle SQL Developer. To format the code further, right-click to invoke the context menu and select Format SQL...

Compile the PL/SQL subprogram by clicking the Save button in the toolbar.

```

4  ) AS
5  CURSOR emp_cursor IS
6    SELECT l.state_province, l.country_id, d.department_name, e.la
7      j.job_title, e.salary, e.commission_pct
8    FROM locations l, departments d, employees e, jobs j
9    WHERE l.location_id = d.location_id
10   AND d.department_id = e.department_id
11   AND e.job_id = j.job_id;
12   emp_record emp_cursor%ROWTYPE;
13   TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINA
14   emp_tab emp_tab_type;
15   i NUMBER := 1;
16  BEGIN
17    OPEN emp_cursor;
18    FETCH emp_cursor INTO emp_record;
19    emp_tab(i) := emp_record;
20  WHILE ((emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21    i := i + 1;
22    FETCH emp_cursor INTO emp_record;
23    emp_tab(i) := emp_record;
24  END LOOP;
25  CLOSE emp_cursor;
26  FOR j IN REVERSE 1..i LOOP
27    DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28  END LOOP;
29 END;

```

6. Expand **Procedures** in the navigator.

```

1 CREATE OR REPLACE
2 PROCEDURE EMP_LIST
3 ( pMaxRows IN NUMBER
4 ) AS
5 CURSOR emp_cursor IS
6   SELECT l.state_province, l.country_id, d.department_name, e.la
7     j.job_title, e.salary, e.commission_pct
8   FROM locations l, departments d, employees e, jobs j
9   WHERE l.location_id = d.location_id
10  AND d.department_id = e.department_id
11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINA
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17  OPEN emp_cursor;
18  FETCH emp_cursor INTO emp_record;
19  emp_tab(i) := emp_record;

```

Messages - Log

EMP_LIST Compiled (with errors)

7. Note that when an invalid PL/SQL subprogram is detected by Oracle SQL Developer, the status is indicated with a red X over the icon for the subprogram in the Connections Navigator.

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays the HR_ORCL schema with various objects like Tables, Views, Procedures, and Functions. The Procedures node is expanded, showing ADD_JOB_HISTORY, EMP_LIST, and SECURE_DML. The EMP_LIST procedure is selected and its code is displayed in the central editor pane:

```

1 CREATE OR REPLACE
2 PROCEDURE EMP_LIST
3 ( pMaxRows IN NUMBER
4 ) AS
5 CURSOR emp_cursor IS
6   SELECT l.state_province, l.country_id, d.department_name, e.job_title, e.salary, e.commission_pct
7   FROM locations l, departments d, employees e, jobs j
8   WHERE l.location_id = d.location_id
9   AND d.department_id = e.department_id
10  AND e.job_id = j.job_id;
11  emp_record emp_cursor%ROWTYPE;
12  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BIN
13  emp_tab emp_tab_type;
14  i NUMBER := 1;
15
16 BEGIN
17  OPEN emp_cursor;
18  FETCH emp_cursor INTO emp_record;
19  emp_tab(i) := emp_record;

```

Below the editor is the Compiler - Log window, which shows:

EMP_LIST Compiled (with errors)

At the bottom of the interface are tabs for Migration Log, Logging Page, Compiler, and Messages.

Compilation errors are shown in the log window. You can navigate to the line reported in the error by simply double-clicking on the error. Oracle SQL Developer also displays errors and hints in the right hand gutter. If you hover over each of the red bars in the gutter, the error message displays

In this case, the error messages indicate that there is a formatting error in the LOOP statement. After reviewing the code further, you see an extra parenthesis in the WHILE statement. Delete the extra parenthesis.

The screenshot shows the Oracle SQL Developer interface after the extra parenthesis in the WHILE statement was removed. The code now looks like this:

```

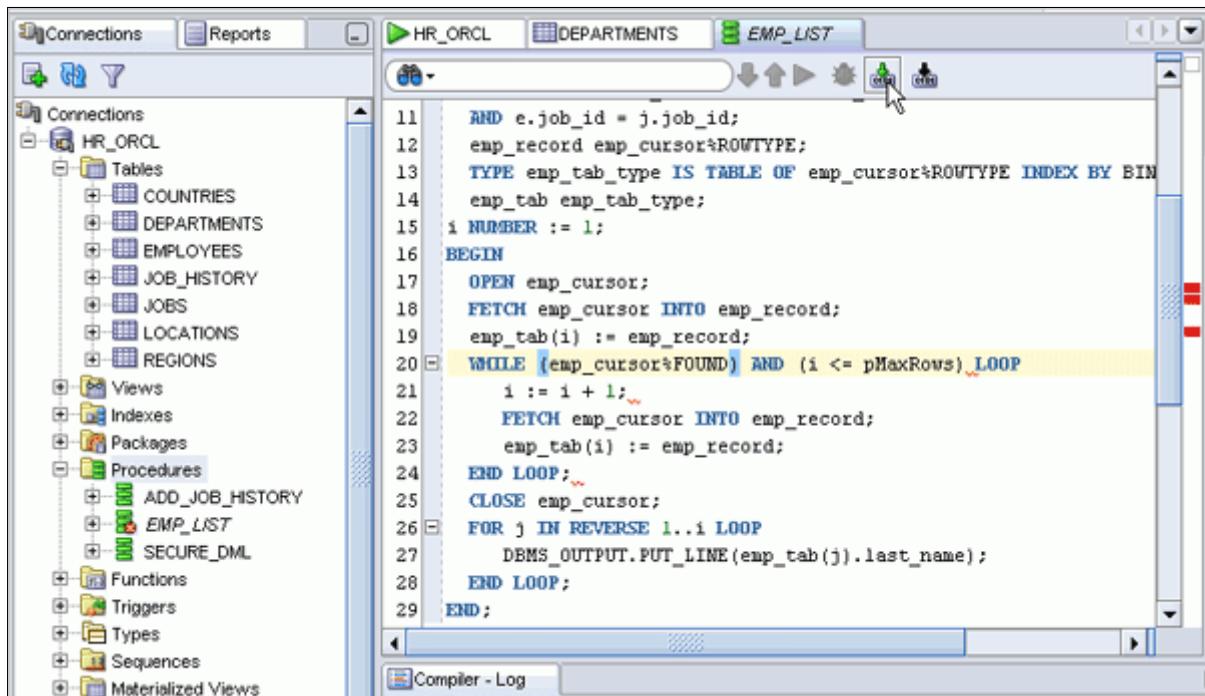
11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BIN
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16 BEGIN
17  OPEN emp_cursor;
18  FETCH emp_cursor INTO emp_record;
19  emp_tab(i) := emp_record;
20  WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21    i := i + 1;
22    FETCH emp_cursor INTO emp_record;
23    emp_tab(i) := emp_record;
24  END LOOP;
25  CLOSE emp_cursor;
26  FOR j IN REVERSE 1..i LOOP
27    DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28  END LOOP;
29 END;

```

The Compiler - Log window now shows three errors:

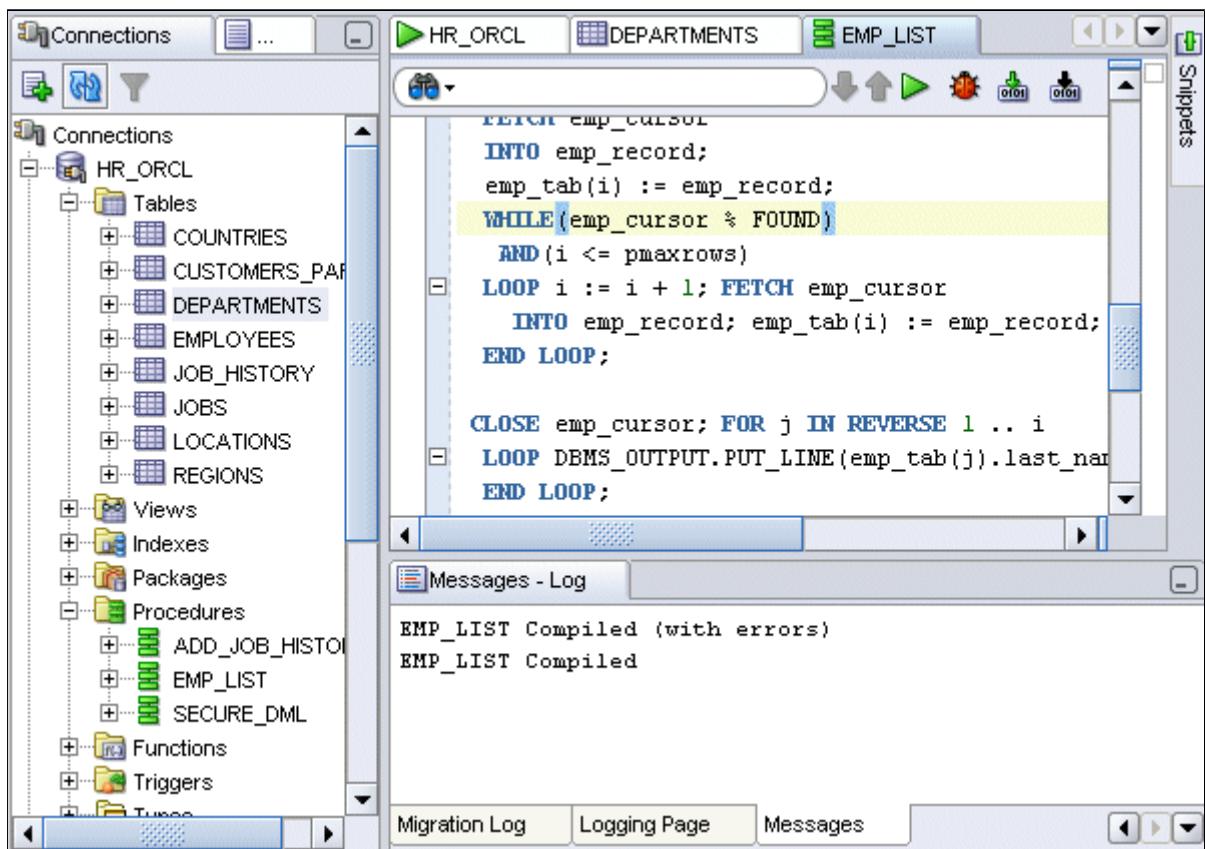
- Error(20,49): PLS-00103: Encountered the symbol "LOOP" when expecting one of the following:
- Error(22,6): PLS-00103: Encountered the symbol 'FETCH' when expecting one of the following:
- Error(25,3): PLS-00103: Encountered the symbol "CLOSE" when expecting one of the following:

8. Click the **Compile** icon.



```
11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BIN
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17      OPEN emp_cursor;
18      FETCH emp_cursor INTO emp_record;
19      emp_tab(i) := emp_record;
20  WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21      i := i + 1;
22      FETCH emp_cursor INTO emp_record;
23      emp_tab(i) := emp_record;
24  END LOOP;
25  CLOSE emp_cursor;
26  FOR j IN REVERSE 1..i LOOP
27      DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28  END LOOP;
29 END;
```

9. The Procedure compiled successfully. You are now ready to run the procedure.



```
11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BIN
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17      OPEN emp_cursor;
18      FETCH emp_cursor INTO emp_record;
19      emp_tab(i) := emp_record;
20  WHILE (emp_cursor%FOUND)
21      AND (i <= pmaxrows)
22  LOOP
23      i := i + 1; FETCH emp_cursor
24          INTO emp_record; emp_tab(i) := emp_record;
25  END LOOP;

26  CLOSE emp_cursor; FOR j IN REVERSE 1 .. i
27  LOOP DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28  END LOOP;
```

Messages - Log

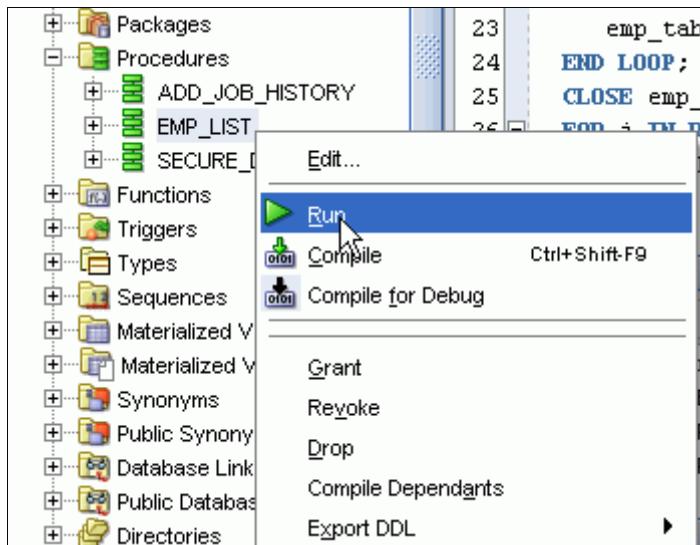
```
EMP_LIST Compiled (with errors)
EMP_LIST Compiled
```

[Back to Topic List](#)

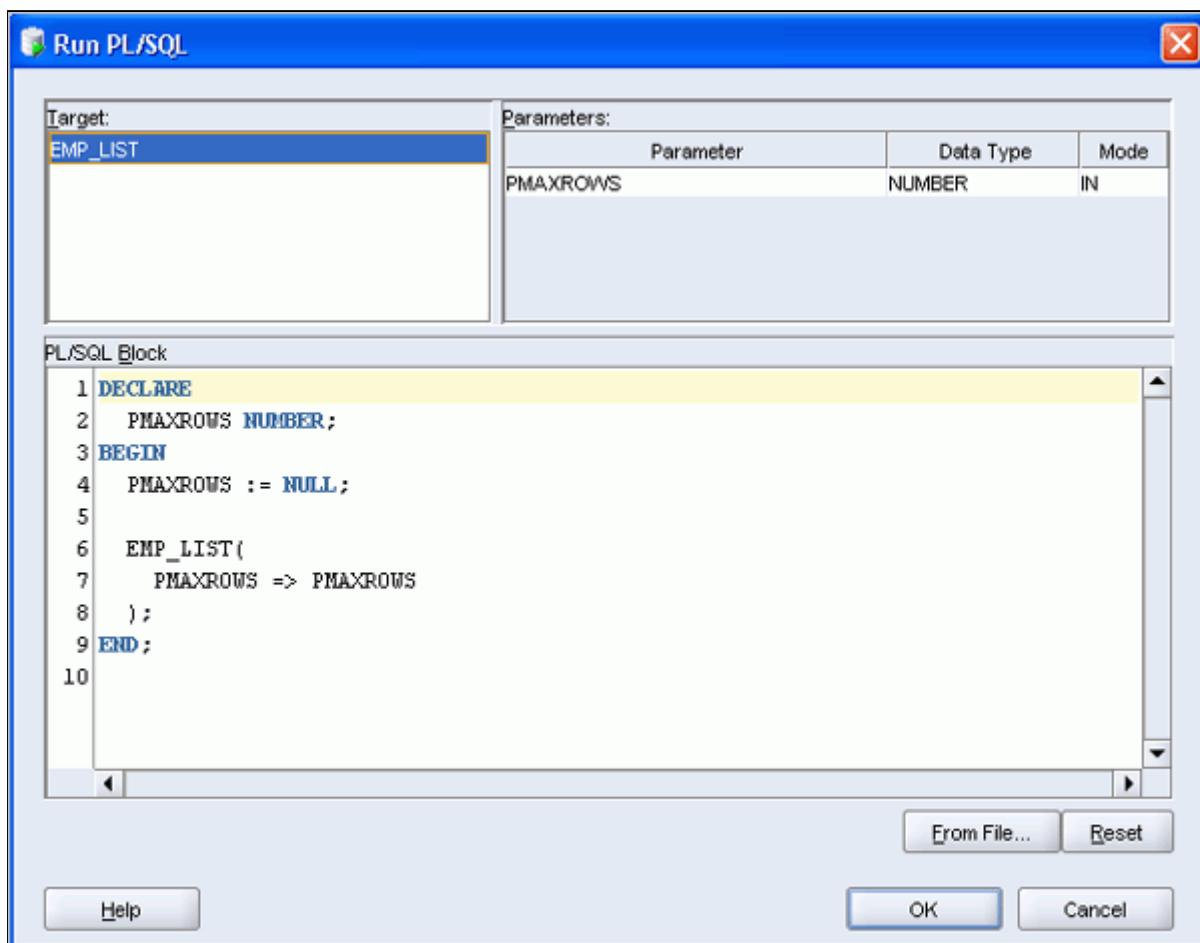
Running a PL/SQL Procedure

Once you have created and compiled a PL/SQL procedure, you can run it using Oracle SQL Developer. Perform the following steps:

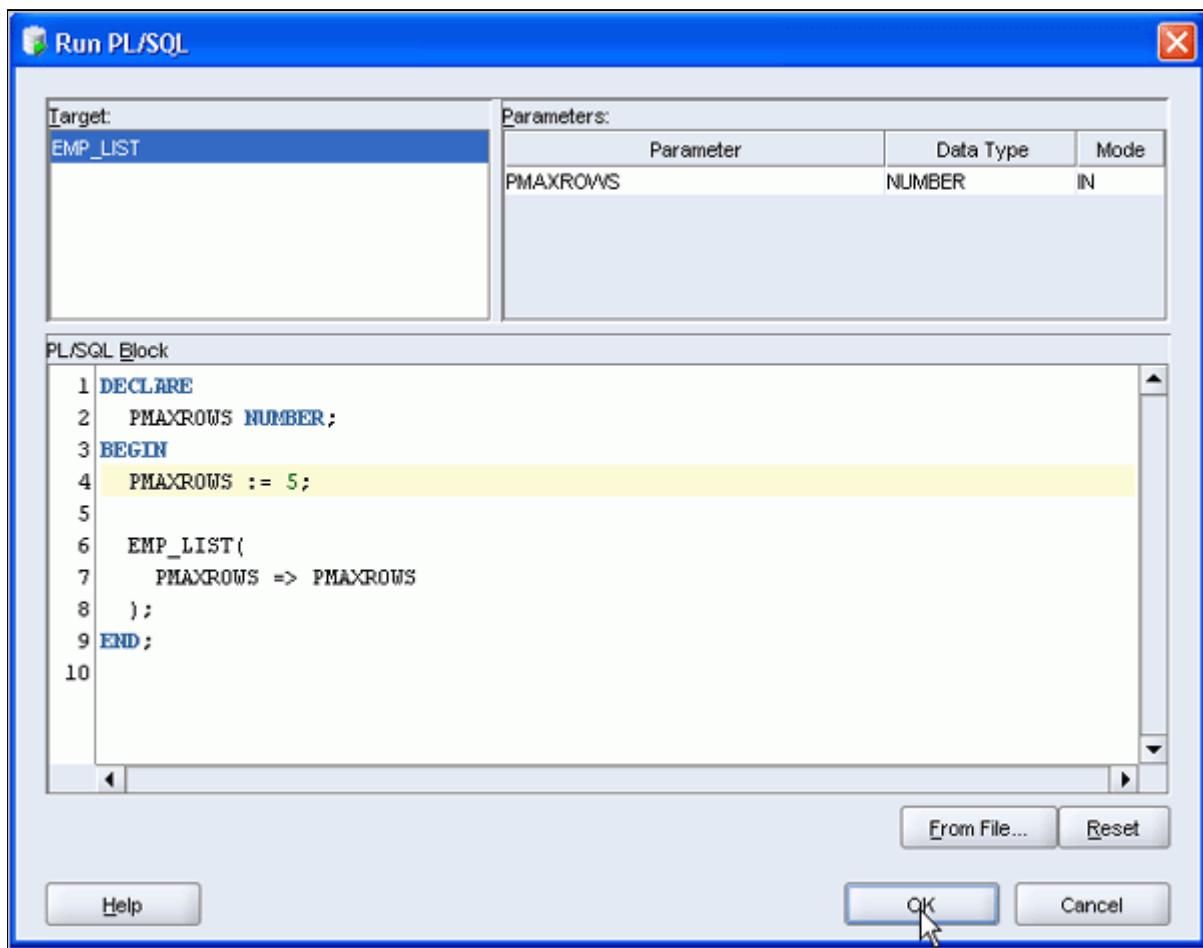
1. Right-click on **EMP_LIST** in the left navigator and select **Run**.



2. This invokes the Run PL/SQL dialog. The Run PL/SQL dialog allows you to select the target procedure or function to run (useful for packages) and displays a list of parameters for the selected target. In the PL/SQL block text area is generated code that Oracle SQL Developer uses to call the selected program. You can use this area to populate parameters to be passed to the program unit and to handle complex return types.



Change `PMAXROWS := NULL;` to `PMAXROWS := 5;` Then click **OK**.



- The results of the 5 rows returned are displayed in the log window.

```

11   AND e.job_id = j.job_id;
12   emp_record emp_cursor%ROWTYPE;
13   TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
14   emp_tab emp_tab_type;
15   i NUMBER := 1;
16   BEGIN
17     OPEN emp_cursor;
18     FETCH emp_cursor INTO emp_record;
19     emp_tab(i) := emp_record;
20   WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21     i := i + 1;
22     FETCH emp_cursor INTO emp_record;
23     emp_tab(i) := emp_record;
24   END LOOP;
25   CLOSE emp_cursor;

```

Running - Log

Connecting to the database HR_ORCL.
Popp
Greenberg
Whalen
Kochhar
De Haan
King
Process exited.
Disconnecting from the database HR_ORCL.

Debugging a PL/SQL Procedure

Oracle SQL Developer also supports PL/SQL debugging with Oracle databases. In this topic, you debug a PL/SQL Procedure, step through the code and modify a value at runtime. Perform the following steps:

- Set a breakpoint in the EMP_LIST procedure by clicking in the margin at the line with the `OPEN emp_cursor;` statement.

```

11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17  OPEN emp_cursor;
18  FETCH emp_cursor INTO emp_record;
19  emp_tab(i) := emp_record;
20  WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21      i := i + 1;
22      FETCH emp_cursor INTO emp_record;
23      emp_tab(i) := emp_record;
24  END LOOP;

```

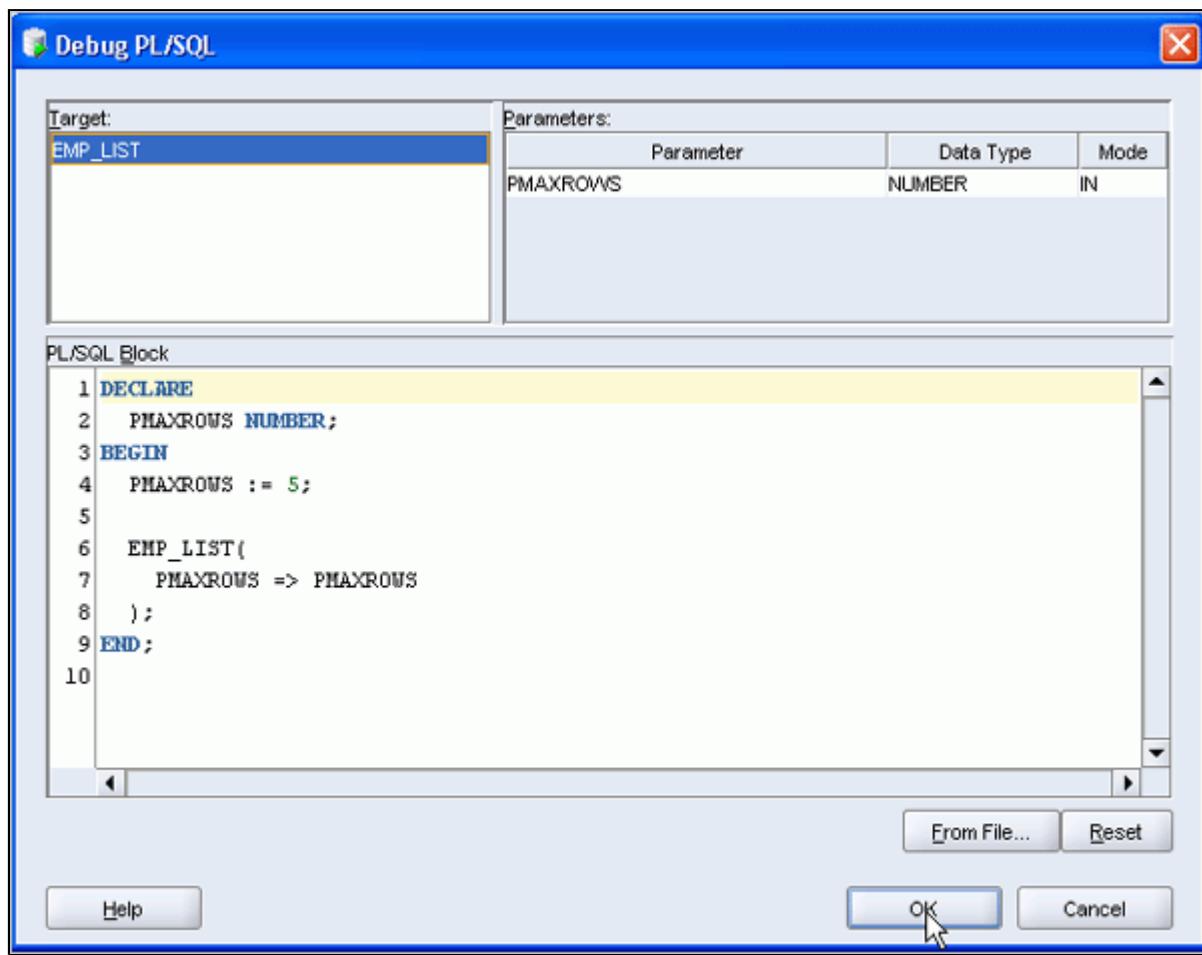
- Click the **Debug** icon (ladybug).

```

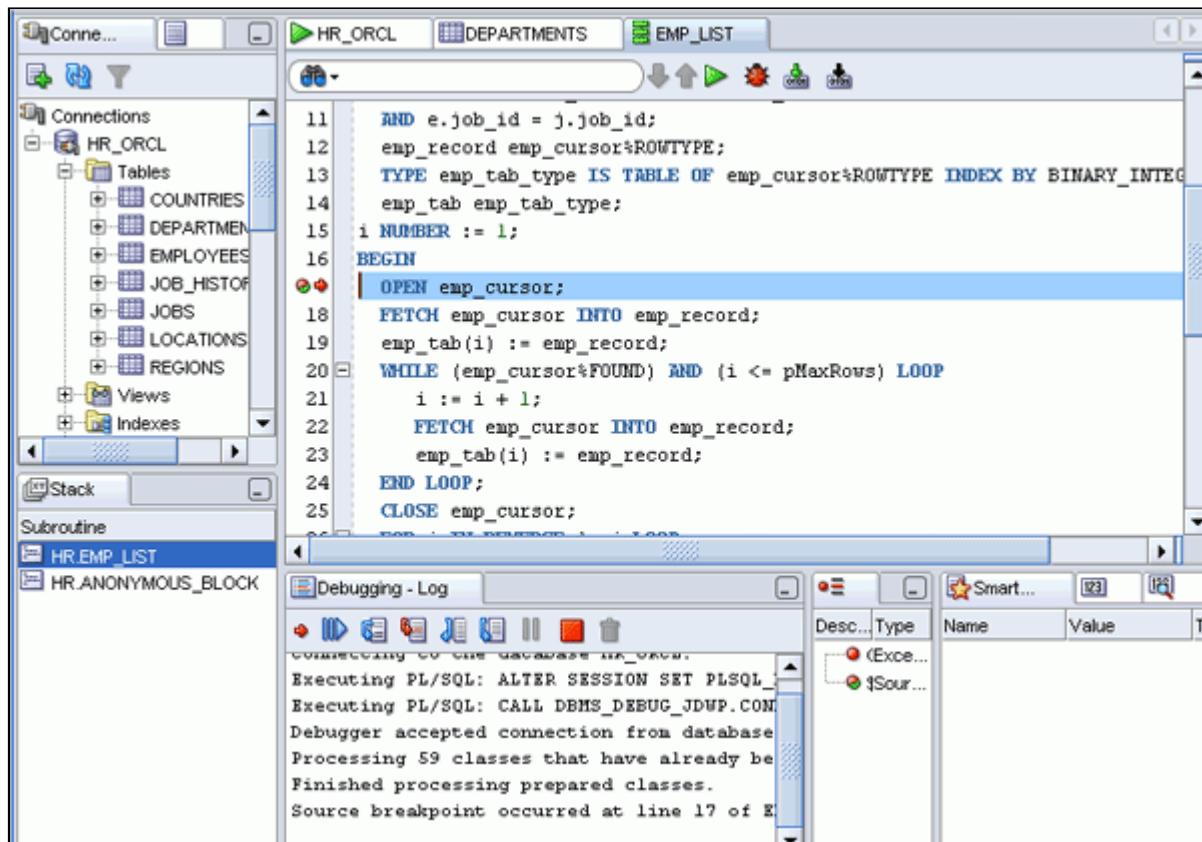
11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BIN
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17  ● OPEN emp_cursor;
18  FETCH emp_cursor INTO emp_record;
19  emp_tab(i) := emp_record;
20  WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21      i := i + 1;
22      FETCH emp_cursor INTO emp_record;
23      emp_tab(i) := emp_record;
24  END LOOP;
25  CLOSE emp_cursor;

```

- The Debug PL/SQL dialog should still show the value `PMAXROWS = 5`; Click **OK**.



4. The debugger should halt at the line where you placed the breakpoint. You can now control the flow of execution, modify values of variables and perform other debugging functions.



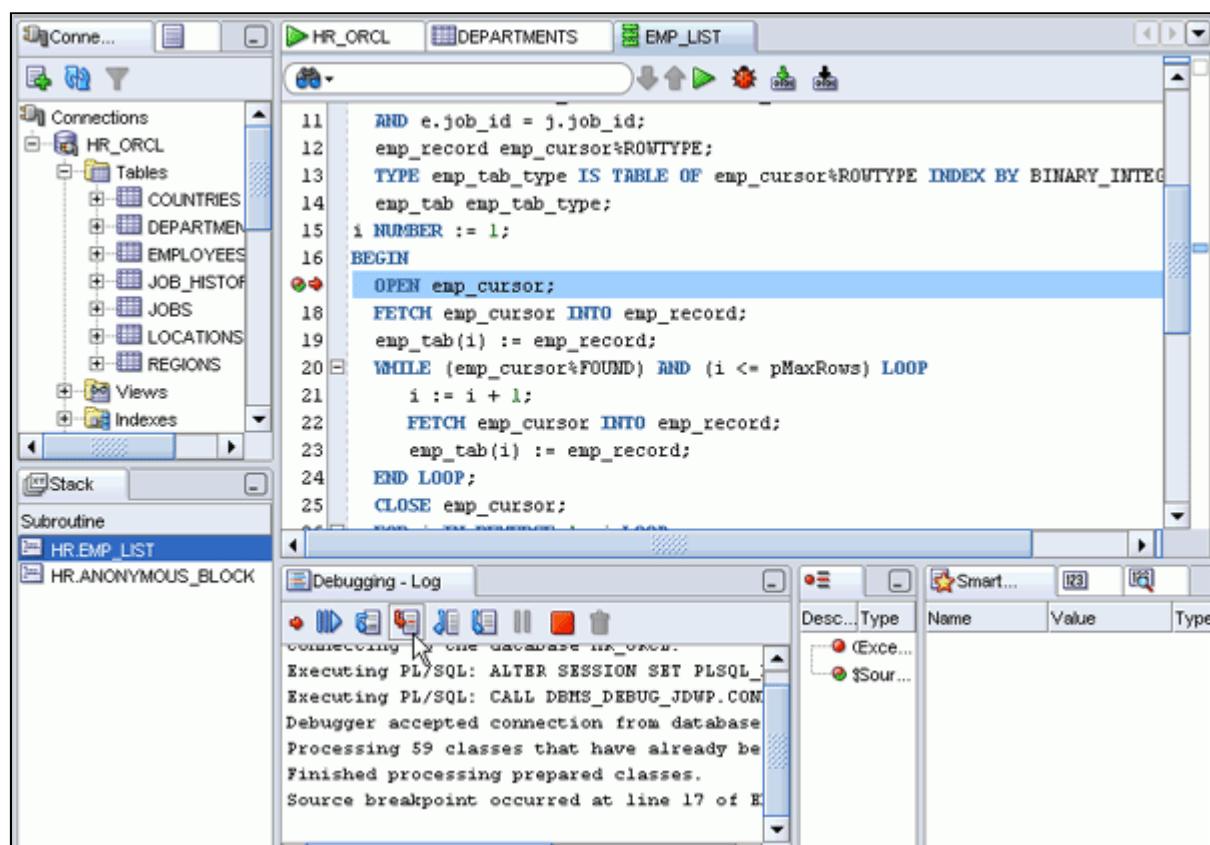
Note: If at this point you receive the error message "This session requires DEBUG CONNECT SESSION and DEBUG ANY PROCEDURE user privileges", you need to complete Step 4 of [Prerequisites](#).

```
Debugging - Log

Connecting to the database HR_ORCL.
Executing PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE
Executing PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_TCP( '127.0.0.1', '4127' )
ORA-01031: insufficient privileges
ORA-06512: at "SYS.DBMS_DEBUG_JDWP", line 68
ORA-06512: at line 1
This session requires DEBUG CONNECT SESSION and DEBUG ANY PROCEDURE user privileges.
Process exited.
Disconnecting from the database HR_ORCL.

Messages Data Editor Running Debugging
```

5. Click Step Into .



The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to HR_ORCL. In the center, the code editor displays the PL/SQL code for the EMP_LIST procedure. The code includes various declarations and a cursor loop. The line 'OPEN emp_cursor;' is highlighted. Below the code editor is the Debugging - Log window, which shows the connection process and the acceptance of the debugger's connection. A tooltip for the Step Into button is visible, indicating it will take you to the first line of the cursor.

```
11    AND e.job_id = j.job_id;
12    emp_record emp_cursor%ROWTYPE;
13    TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
14    emp_tab emp_tab_type;
15    i NUMBER := 1;
16
17    BEGIN
18        OPEN emp_cursor;
19        FETCH emp_cursor INTO emp_record;
20        emp_tab(i) := emp_record;
21        WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
22            i := i + 1;
23            FETCH emp_cursor INTO emp_record;
24            emp_tab(i) := emp_record;
25        END LOOP;
26        CLOSE emp_cursor;
```

```
Debugging - Log
Connecting to the database HR_ORCL.
Executing PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE
Executing PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_TCP( '127.0.0.1', '4127' )
Debugger accepted connection from database
Processing 59 classes that have already been loaded.
Finished processing prepared classes.
Source breakpoint occurred at line 17 of EMP_LIST.
```

6. This takes you to the first line of the cursor. Click Step Into  again.

```
1 CREATE OR REPLACE
2 PROCEDURE EMP_LIST
3 ( pMaxRows IN NUMBER
4 ) AS
5 CURSOR emp_cursor IS
6   SELECT l.state_province, l.country_id, d.department_name, e.last_name,
7         j.job_title, e.salary, e.commission_pct
8   FROM locations l, departments d, employees e, jobs j
9  WHERE l.location_id = d.location_id
10 AND d.department_id = e.department_id
11 AND e.job_id = j.job_id;
12 emp_record emp_cursor%ROWTYPE;
13 TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
14 emp_tab emp_tab_type;
15 i NUMBER := 1;
```

Debugging - Log

- Connecting to the database HR_ORCL.
- Executing PL/SQL: ALTER SESSION SET PLSQL_
- Executing PL/SQL: CALL DBMS_DEBUG_JDWP.COM
- Debugger accepted connection from database
- Processing 59 classes that have already be
- Finished processing prepared classes.
- Source breakpoint occurred at line 17 of E

7. You should now be selecting the first row of the cursor. Click **Step Into** 3 more times.

```
1 CREATE OR REPLACE
2 PROCEDURE EMP_LIST
3 ( pMaxRows IN NUMBER
4 ) AS
5 CURSOR emp_cursor IS
6   SELECT l.state_province, l.country_id, d.department_name, e.last_name,
7         j.job_title, e.salary, e.commission_pct
8   FROM locations l, departments d, employees e, jobs j
9  WHERE l.location_id = d.location_id
10 AND d.department_id = e.department_id
11 AND e.job_id = j.job_id;
12 emp_record emp_cursor%ROWTYPE;
13 TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
14 emp_tab emp_tab_type;
15 i NUMBER := 1;
```

Debugging - Log

- Connecting to the database HR_ORCL.
- Executing PL/SQL: ALTER SESSION SET PLSQL_
- Executing PL/SQL: CALL DBMS_DEBUG_JDWP.COM
- Debugger accepted connection from database
- Processing 59 classes that have already be
- Finished processing prepared classes.
- Source breakpoint occurred at line 17 of E

8. Click the **Data** tab.

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the Connections tree shows HR_ORCL selected. The main pane displays a PL/SQL block:

```

11  AND e.job_id = j.job_id;
12  emp_record emp_cursor%ROWTYPE;
13  TYPE emp_tab_type IS TABLE OF emp_cursor%ROWTYPE INDEX BY BINARY_INTEGER;
14  emp_tab emp_tab_type;
15  i NUMBER := 1;
16  BEGIN
17      OPEN emp_cursor;
18      FETCH emp_cursor INTO emp_record;
19      emp_tab(i) := emp_record;
20      WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21          i := i + 1;
22          FETCH emp_cursor INTO emp_record;
23          emp_tab(i) := emp_record;
24      END LOOP;
25      CLOSE emp_cursor;

```

The line `OPEN emp_cursor;` is highlighted in red, indicating it is the current line of execution. The bottom-right pane, titled "Data", contains a table with one row:

Name	Type
(Exce...)	
\$Sour...	

9. The **Data** window starts to show a limited list of variables which are used in the line of code that is about to be executed, and in the previously executed line.

The screenshot shows the Oracle SQL Developer interface. The code editor and debugging log are identical to the previous screenshot. The "Data" window is now fully visible and shows the following variable list:

Name	Type
PMAXROWS	NUMBER
EMP_RECORD	Rowtype
EMP_TAB	Index By Binary Integer
i	NUMBER

10. Right-click the line that reads `DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);` and select **Run to Cursor**.

```

16 BEGIN
17   OPEN emp_cursor;
18   FETCH emp_cursor INTO emp_record;
19   emp_tab(i) := emp_record;
20   WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21     i := i + 1;
22     FETCH emp_cursor INTO emp_record;
23     emp_tab(i) := emp_record;
24   END LOOP;
25   CLOSE emp_cursor;
26   FOR j IN REVERSE 1..i LOOP
27     DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28   END LOOP;
29 END;

```

Stack

Subroutine

- HR.EMP_LIST
- HR.ANONYMOUS_BLOCK

Debugging - Log

executing PL/SQL: anonymous block

Executing PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_LISTENER

Debugger accepted connection from database on port 58080

Processing 59 classes that have already been prepared

Finished processing prepared classes.

Source breakpoint occurred at line 17 of EMP_LIST

Watches

Name	Type	Value	Type
MAXROWS	NUMBER	5	NUMBER
_RECORD	Rowtype		
_TAB	index... EMP_TA...	1	NUMBER

11. Expand **emp_tab > values > [1] > _value**. You see the values of the fields in a given record of the table. Select the **LAST_NAME** field.

```

19 emp_tab(i) := emp_record;
20 WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21   i := i + 1;
22   FETCH emp_cursor INTO emp_record;
23   emp_tab(i) := emp_record;
24 END LOOP;
25 CLOSE emp_cursor;
26 FOR j IN REVERSE 1..i LOOP
27   DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28 END LOOP;
29 END;

```

Smart Data

Connecting to the database HR_ORCL.

Executing PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE

Executing PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_LISTENER

Debugger accepted connection from database on port 58080

Processing 59 classes that have already been prepared

Finished processing prepared classes.

Source breakpoint occurred at line 17 of EMP_LIST

Name	Type	Value	Type
[1]	EMP_TAB...		
_key	PLS_INTEGER	1	PLS_INTEGER
_value	Rowtype		
STATE_PROVINCE	VARCHAR2	Vashi...	VARCHAR2
COUNTRY_ID	CHAR(2)	US	CHAR(2)
DEPARTMENT_NAME	VARCHAR2	Executive...	VARCHAR2
LAST_NAME	VARCHAR2	King	VARCHAR2
JOB_TITLE	VARCHAR2	President...	VARCHAR2
SALARY	NUMBER(8,2)	24000	NUMBER(8,2)
COMMISSION_PCT	NUMBER(2,2)	NULL	NUMBER(2,2)

12. Right-click the **LAST_NAME** field and select **Modify Value**.

```

19: emp_tab(i) := emp_record;
20: WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21:   i := i + 1;
22:   FETCH emp_cursor INTO emp_record;
23:   emp_tab(i) := emp_record;
24: END LOOP;
25: CLOSE emp_cursor;
26: FOR j IN REVERSE 1..i LOOP
27:   DEMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28: END LOOP;
29: END;

```

Name	Type
EMP_TAB_1	PLS_INTEGER
STATE_PROVINCE	VARCHAR2
COUNTRY_ID	CHAR(2)
DISTRICT_ID	VARCHAR2
ST_NAME	VARCHAR2
B_TITLE	VARCHAR2
SALARY	NUMBER(8,2)
MISSION_PCT	NUMBER(2,2)
EMP_TAB_2	PLS_INTEGER

13. Change the name to something else and click **OK**.



14. Click the Resume icon  to allow the PL/SQL to run to completion.

```

19:     emp_tab(i) := emp_record;
20:   WHILE (emp_cursor%FOUND) AND (i <= pMaxRows) LOOP
21:     i := i + 1;
22:     FETCH emp_cursor INTO emp_record;
23:     emp_tab(i) := emp_record;
24:   END LOOP;
25:   CLOSE emp_cursor;
26:   FOR j IN REVERSE 1..i LOOP
27:     DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);
28:   END LOOP;
29: END;

```

Name	Type
[1]	PLS_INTEGER
key	Rowtype
value	<ul style="list-style-type: none"> STATE_PROVINCE_WASHINGTON COUNTRY_ID 'US' DEPARTMENT_NAME 'Executive' LAST_NAME 'James' JOB_TITLE 'President' SALARY 24000 COMMISSION_PCTNULL

15. Check to see that your modified value is displayed in the Log window.

Debugger accepted connection from database on port 4762.
Processing 59 classes that have already been prepared...
Finished processing prepared classes.
Source breakpoint occurred at line 17 of EMP_LIST.pls.
Popp
Greenberg
Whalen
Kochhar
De Haan
James
Process exited.

Summary

In this lesson, you learned how to:

- Create a Database Connection
- Browse the Database
- Create and Compile a PL/SQL Procedure
- Run a PL/SQL Procedure
- Debug a PL/SQL Procedure

[Back to Topic List](#)

Place the cursor over this icon to hide all screenshots.

Creating XML Extensions for Oracle SQL Developer

Purpose

This tutorial shows you how to add XML extensions to Oracle SQL Developer.

Time to Complete

Approximately 30 minutes

Topics

This tutorial covers the following topics:

- [Overview](#)
- [Prerequisites](#)
- [Creating a Database Connection](#)
- [Sharing Reports](#)
- [Adding an Extra Tab to Display Column Comments](#)
- [Reviewing the XML File structure](#)
- [Adding the Tab to Describe Sub partitions](#)
- [Adding a Context Menu](#)
- [Summary](#)

Viewing Screenshots

 Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)

Note: Alternatively, you can place the cursor over an individual icon in the following steps to load and view only the screenshot associated with that step. You can hide an individual screenshot by clicking it.

Overview

Oracle SQL Developer is a free graphical tool that enhances productivity and simplifies database development tasks. Using Oracle SQL Developer, users can browse database objects, run SQL statements, edit and debug PL/SQL statements and run reports, whether provided or created.

Developed in Java, Oracle SQL Developer runs on Windows, Linux and the Mac OS X. This is a great advantage to the increasing numbers of developers using alternative platforms. Oracle SQL Developer is built on an extensible framework and, as such, is extensible itself. Users can create basic XML extensions or more involved Java extensions to add utilities or other functionality to the product.

[Back to Topic List](#)

Prerequisites

For this Hands On Session, the following has already been installed for you:

1. Oracle Database 10g.
Note: To repeat this exercise later you can use any Oracle Database above 9.2.0.1
2. Oracle SQL Developer 1.2.1.
Note: Oracle SQL Developer is available for download for FREE from [OTN](#). To install Oracle SQL Developer, unzip it into any directory on your machine.
3. The files you use throughout the tutorial are located in the **\SQLDev_HOS\Extensions\files** directory. The files have also been included in the [extensions.zip](#) file.
4. The shipped **HR** schema.

Creating a Database Connection

To create a database connection, perform the following steps:

1. Open Windows Explorer and double-click <your_path>\sqldeveloper\sqldeveloper.exe.

Note: If you receive a dialog window asking whether you want to migrate settings from a previous release, click **No**.

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



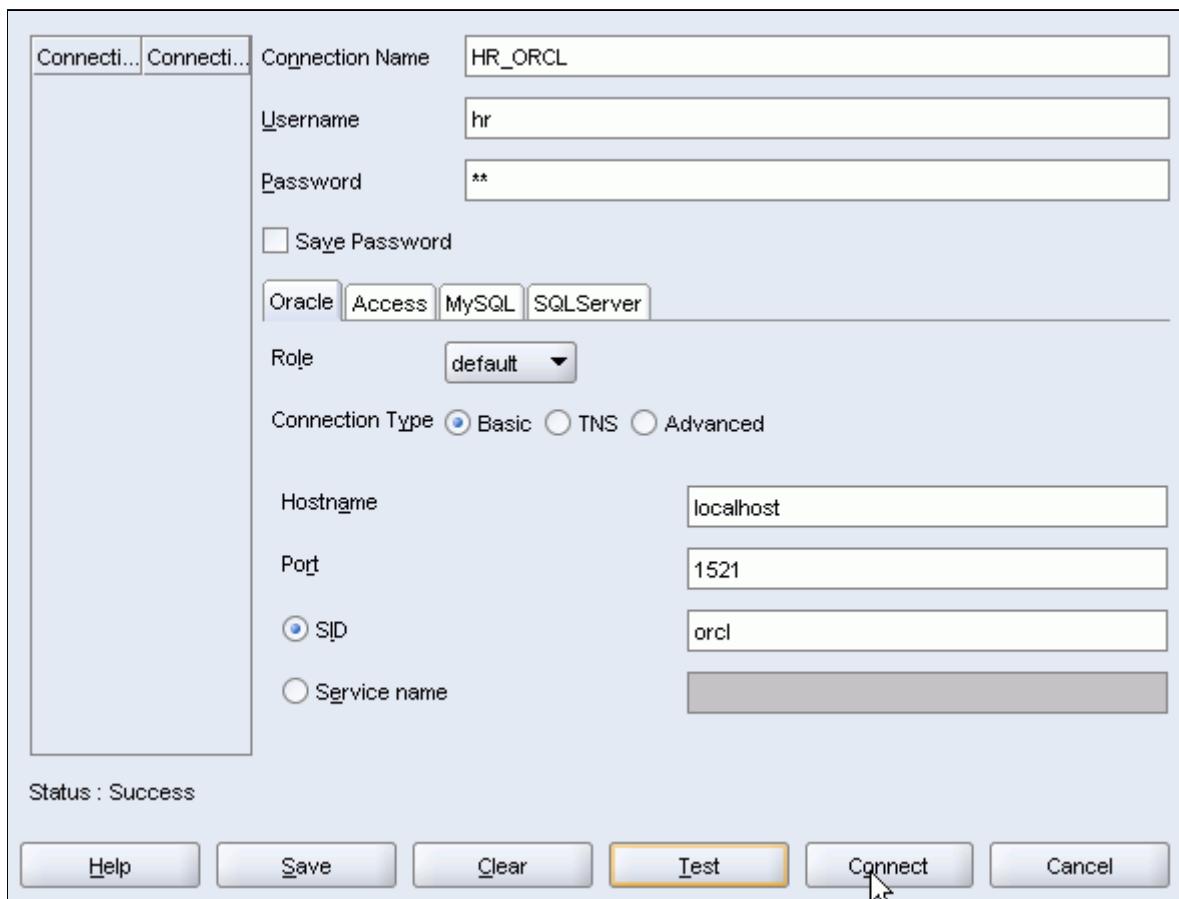
3. Enter **HR_ORCL** for the Connection Name (or any other name that identifies your connection), **HR** for the Username and Password, specify your <hostname> for the Hostname and enter **ORCL** for the SID. Click **Test**.

The screenshot shows the 'New Connection' dialog box. The connection details are as follows:

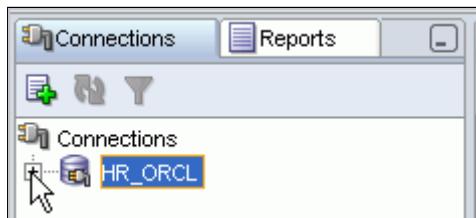
Connection Name	HR_ORCL
Username	hr
Password	**
<input type="checkbox"/> Save Password	
Oracle <input type="radio"/> Access <input type="radio"/> MySQL <input type="radio"/> SQLServer	
Role	default <input type="button" value="▼"/>
Connection Type	<input checked="" type="radio"/> Basic <input type="radio"/> TNS <input type="radio"/> Advanced
Hostname	localhost
Port	1521
<input checked="" type="radio"/> SID	orcl
<input type="radio"/> Service name	

Status :

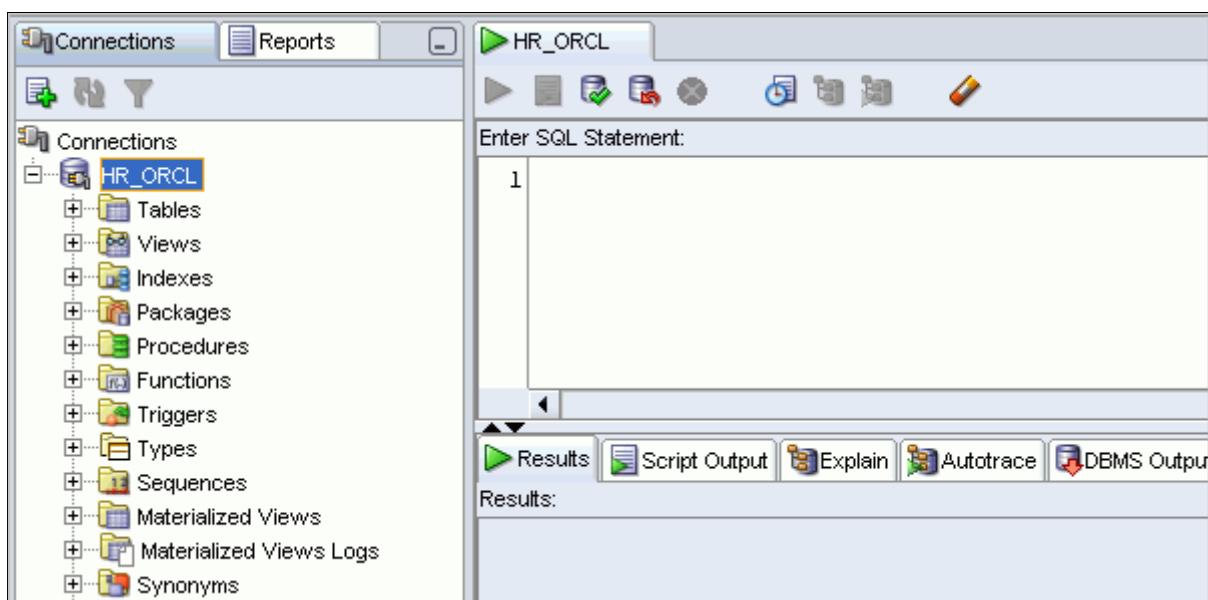
4. The status of the connection was tested successfully. The connection was not saved however. To save the connection, click **Connect**.



5. The connection was saved and you see the database in the list. Expand **HR_ORCL**.



6. When a connection is opened, a SQL Worksheet is opened automatically. The SQL Worksheet allows you to execute SQL against the connection you just created.



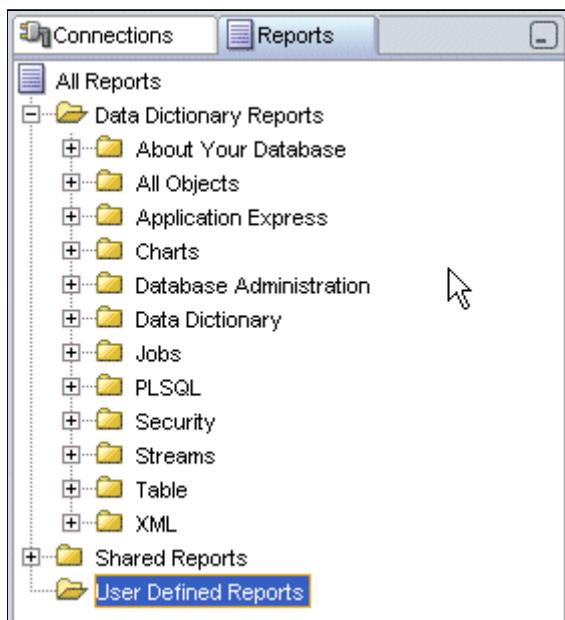
Sharing Reports

Users who want to share user defined reports can add these as an extension to Oracle SQL Developer. To do this perform the following steps:

1. Select the Reports tab.



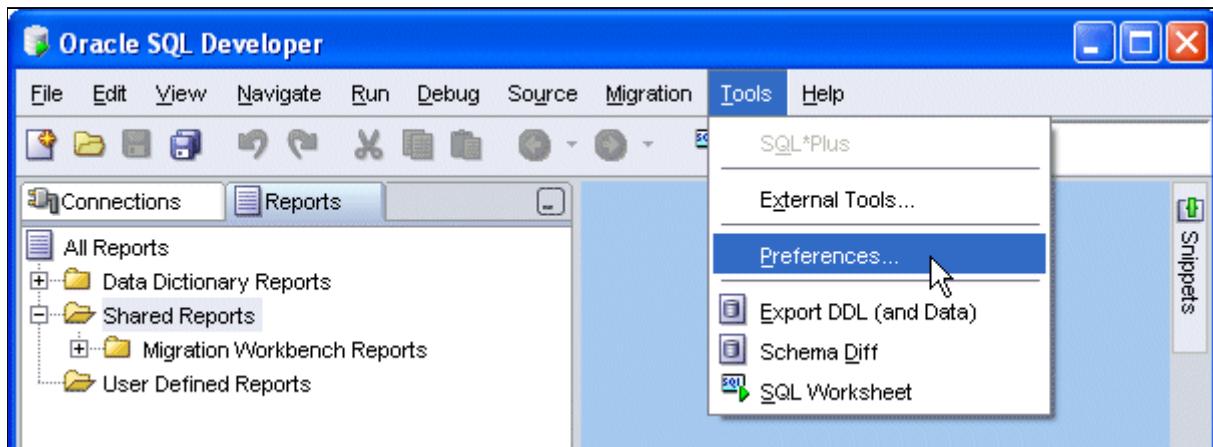
2. Expand each of the nodes. Note that there are a selection of shipped reports under the **Data Dictionary Reports** node.



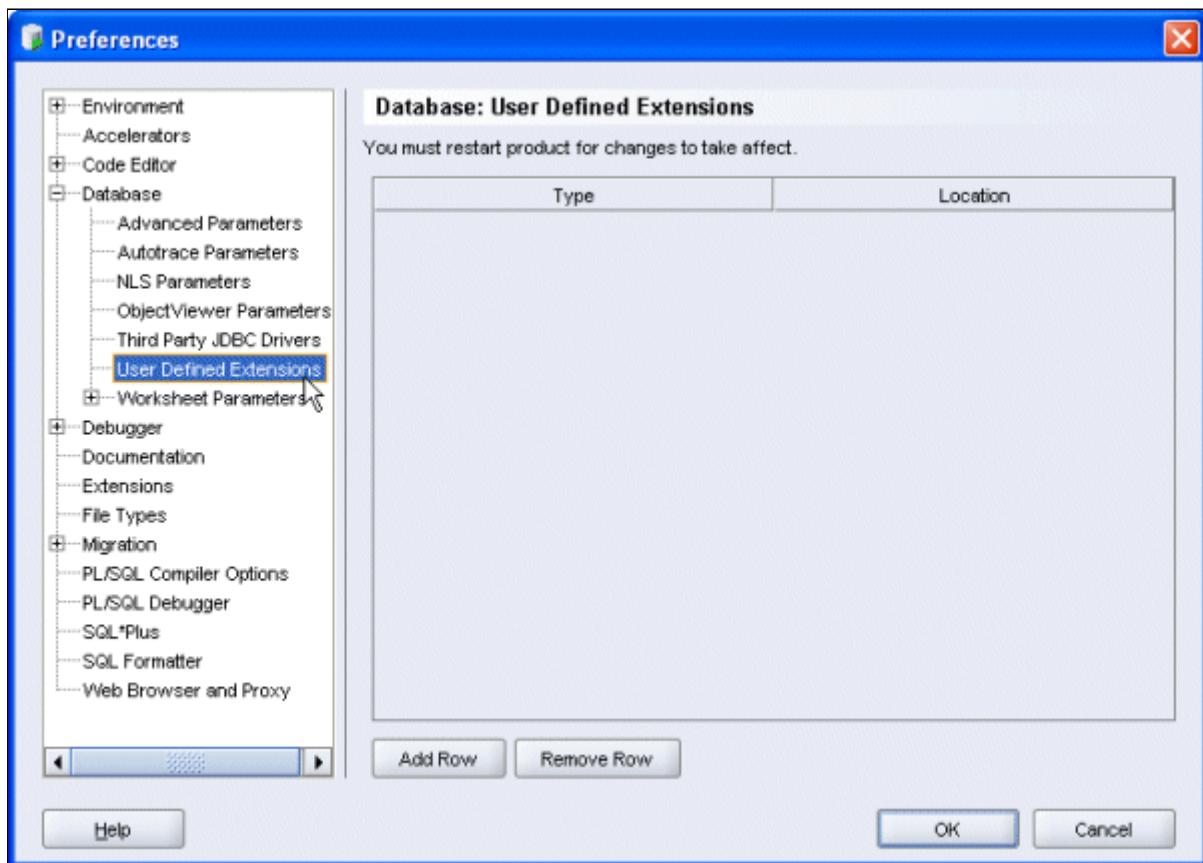
3. There are also shipped Migration Workbench Reports under the **Shared Reports** node. Any reports you create are added under the **User Defined Reports** node.



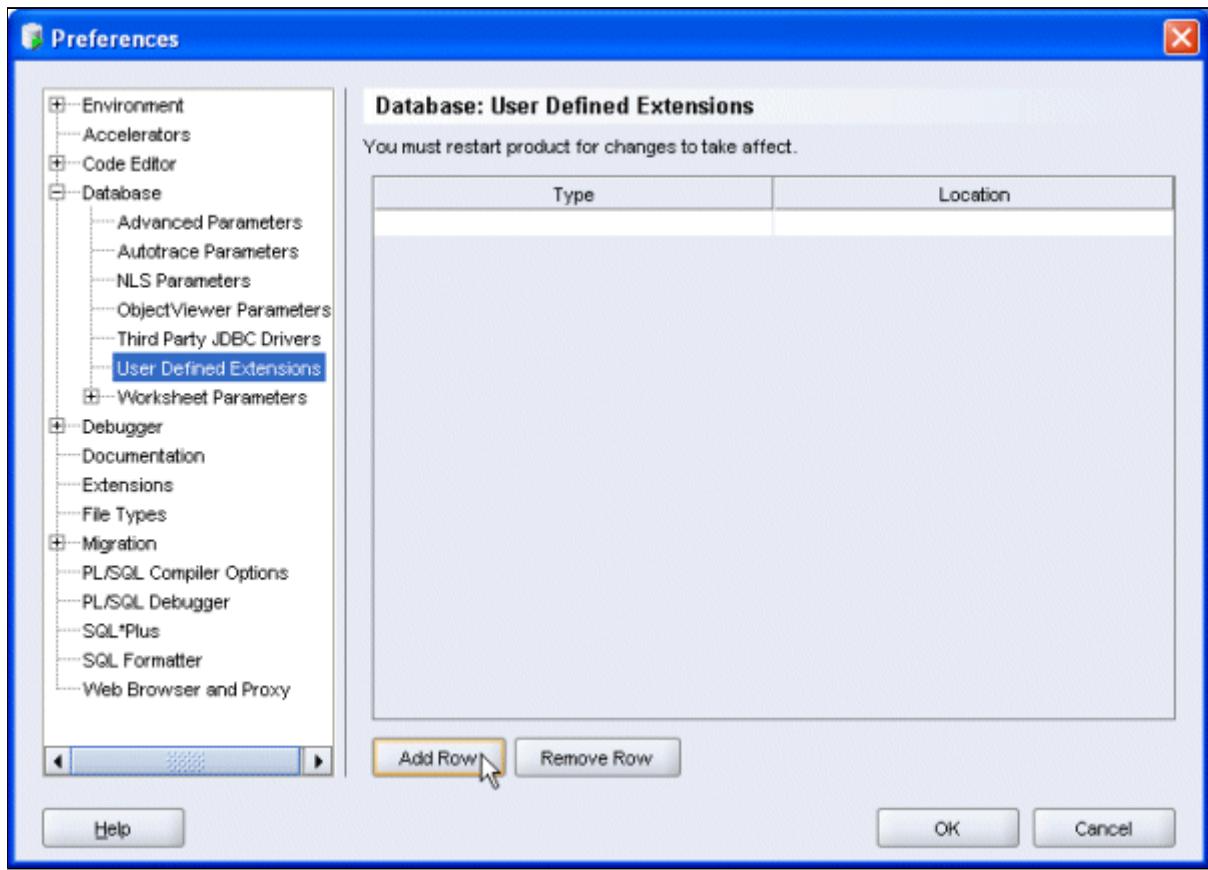
4. To add a shared reports extension, navigate to the **Tools-> Preferences** menu.



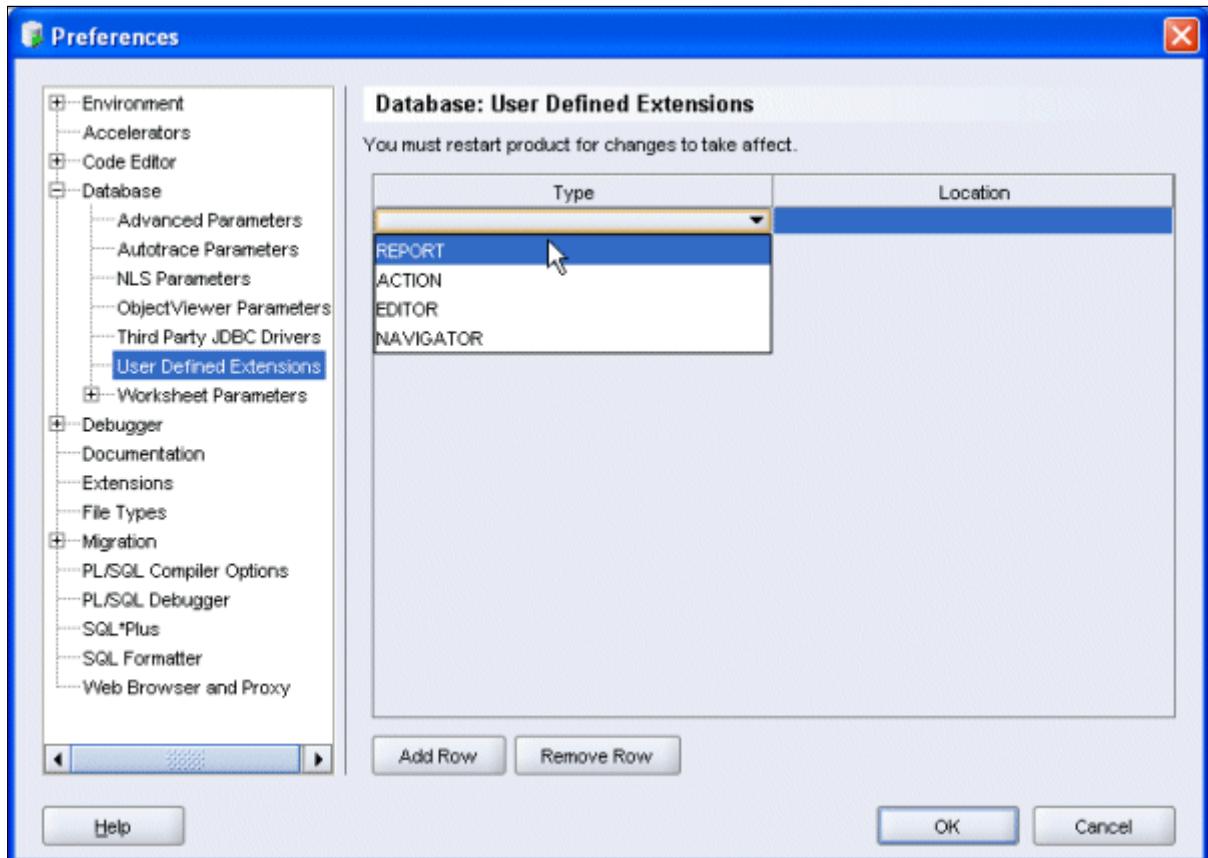
5. Expand the **Database** node and select **User Defined Extensions**.



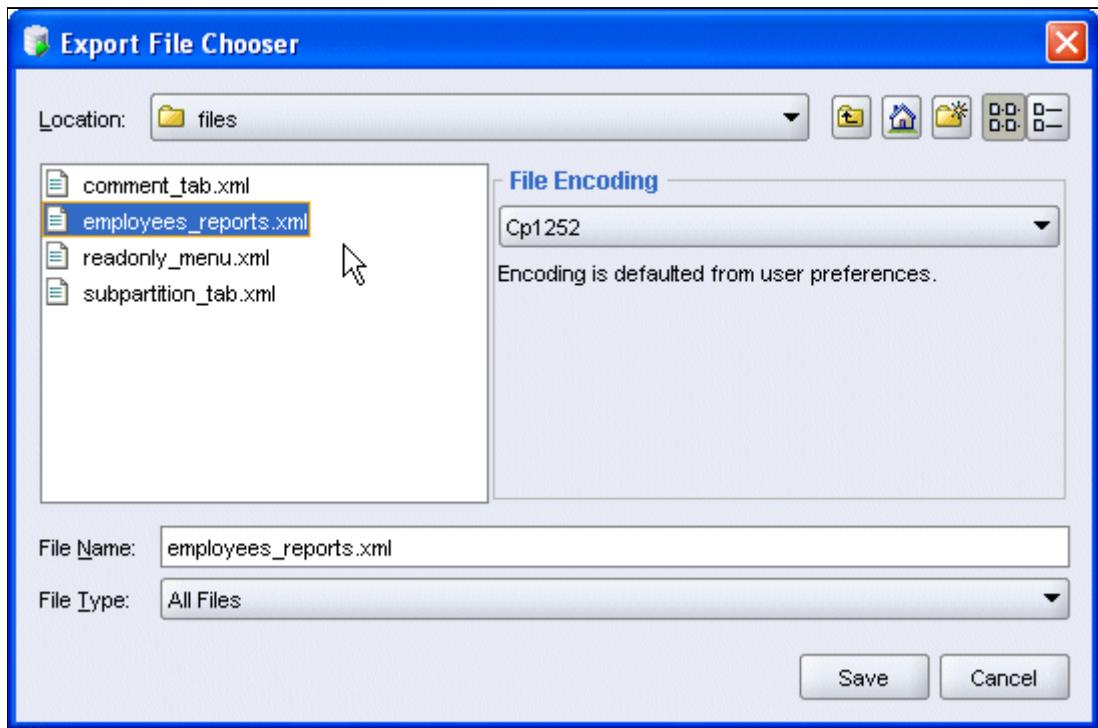
6. Click **Add Row**



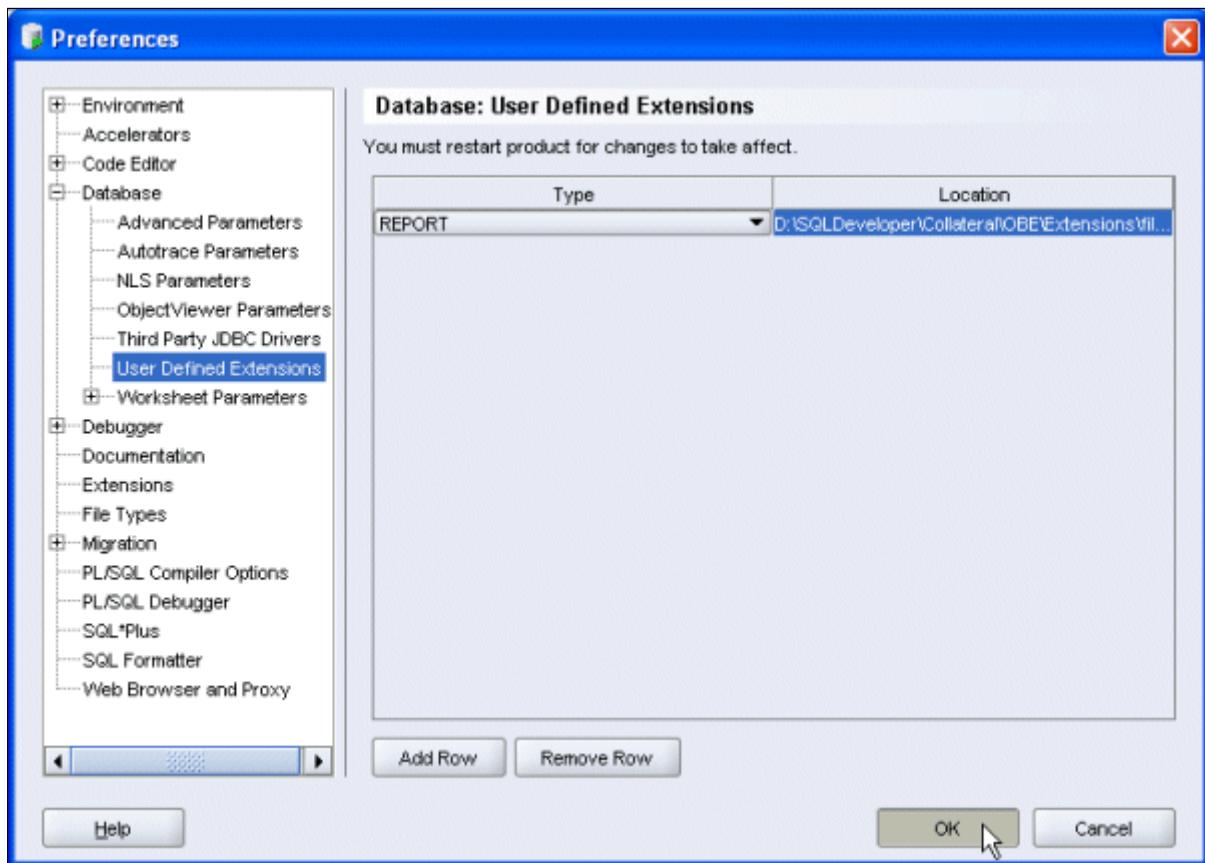
7. Click in the field under **Type** and select **Report**.



8. Click in the field under **Location** and **Browse** to the file location. The files are in the **\SQLDev_HOS\Extensions\files**. Select the **employees_reports.xml** file.

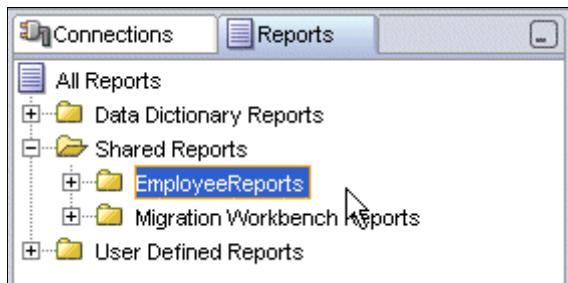


9. Ensure the location is selected, then click **OK**.

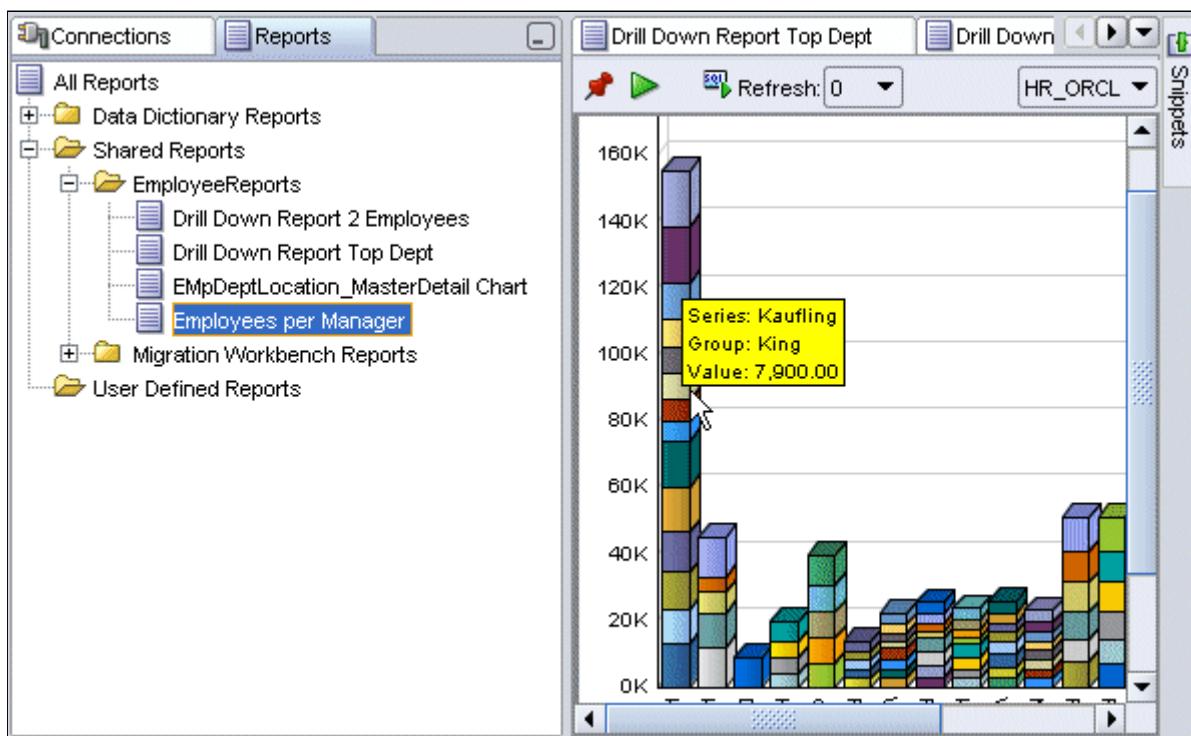


10. Shutdown and restart Oracle SQL Developer for the changes to take effect.

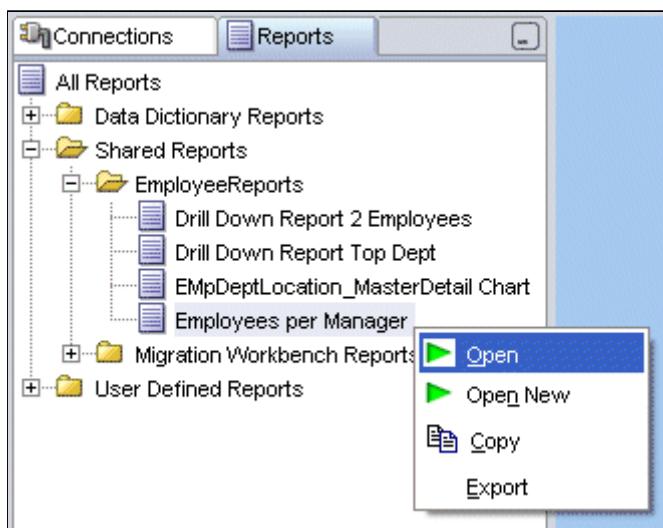
11. Navigate to the Reports tab and select **Shared Reports**.



12. Expand the **EmployeesReports** node and select **Employees per Manager**. Select HR_ORCL from the drop list when prompted and run the report.



13. Right-click on the **Employees per Manager** report. Note that the context menu does not permit you to edit this report. You can copy and recreate it as a User Defined report, but shared reports are not editable.



[Back to Topic List](#)

Adding an Extra Tab to Display Column Comments

Each object in the Navigator has a matching set of definition tabs. These are based on queries against the Data Dictionary. Not all features describing an object are displayed in these tabs. To create a Column Comments tab for a table, perform the following steps:

1. Select the **Connections** tab and expand the **HR_ORCL** connection.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the Navigator with the 'Connections' tab selected. A tree view shows the 'HR_ORCL' connection expanded, revealing nodes for Tables, Views, Indexes, Packages, Procedures, Functions, Triggers, Types, Sequences, Materialized Views, Materialized Views Logs, Synonyms, and Public Synonyms. The right pane shows the 'EMPLOYEES' table details. The 'Columns' tab is active, displaying a grid of columns with their names, data types, and nullability. The columns listed are: EMPLOYEE_ID (NUMBER(6,0)), FIRST_NAME (VARCHAR2(20 BYTE)), LAST_NAME (VARCHAR2(25 BYTE)), EMAIL (VARCHAR2(25 BYTE)), PHONE_NUMBER (VARCHAR2(20 BYTE)), HIRE_DATE (DATE), JOB_ID (VARCHAR2(10 BYTE)), SALARY (NUMBER(8,2)), COMMISSION_PCT (NUMBER(2,2)), MANAGER_ID (NUMBER(6,0)), and DEPARTMENT_ID (NUMBER(4,0)).

2. Expand the **Tables** node and select **EMPLOYEES**.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the Navigator with the 'Connections' tab selected. The 'Tables' node under 'HR_ORCL' is expanded, showing various tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS. The 'EMPLOYEES' table is highlighted with a yellow selection bar. The right pane shows the 'EMPLOYEES' table details, identical to the previous screenshot, with the 'Columns' tab active.

3. The initial tab displayed is the **Columns** tab. Select the **Constraints** tab and view the details displayed.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to HR_ORCL. Under Tables, the EMPLOYEES table is selected. The main panel displays the constraints for the EMPLOYEES table. The 'Constraints' tab is active, showing the following data:

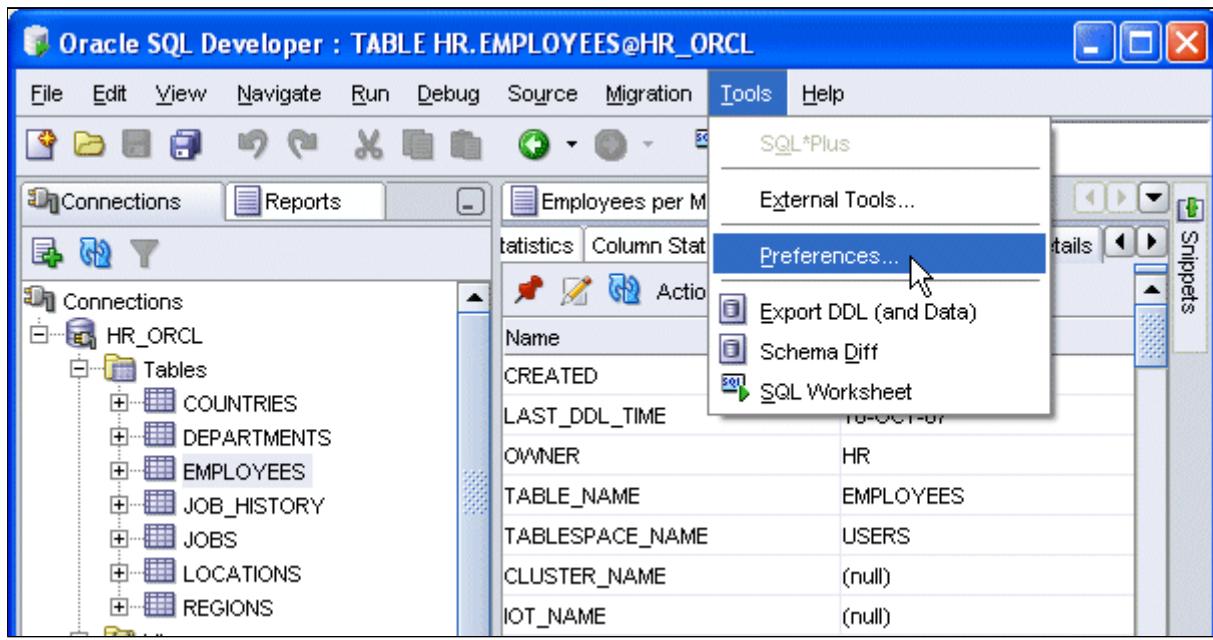
Constraint Name	Constraint Type	Search Condition
EMP_DEPT_FK	Foreign_Key	(null)
EMP_EMAIL_NN	Check	"EMAIL" IS NOT NULL
EMP_EMAIL_UK	Unique	(null)
EMP_EMP_ID_PK	Primary_Key	(null)
EMP_HIRE_DATE_NN	Check	"HIRE_DATE" IS NOT NULL
EMP_JOB_FK	Foreign_Key	(null)
EMP_JOB_NN	Check	"JOB_ID" IS NOT NULL
EMP_LAST_NAME_NN	Check	"LAST_NAME" IS NOT NULL
EMP_MANAGER_FK	Foreign_Key	(null)
EMP_SALARY_MIN	Check	salary > 0

4. Click through the other tabs to see the full details available for a table.

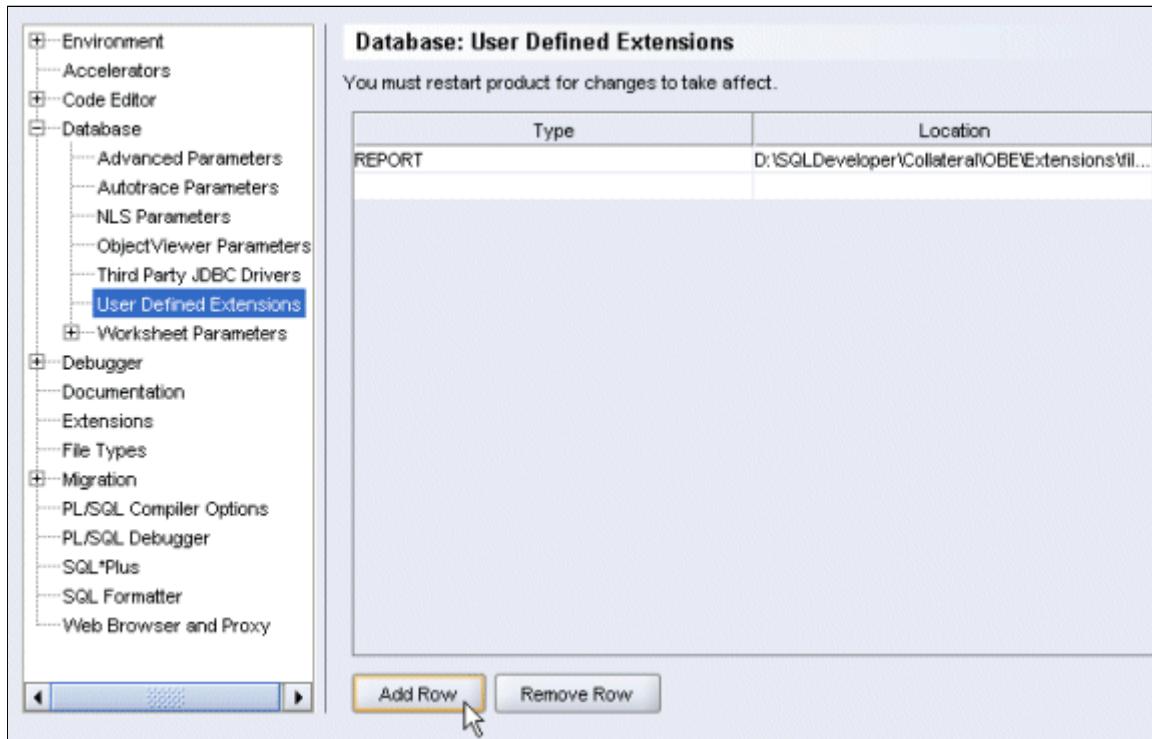
The screenshot shows the Oracle SQL Developer interface. The connections tree and table selection are identical to the previous screenshot. The main panel now displays the table's properties under the 'Details' tab. The table has the following properties:

Name	Value
CREATED	16-OCT-07
LAST_DDL_TIME	16-OCT-07
OWNER	HR
TABLE_NAME	EMPLOYEES
TABLESPACE_NAME	USERS
CLUSTER_NAME	(null)
IOT_NAME	(null)
STATUS	VALID
PCT_FREE	10
PCT_USED	(null)
INI_TRANS	1
MAX_TRANS	255
INITIAL_EXTENT	65536

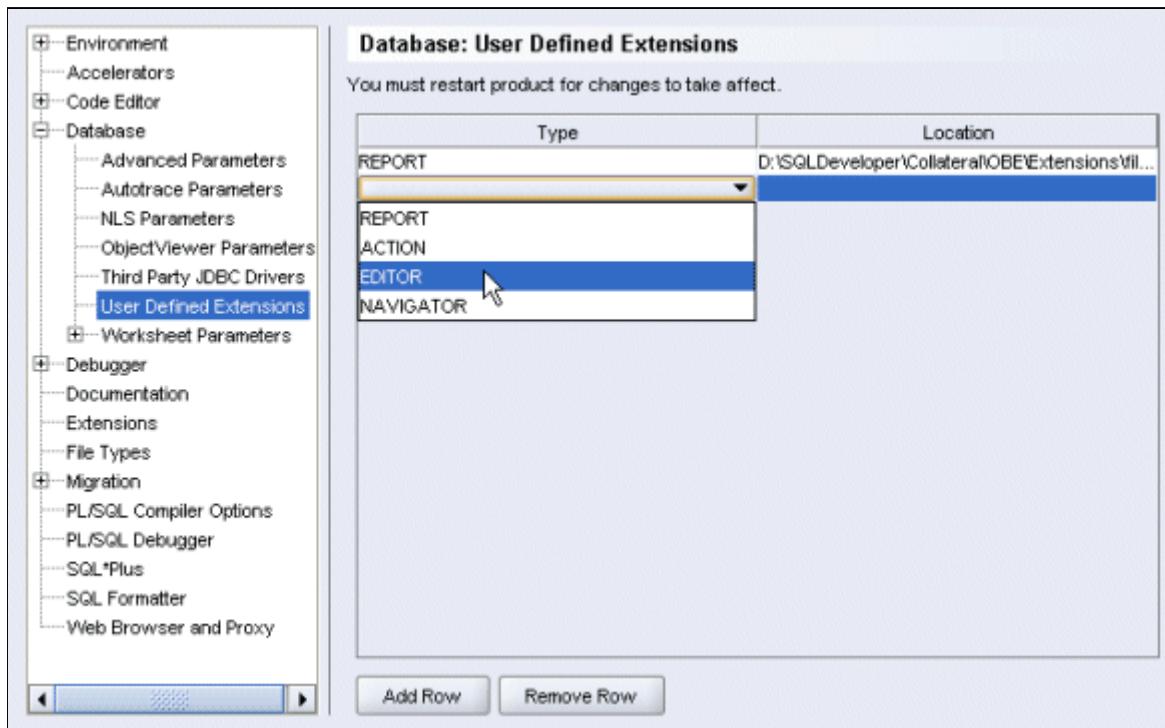
5. Some users would like to display the column comments for a table in a tab too. To do this you need to add in an extension. Select **Tools->Preferences**.



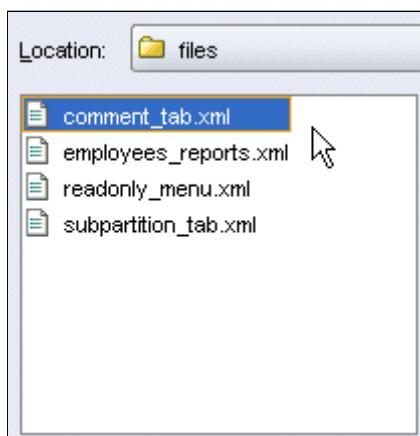
6. As before, select **Database** and **User Defined Extensions**. Click Add Row.



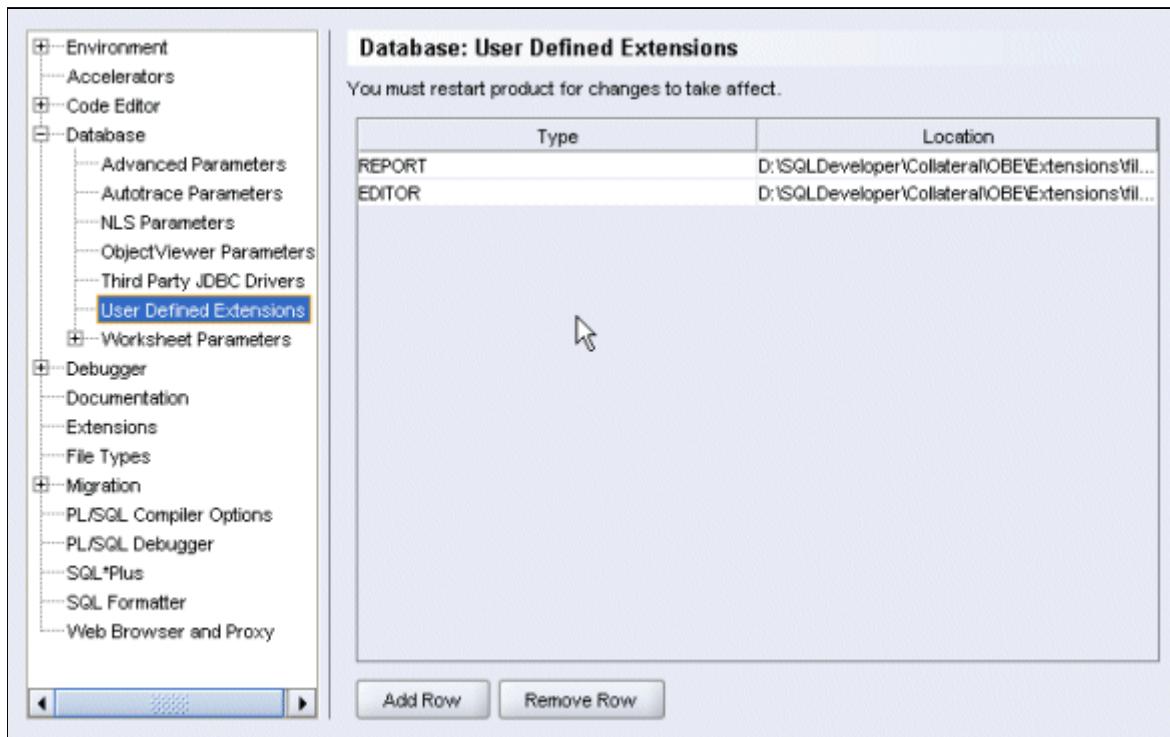
7. Select **Editor** from the drop list for **Type**.



8. Set the **Location** by browsing to the **\SQLDev_HOS\Extensions\files** directory and select the **comment_tab.xml** file.



9. Ensure the Location is stored and click **OK**.



10. Shut down and restart Oracle SQL Developer for your changes to take effect.

11. Select the **HR_ORCL** connection. Expand the tables node and select **EMPLOYEES**.

Column Name	Data Type	Nullable	Default Value
EMPLOYEE_ID	NUMBER(6,0)	No	(null)
FIRST_NAME	VARCHAR2(20 BYTE)	Yes	(null)
LAST_NAME	VARCHAR2(25 BYTE)	No	(null)
EMAIL	VARCHAR2(25 BYTE)	No	(null)
PHONE_NUMBER	VARCHAR2(20 BYTE)	Yes	(null)
HIRE_DATE	DATE	No	(null)
JOB_ID	VARCHAR2(10 BYTE)	No	(null)
SALARY	NUMBER(8,2)	Yes	(null)
COMMISSION_PCT	NUMBER(2,2)	Yes	(null)
MANAGER_ID	NUMBER(6,0)	Yes	(null)
DEPARTMENT_ID	NUMBER(4,0)	Yes	(null)

12. Tab through to the last tab, **Column Comments**. The new tab is added after the other tabs.

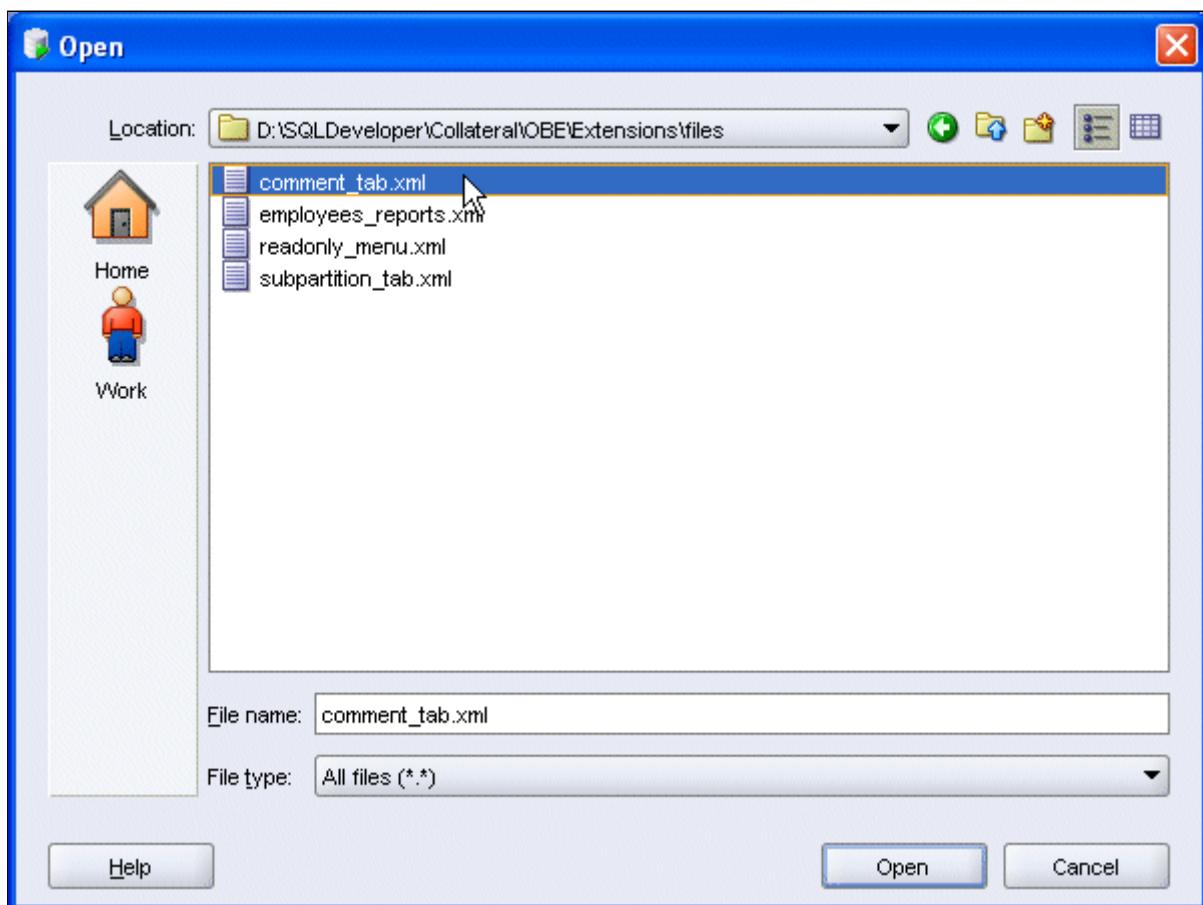
The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database objects under 'Connections' for 'HR_ORCL'. Under 'Tables', several tables are listed: COUNTRIES, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS. The 'EMPLOYEES' table is selected. The main panel shows the 'EMPLOYEES' table structure with columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID, and DEPARTMENT_ID. Each column has a detailed description provided in the 'Comments' section.

[Back to Topic List](#)

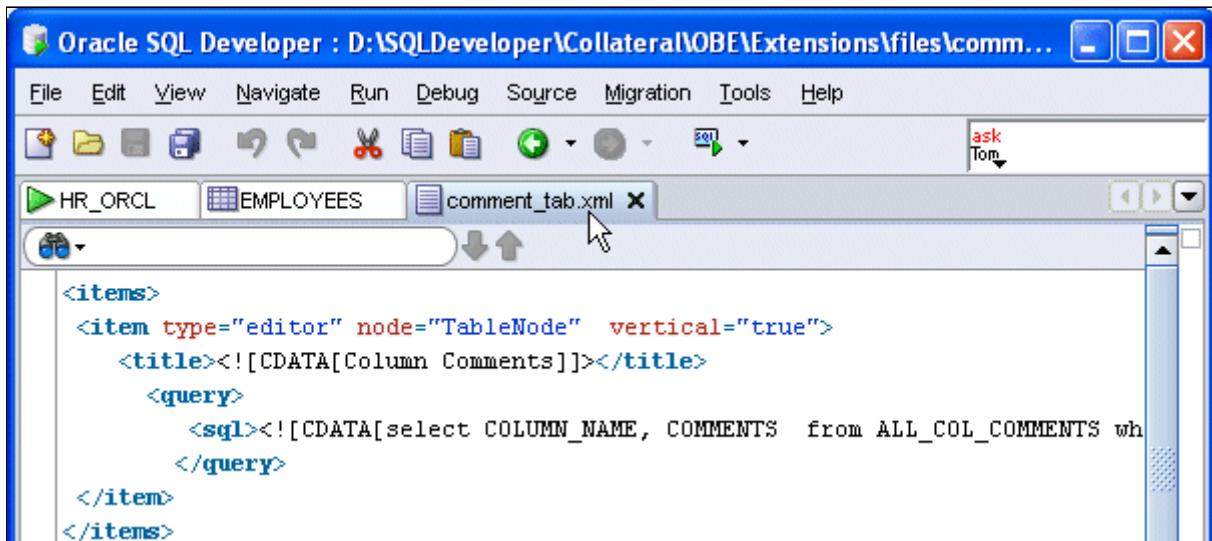
Reviewing the XML File structure

In order to add an XML extension to Oracle SQL Developer, you need to create the XML file. In this section you review the structure of the XML File.

1. In Oracle SQL Developer, select **File ->Open** and browse to the **\SQLDev_HOS\Extensions\files** directory and select the **comment_tab.xml** file.



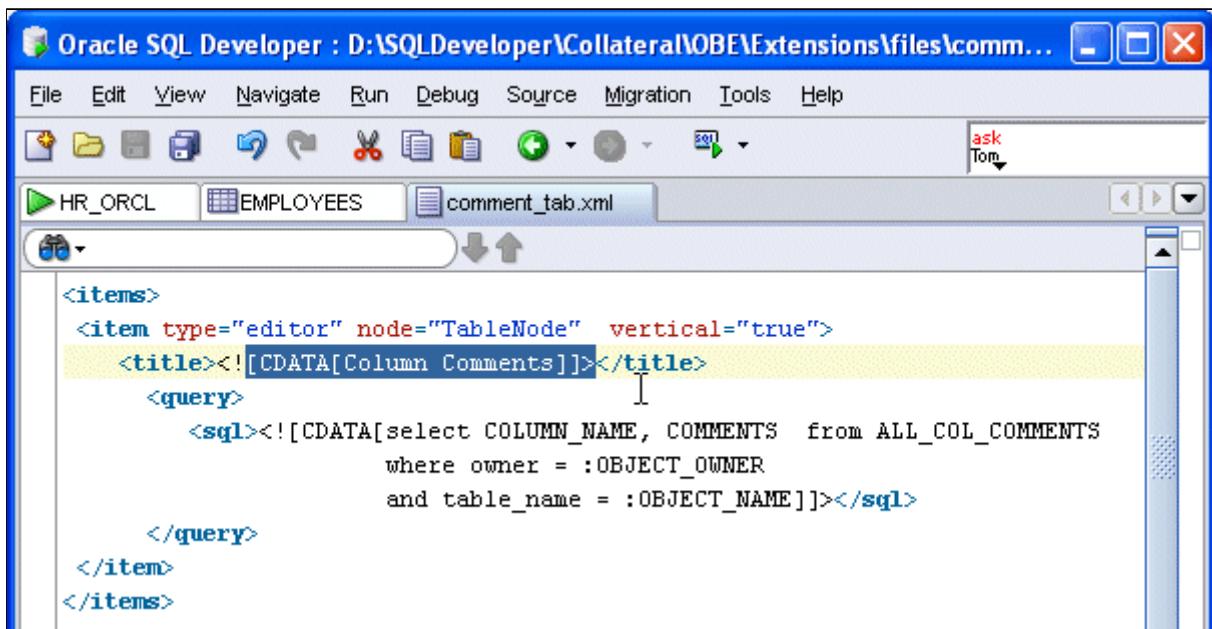
2. Double-click on the tab to maximize the editor.



The screenshot shows the Oracle SQL Developer interface with the title bar "Oracle SQL Developer : D:\SQLDeveloper\Collateral\OBE\Extensions\files\comm...". The menu bar includes File, Edit, View, Navigate, Run, Debug, Source, Migration, Tools, and Help. The toolbar has various icons for file operations like Open, Save, and Run. A search bar at the top right says "ask Tom". Below the toolbar, there are three tabs: HR_ORCL, EMPLOYEES, and comment_tab.xml, with comment_tab.xml currently selected. The main editor area displays the following XML code:

```
<items>
  <item type="editor" node="TreeNode" vertical="true">
    <title><![CDATA[Column Comments]]></title>
    <query>
      <sql><![CDATA[select COLUMN_NAME, COMMENTS  from ALL_COL_COMMENTS wh
      </query>
    </item>
</items>
```

3. The Oracle SQL Developer framework recognizes the XML tags. The pieces of code you are interested in are Title and the SQL Query. The Title here is **Column Comments** and is used to name the tab.



The screenshot shows the Oracle SQL Developer interface with the title bar "Oracle SQL Developer : D:\SQLDeveloper\Collateral\OBE\Extensions\files\comm...". The menu bar, toolbar, and search bar are identical to the previous screenshot. The main editor area displays the same XML code as before, but the title tag is highlighted with a yellow background:

```
<items>
  <item type="editor" node="TreeNode" vertical="true">
    <title><![CDATA[Column Comments]]></title>
    <query>
      <sql><![CDATA[select COLUMN_NAME, COMMENTS  from ALL_COL_COMMENTS wh
      </query>
    </item>
</items>
```

4. The SQL Query is querying the Data Dictionary for the column comments for the **OWNER** and the **OBJECT_NAME**, in this example the EMPLOYEES Table for schema HR.

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : D:\SQLDeveloper\Collateral\OBE\Extensions\files\subpa...". The menu bar includes File, Edit, View, Navigate, Run, Debug, Source, Migration, Tools, and Help. The toolbar has various icons for file operations like Open, Save, and Print. The main window shows a code editor with the following XML code:

```

<items>
  <item type="editor" node="TreeNode" vertical="true">
    <title><![CDATA[SubPartitions]]></title>
    <query>
      <sql><![CDATA[Select PARTITION_NAME, SUBPARTITION_NAME, HIGH_VALUE, HIG
from ALL_TAB_SUBPARTITIONS where table_owner = :OBJECT_OWNER and table_name = :TABLE_NAME]]></sql>
    </query>
  </item>
</items>

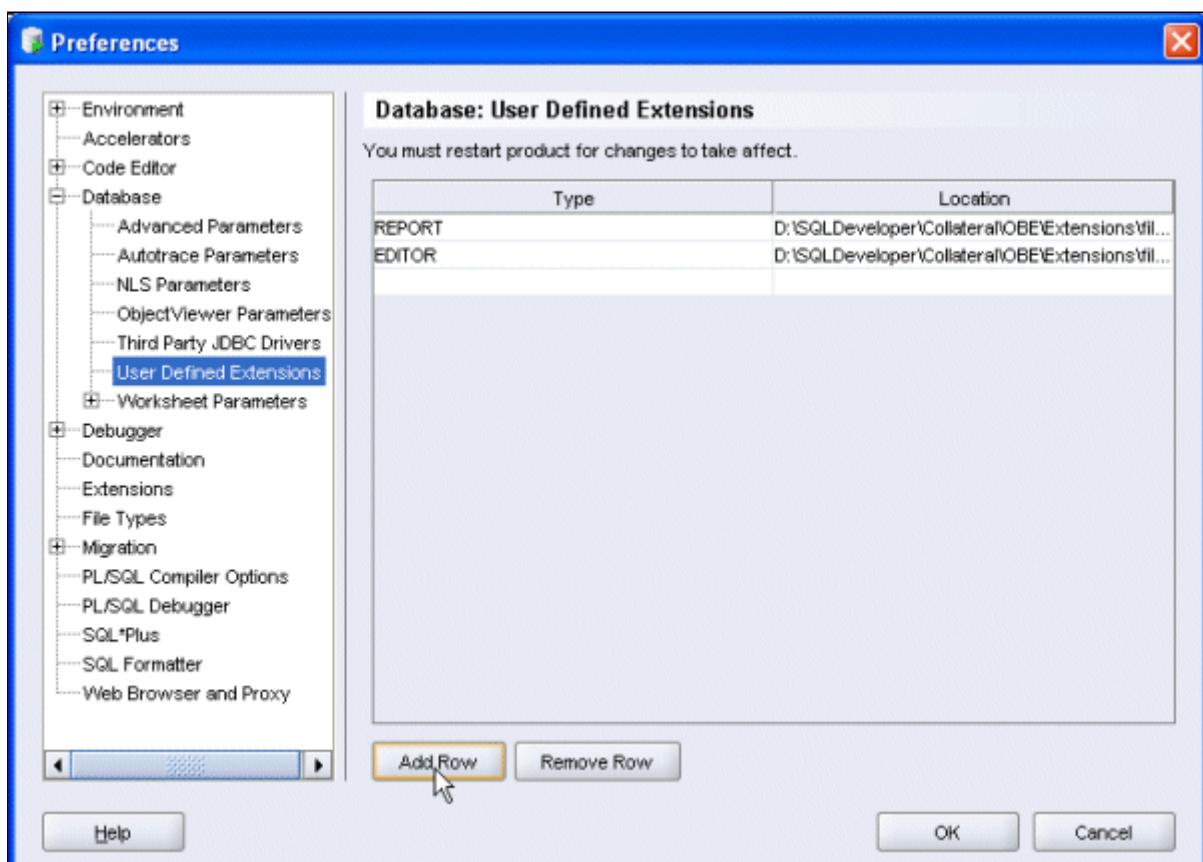
```

[Back to Topic List](#)

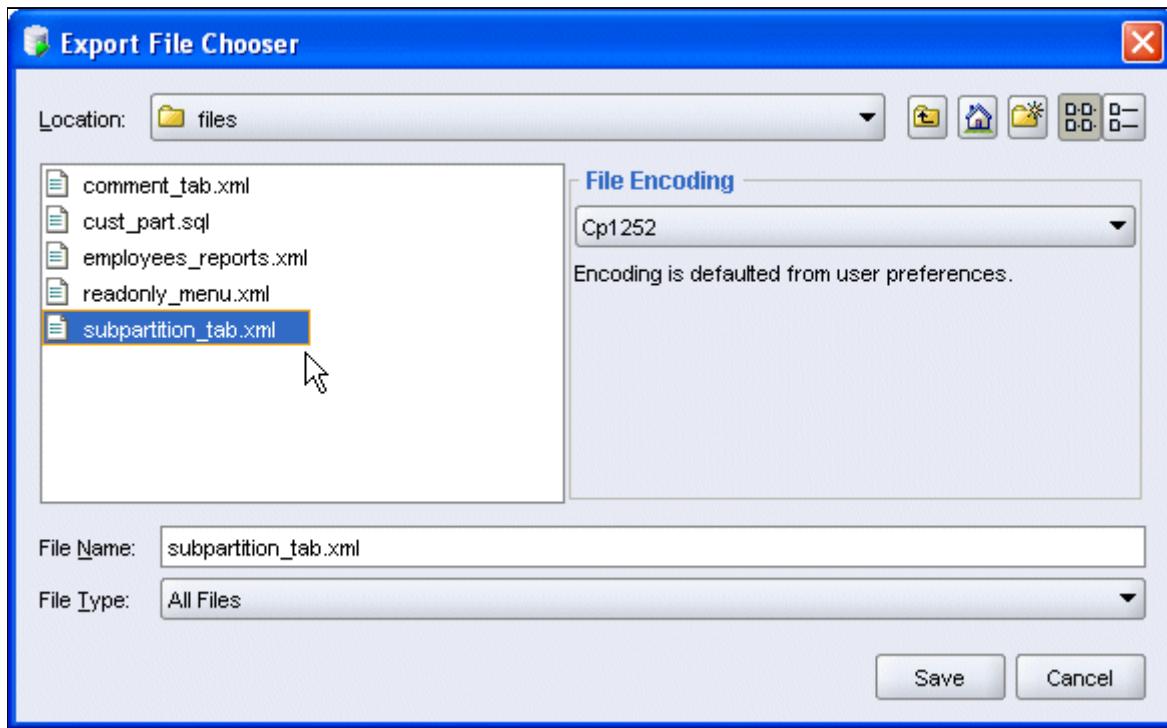
Adding a Tab to Describe Sub-Partitions

This example is the same as the one just completed above. In this instance you are going to add a tab to describe sub partitions. You create a table with partitions and sub partitions to review the details. To add additional tabs to the Table definitions, perform the following steps:

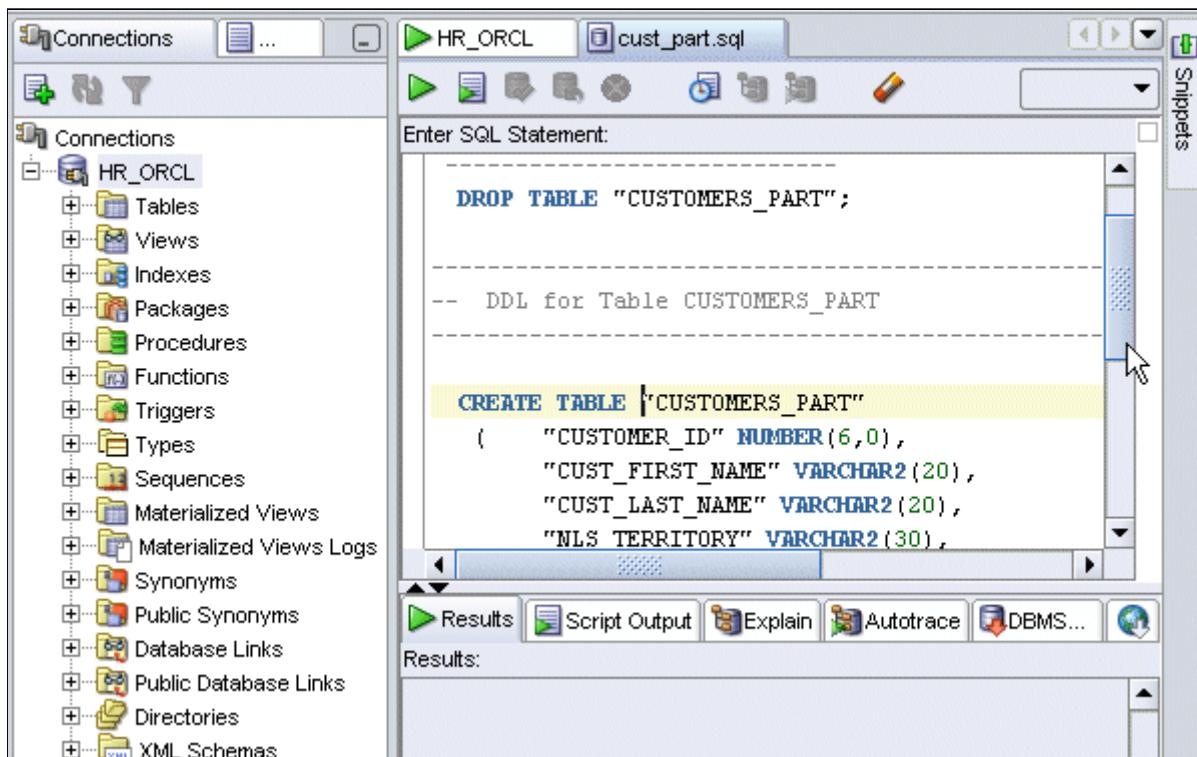
1. Select **Tools -> Preferences** to invoke the dialog. Expand the **Database** node and select **User Defined Extensions**. Click **Add Row**.



2. The Extension Type is Editor. Browse to the **\SQLDev_HOS\Extensions\files** directory and select the **subpartition_tab.xml** file.



3. Shutdown and restart Oracle SQL Developer for your changes to take effect.
4. Expand the HR_ORCL connection. Select File ->Open and browse to the the \SQLDev_HOS\Extensions\files directory. Select the **cust_part.sql** file.



5. Review the file. This script creates a partitioned table, **CUSTOMERS_PART**. Select the **HR_ORCL** connection from the drop list on the right of the SQL Worksheet.

The screenshot shows the Oracle SQL Developer interface. On the left is the Connections Navigator, which lists the HR_ORCL connection with its various schema objects like Tables, Views, Indexes, etc. The main area is the SQL Editor, titled 'cust_part.sql'. It contains the following SQL code:

```
Enter SQL Statement:  
HR_ORCL  
---  
DROP TABLE "CUSTOMERS_PART";  
---  
-- DDL for Table CUSTOMERS_PART  
---  
CREATE TABLE "CUSTOMERS_PART"  
(  
    "CUSTOMER_ID" NUMBER(6,0),  
    "CUST_FIRST_NAME" VARCHAR2(20),  
    "CUST_LAST_NAME" VARCHAR2(20),  
    "NLS_TERRITORY" VARCHAR2(30),  
);
```

The 'Results' tab is selected at the bottom.

6. Click the **Run Script (F5)** icon.

The screenshot shows the Oracle SQL Developer interface again. The Connections Navigator is the same. The SQL Editor has the 'Run Script (F5)' icon highlighted in the toolbar. The SQL code is identical to the previous screenshot:

```
Enter SQL Statement:  
Run Script (F5)  
---  
DROP TABLE "CUSTOMERS_PART";  
---  
-- DDL for Table CUSTOMERS_PART  
---  
CREATE TABLE "CUSTOMERS_PART"  
(  
    "CUSTOMER_ID" NUMBER(6,0),  
    "CUST_FIRST_NAME" VARCHAR2(20),  
    "CUST_LAST_NAME" VARCHAR2(20),  
    "NLS_TERRITORY" VARCHAR2(30),  
);
```

The 'Results' tab is selected at the bottom.

7. Expand the Tables node in the Connections Navigator and select the **CUSTOMERS_PART** table.

NOTE: If the tables node is already expanded, you need to click Refresh, to see the new table.

8. Select the Partitions Tab.

9. The new **SubPartitions** tab is added at the end of all the tabs. Navigate to the end of the tabs and review the new details.

10. If you have time, consider how you might add or remove columns from this SubPartitions tab.

HINT: The file is **\SQLDev_HOS\Extensions\files\subpartition_tab.xml**

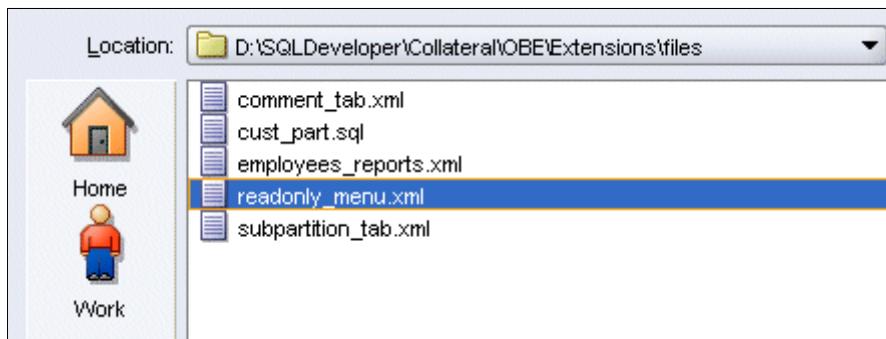
[Back to Topic List](#)

Adding a Context Menu

In the above exercises, you added tabs to the table definition, using XML files. You can add context menus using a similar approach. Context menus can be added at any level to the existing context menu structure. In this exercise you add a high-level context menu to set a transaction to READ ONLY.

To add a new context menu perform the following steps:

1. Select **File->Open**. Navigate to the **\SQLDev_HOS\Extensions\files** directory and select the **readonly_menu.xml** file.



2. Double-click on the **readonly_menu.xml** tab to maximize the editor.

A screenshot of the SQL Developer XML editor window. The title bar shows the connection 'HR_ORCL' and the tab 'readonly_menu.xml'. The editor displays the XML code for setting connection read-only status. The code includes sections for 'Set R0' and 'Set RW', each with a confirmation dialog and SQL command to change the transaction mode.

```
<?xml version="1.0" encoding="UTF-8"?>
<items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="dialogs.xsd">
    <item type="CONNECTION" reload="true">
        <title>Set R0</title>
        <prompt type="confirm">
            <label>Confirm Setting Connection to Read Only.</label>
        </prompt>
        <sql>
            <![CDATA[begin dbms_transaction.read_only; end;]]>
        </sql><help>Issues dbms_transaction.read_only; </help>
        <confirmation>
            <title>Confirmation</title>
            <prompt>Success</prompt>
        </confirmation>
    </item>
    <item type="CONNECTION" reload="true">
        <title>Set RW</title>
        <prompt type="confirm">
            <label>Confirm Setting Connection to Read Write.</label>
        </prompt>
    </item>
</items>
```

3. This XML file is more involved. Notice that in this case there are two sections; one for **Set Read Only** and the other for **Set Read/Write**. The image below highlights one of the sections between the `<item>` tags..

The screenshot shows the Oracle SQL Developer interface with the Database Editor open. The title bar reads "HR_ORCL CUSTOMERS_PART readonly_menu.xml". The code editor displays XML configuration for a menu item. The XML includes tags for item type ("CONNECTION"), title ("Set Read Only"), prompt type ("confirm"), label ("Confirm Setting Connection to Read Only."), SQL code to set the connection to read-only, help text ("Issues dbms_transaction.read_only;"), confirmation title ("Confirmation"), confirmation prompt ("Success"), and another item for setting read/write.

```
<?xml version="1.0" encoding="UTF-8"?>
<items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="dialogs.xsd">
<item type="CONNECTION" reload="true">
<title>Set Read Only</title>
<prompt type="confirm">
<label>Confirm Setting Connection to Read Only.</label>
</prompt>
<sql>
<![CDATA[begin dbms_transaction.read_only; end;]]>
</sql>
<help>Issues dbms_transaction.read_only; </help>
<confirmation>
<title>Confirmation</title>
<prompt>Success</prompt>
</confirmation>
</item>
<item type="CONNECTION" reload="true">
<title>Set Read/Write</title>
<prompt type="confirm">
<label>Confirm Setting Connection to Read Write.</label>

```

4. Notice the **<prompt>** and **<confirmation>** tags.

The screenshot shows the Oracle SQL Developer interface with the Database Editor open. The title bar reads "HR_ORCL CUSTOMERS_PART readonly_menu.xml". The code editor displays the same XML configuration as the previous screenshot, but with a cursor highlighting the entire **<confirmation>** block. This block contains the confirmation title ("Confirmation") and confirmation prompt ("Success").

```
<?xml version="1.0" encoding="UTF-8"?>
<items xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="dialogs.xsd">
<item type="CONNECTION" reload="true">
<title>Set Read Only</title>
<prompt type="confirm">
<label>Confirm Setting Connection to Read Only.</label>
</prompt>
<sql>
<![CDATA[begin dbms_transaction.read_only; end;]]>
</sql>
<help>Issues dbms_transaction.read_only; </help>
<confirmation>
<title>Confirmation</title>
<prompt>Success</prompt>
</confirmation>
</item>
<item type="CONNECTION" reload="true">
<title>Set Read/Write</title>
<prompt type="confirm">
<label>Confirm Setting Connection to Read Write.</label>

```

5. To add the extension to Oracle SQL Developer, select **Tools->Preferences**. Expand the **Database** node and select **User Defined Extensions**. Click **Add Row**.

Database: User Defined Extensions

You must restart product for changes to take affect.

Type	Location
REPORT	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...
EDITOR	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...
EDITOR	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...

Add Row **Remove Row**

6. Select **ACTION** from the Extension Type drop list.

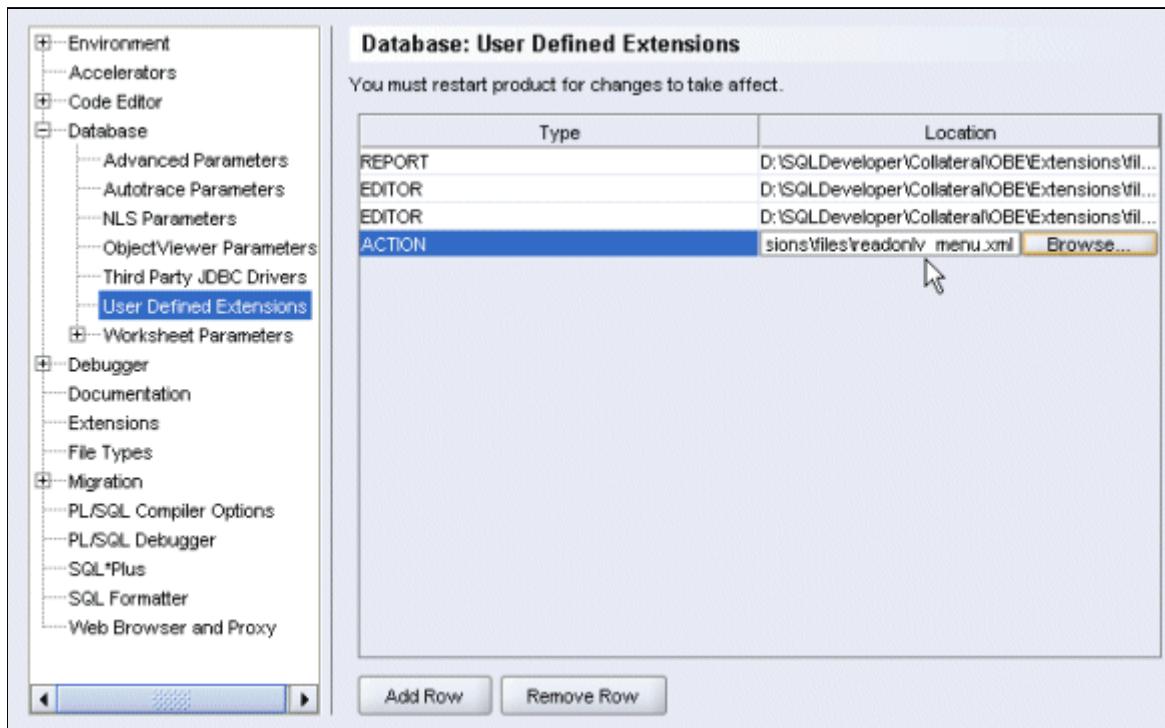
Database: User Defined Extensions

You must restart product for changes to take affect.

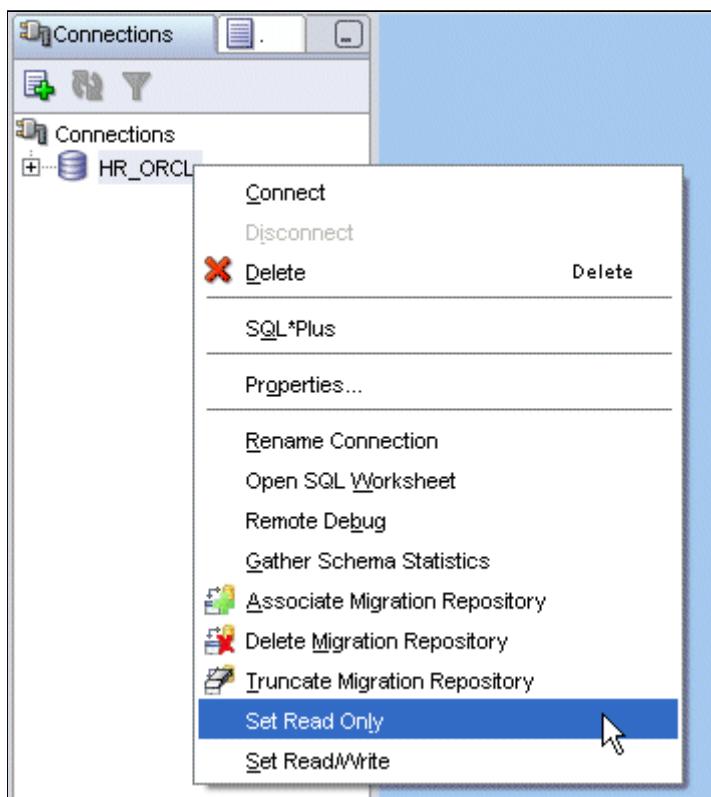
Type	Location
REPORT	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...
EDITOR	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...
EDITOR	D:\SQLDeveloper\Collateral\OBIE\Extensions\fil...
ACTION	
REPORT	
ACTION	
EDITOR	
NAVIGATOR	

Add Row **Remove Row**

7. Select the Location field and browse to the **\SQLDev_HOS\Extensions\files** directory and select the **readonly_menu.xml** file. Ensure the file location is selected and click OK.

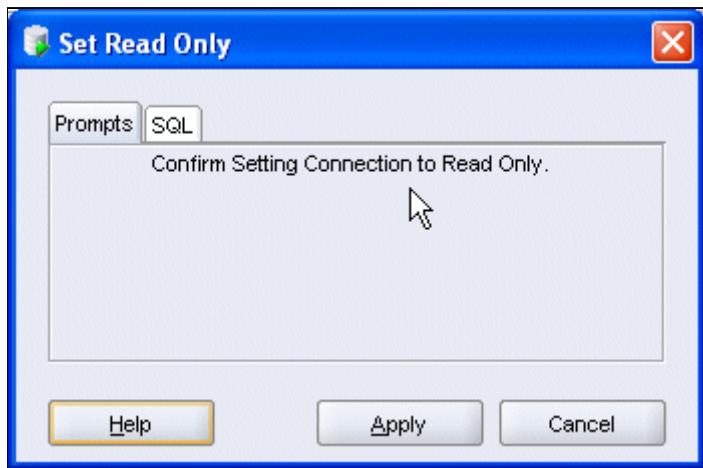


8. Shutdown and restart Oracle SQL Developer for your changes to take effect.
9. Select the **HR_ORCL** connection and right-click to invoke the context menu. Scroll to the bottom and select **Set Read Only**.

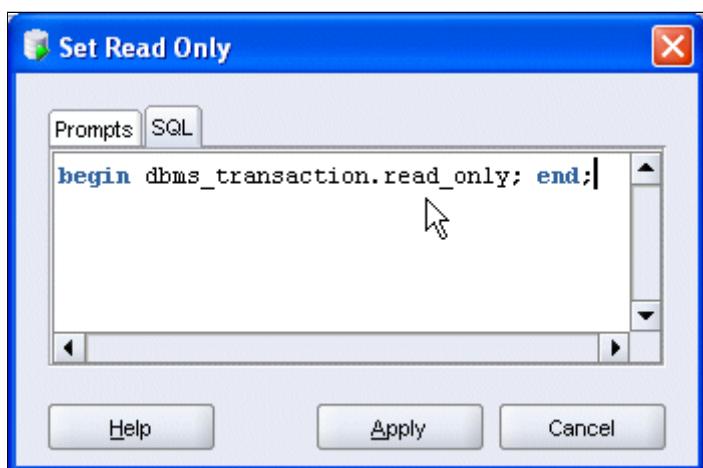


10. The title "**Set Read Only**" is displayed in the dialog header. Notice the Prompt. The text is from the

```
<prompt type="confirm">
<label>Confirm Setting Connection to Read Only.</label>
</prompt>
```



11. Select the **SQL** tab. Find the equivalent SQL code in the XML file. Click **Apply**. The final **Confirmation** dialog is also driven by the XML file.



12. Expand the Tables node for the connection **HR ORCL** and select **EMPLOYEES**. Select the data tab.

The screenshot shows the Oracle SQL Developer interface. On the left is a tree view of database connections, with 'HR_ORCL' expanded to show tables like COUNTRIES, CUSTOMERS_PA, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS, as well as Views, Indexes, Packages, Procedures, Functions, Triggers, Types, Sequences, Materialized Views, and Materialized Log Views. The central area is titled 'Data' and displays the 'EMPLOYEES' table. The table has columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, and EMAIL. The data shows 16 rows of employee information. The right side of the screen shows various toolbars and a status bar.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	Steven	King	SKING
2	Neena	Kochhar	NKOCHH...
3	Lex	De Haan	LDEHAAN
4	Alexander	Hunold	AHUNOLD
5	Bruce	Ernst	BERNST
6	David	Austin	DAUSTIN
7	Valli	Pataballa	VPATAB...
8	Diana	Lorentz	DLORENTZ
9	Nancy	Greenberg	NGREENBE
10	Daniel	Faviet	DFAVIET
11	John	Chen	JCHEN
12	Ismael	Sciarra	ISCIARRA
13	Jose Manuel	Urman	JMURMAN
14	Luis	Popp	LPOPP
15	Den	Raphaely	DRAPHE...
16	Alexander	Khoa	AKHOOO

13. Modify the **FIRST_NAME** for one of the records.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to HR_ORCL. Under Tables, the EMPLOYEES table is selected. The main area displays the data for the EMPLOYEES table:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	100 Steven	King	SKING
2	101 Neena	Kochhar	NKOCHH...
3	102 Lex	De Haan	LDEHAAN...
4	103 Alexander	Hunold	AHUNOLD...
5	104 Bruce	Ernst	BERNST...
6	105 David	Austin	DAUSTIN...
7	106 Valli	Pataballa	VPATAB...
8	107 Diana	Lorentz	DLORENTZ...
9	108 Nancy	Greenberg	NGREENBE...
10	109 Daniel	Faviet	DFAVIET...
11	110 John	Chen	JCHEN...
12	111 Ismael	Sciarra	ISCIARRA...
13	112 Jose Manuel	Urman	JMURMAN...
14	113 Luis	Popp	LPOPP...
15	114 Den	Raphaely	DRAPHE...
16	115 Alexander	Khoo	AKHOO...

14. Select the **Commit** icon to commit the changes.

The screenshot shows the Oracle SQL Developer interface, identical to the previous one, but with a cursor pointing at the Commit icon (a green checkmark) in the toolbar above the data grid.

The data in the EMPLOYEES table remains the same as in the previous screenshot:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	100 Steven	King	SKING
2	101 Neena	Kochhar	NKOCHH...
3	102 Lex	De Haan	LDEHAAN...
4	103 Alexander	Hunold	AHUNOLD...
5	104 Bruce	Ernst	BERNST...
6	105 David	Austin	DAUSTIN...
7	106 Valli	Pataballa	VPATAB...
8	107 Diana	Lorentz	DLORENTZ...
9	108 Nancy	Greenberg	NGREENBE...
10	109 Daniel	Faviet	DFAVIET...
11	110 John	Chen	JCHEN...
12	111 Ismael	Sciarra	ISCIARRA...
13	112 Jose Manuel	Urman	JMURMAN...
14	113 Luis	Popp	LPOPP...
15	114 Den	Raphaely	DRAPHE...
16	115 Alexander	Khoo	AKHOO...

15. You are not able to commit the records due to the **READ ONLY** transaction.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections panel shows a connection to 'HR_ORCL'. Under 'Tables', the 'EMPLOYEES' table is selected. The main area displays the 'EMPLOYEES' table data with 16 rows. The columns are: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, and EMAIL. The data includes rows for Steven King, Neena Kochhar, Lex De Haan, Alexander Hunold, Bruce Ernst, David Austin, Valli Pataballa, Diana Lorentz, Nancy Greenberg, Daniel Faviet, John Chen, Ismael Sciarra, Jose Manuel Urman, Luis Popp, Den Raphaely, and Alexander Khoo.

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	100	Steven	King	SKING
2	101	Neena	Kochhar	NKOCHH...
3	102	Lex	De Haan	LDEHAAN
4	103	Alexander	Hunold	AHUNOLD
5	104	Bruce	Ernst	BERNST
6	105	David	Austin	DAUSTIN
7	106	Valli	Pataballa	VPATAB...
8	107	Diana	Lorentz	DLORENTZ
9	108	Nancy	Greenberg	NGREENBE
10	109	Daniel	Faviet	DFAVIET
11	110	John	Chen	JCHEN
12	111	Ismael	Sciarra	ISCIARRA
13	112	Jose Manuel	Urman	JMURMAN
14	113	Luis	Popp	LPOPP
15	114	Den	Raphaely	DRAPHE...
16	115	Alexander	Khoo	AKHOO

[Back to Topic List](#)

Summary

In this tutorial, you learned how to:

- Creating a Database Connection
- Sharing Reports
- Adding an Extra Tab to Display Column Comments
- Reviewing the XML File structure
- Adding the Tab to describe Subpartitions
- Adding a Context Menu

[Back to Topic List](#)

Move your mouse over this icon to hide all screenshots.

Migrating a Microsoft Access Database to Oracle

Purpose

This tutorial describes how to migrate a Microsoft Access Database to Oracle.

Time to Complete

Approximately 30 minutes

Topics

This tutorial covers the following topics:

- [Overview](#)
- [Prerequisites](#)
- [Creating the OMWBREP User](#)
- [Creating the Migration Repository](#)
- [Capturing the Microsoft Access Exported XML](#)
- [Converting to the Oracle Model](#)
- [Generating and Executing the Script to Create the Oracle Database Objects](#)
- [Migrating the Data](#)
- [Summary](#)

Viewing Screenshots

 Place the cursor over this icon to load and view all the screenshots for this tutorial. (Caution: This action loads all screenshots simultaneously, so response time may be slow depending on your Internet connection.)

Note: Alternatively, you can place the cursor over an individual icon in the following steps to load and view only the screenshot associated with that step. You can hide an individual screenshot by clicking it.

Overview

Using Oracle SQL Developer Migration Workbench, you can quickly migrate your Microsoft Access database to Oracle. There are four main steps in the database migration process:

Capture the Source Database	The Northwind.xml file generated by the Exported tool contains the database schema information for the Microsoft Access Northwind Traders database. Oracle SQL Developer Migration Workbench uses this file as the basis for building a representation of the structure of the source Microsoft Access database. This structure is called the Captured Model.
Convert the Captured Database	Oracle SQL Developer Migration Workbench uses the Captured Model to convert the captured objects to Oracle-format objects, building up a representation of the structure of the destination database. This structure is called the Converted Model.

Generate the Oracle Database	Oracle SQL Developer Migration Workbench generates DDL statements to create the new Oracle database, based upon the Converted Model objects. Running the DDL statements will result in the creation of the objects in the Oracle database.
Migrate the Data	The last step in the process is to migrate the data. You can do this in one of two ways. You can create a connection from within Oracle SQL Developer to your Microsoft Access source database and migrate the data, or you can export the data from Microsoft Access. Microsoft Access will create a series of sqldr files that you can run from a batch file. The second method is what you will perform in this tutorial.

[Back to Topic List](#)

Prerequisites

For this Hands On Session, the following has already been installed for you

1. Oracle Database 10g.
Note: To repeat this exercise later you can use any Oracle Database above 9.2.0.1
2. Oracle SQL Developer 1.2.1.
Note: Oracle SQL Developer is available for download for FREE from [OTN](#). To install Oracle SQL Developer, unzip it into any directory on your machine.
3. The files you use throughout the tutorial are located in the **\SQLDev_HOS\Migration\files** directory. The files have also been included in the [msamigrate.zip](#) file.

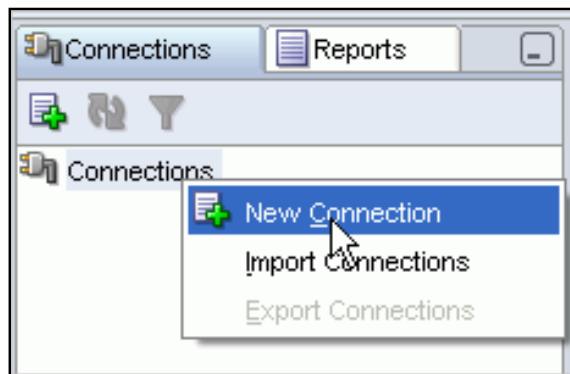
[Back to Topic List](#)

Creating the OMWBREP User

The first step to managing database objects using Oracle SQL Developer is to create a database connection. Perform the following steps:

1. From the directory where you installed Oracle SQL Developer, double-click on **sqldeveloper.exe**.

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

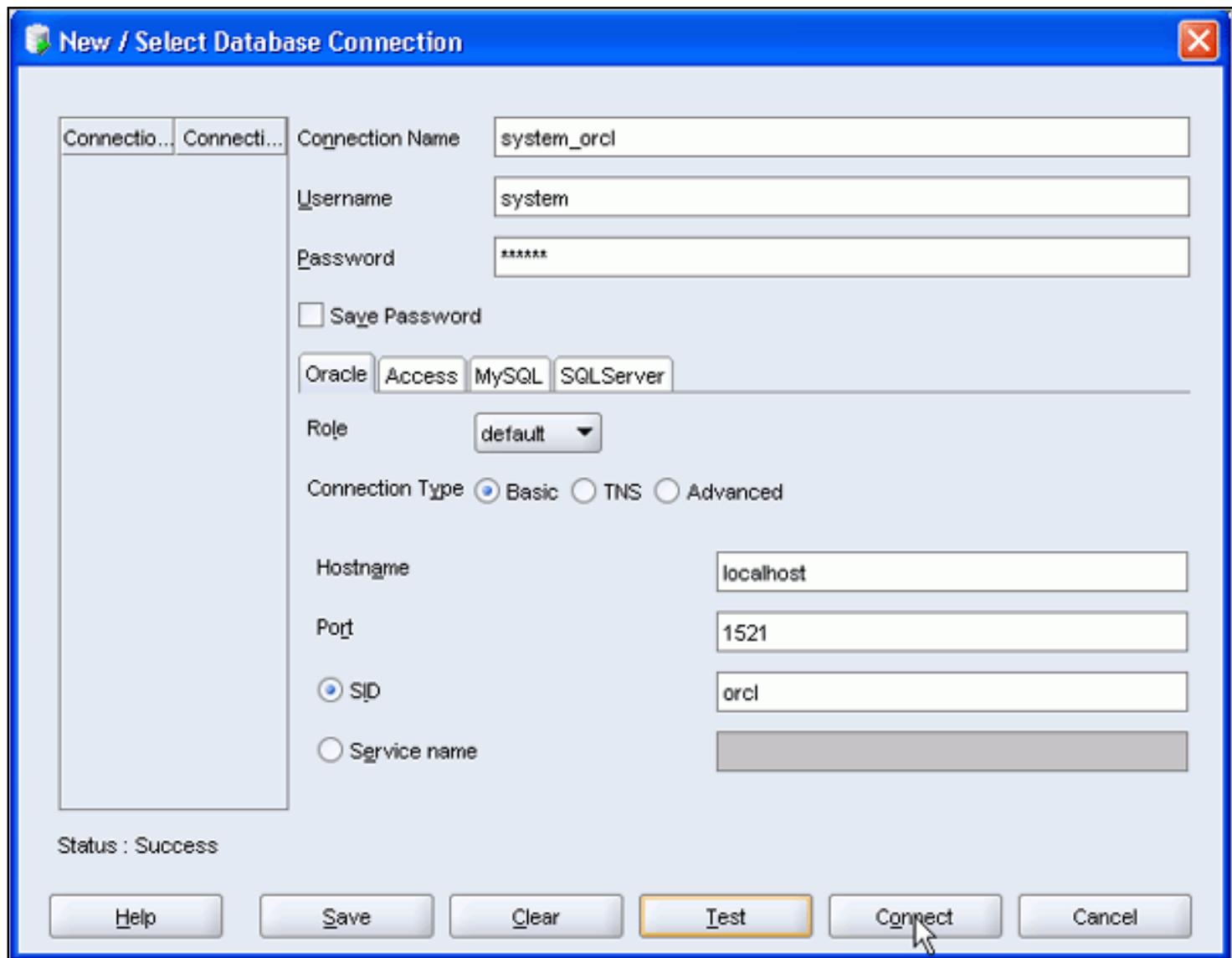


3. Enter **system_orcl** for the Connection Name (or any other name that identifies your connection), **system** for the Username and **<your_password>** for the Password, specify your <hostname> for the Hostname and enter **orcl** for the SID. Then click **Test**.

New / Select Database Connection

Connection Name	system_orcl
Username	system
Password	*****
<input type="checkbox"/> Save Password	
Oracle Access MySQL SQLServer	
Role	default
Connection Type	<input checked="" type="radio"/> Basic <input type="radio"/> TNS <input type="radio"/> Advanced
Hostname	localhost
Port	1521
<input checked="" type="radio"/> SID	orcl
<input type="radio"/> Service name	
Status :	
Help Save Clear Test Connect Cancel	

4. The status of the connection was tested successfully. The connection was not saved however. To save the connection, click **Connect**.

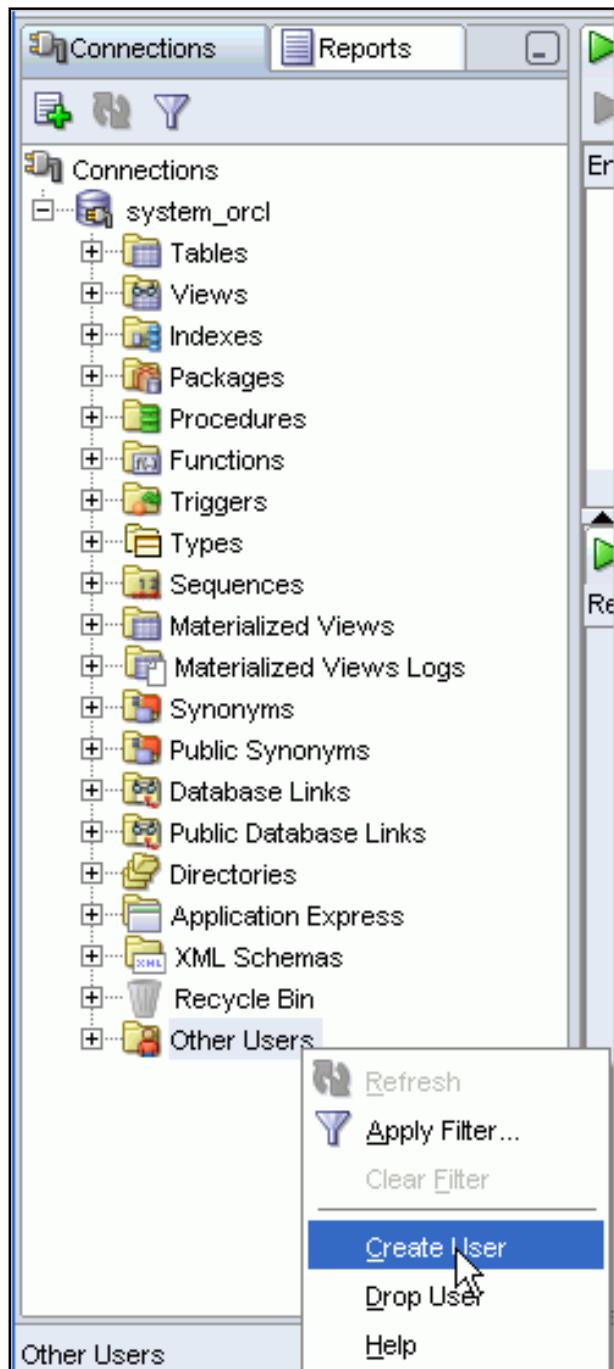


5. The connection was saved and you see the database in the list. Expand **system_orcl**.

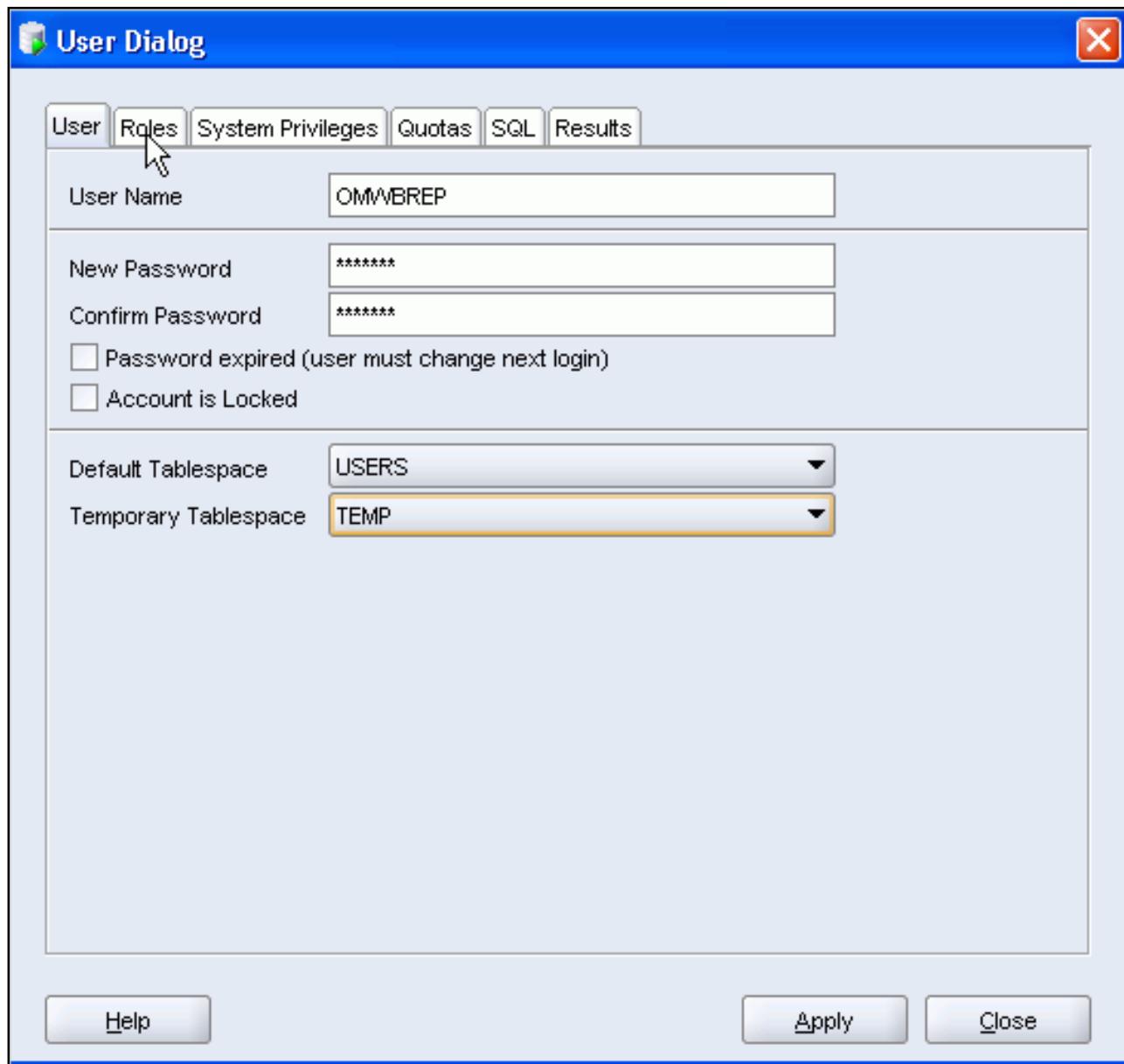


Note: When a connection is opened, a SQL Worksheet is opened automatically. The SQL Worksheet allows you to execute SQL against the connection you just created.

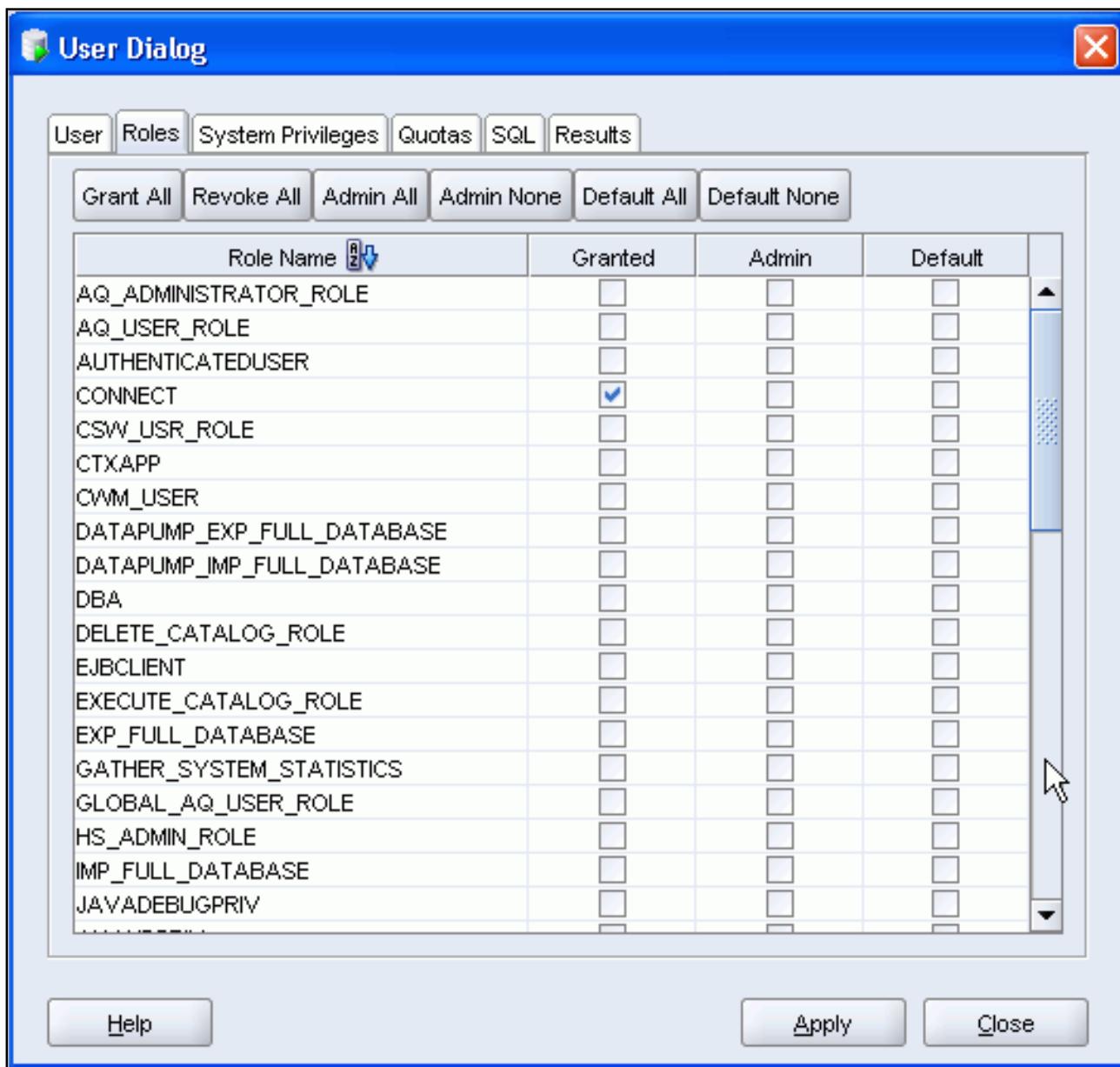
6. Right click on **Other Users** and select **Create User**.



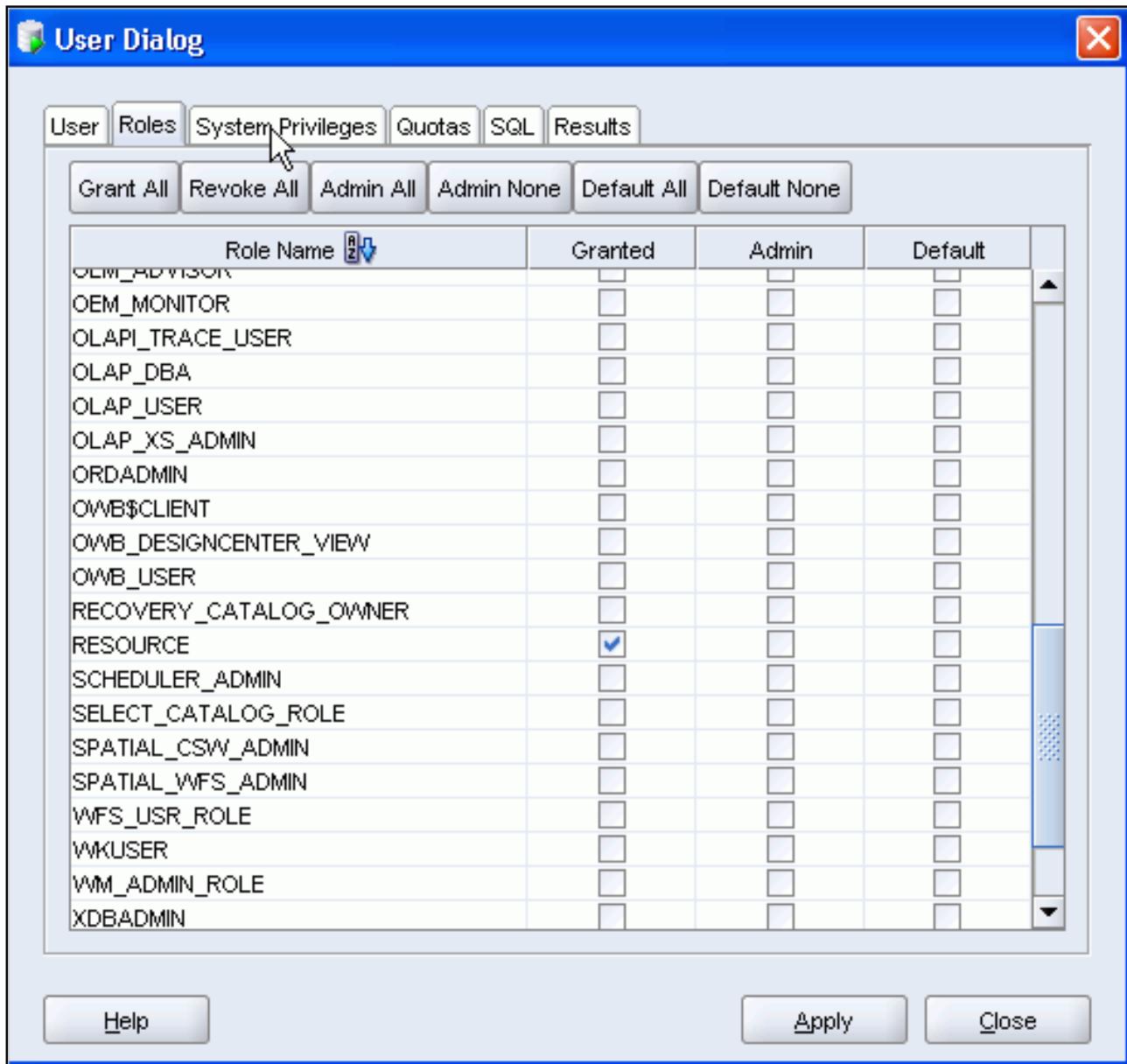
7. Enter **OMWBREP** for the User Name, **omwbrep** for the Password and Confirm Password, **USERS** for the Default Tablespace and **TEMP** for the Temporary Tablespace. Then click the **Roles** tab.



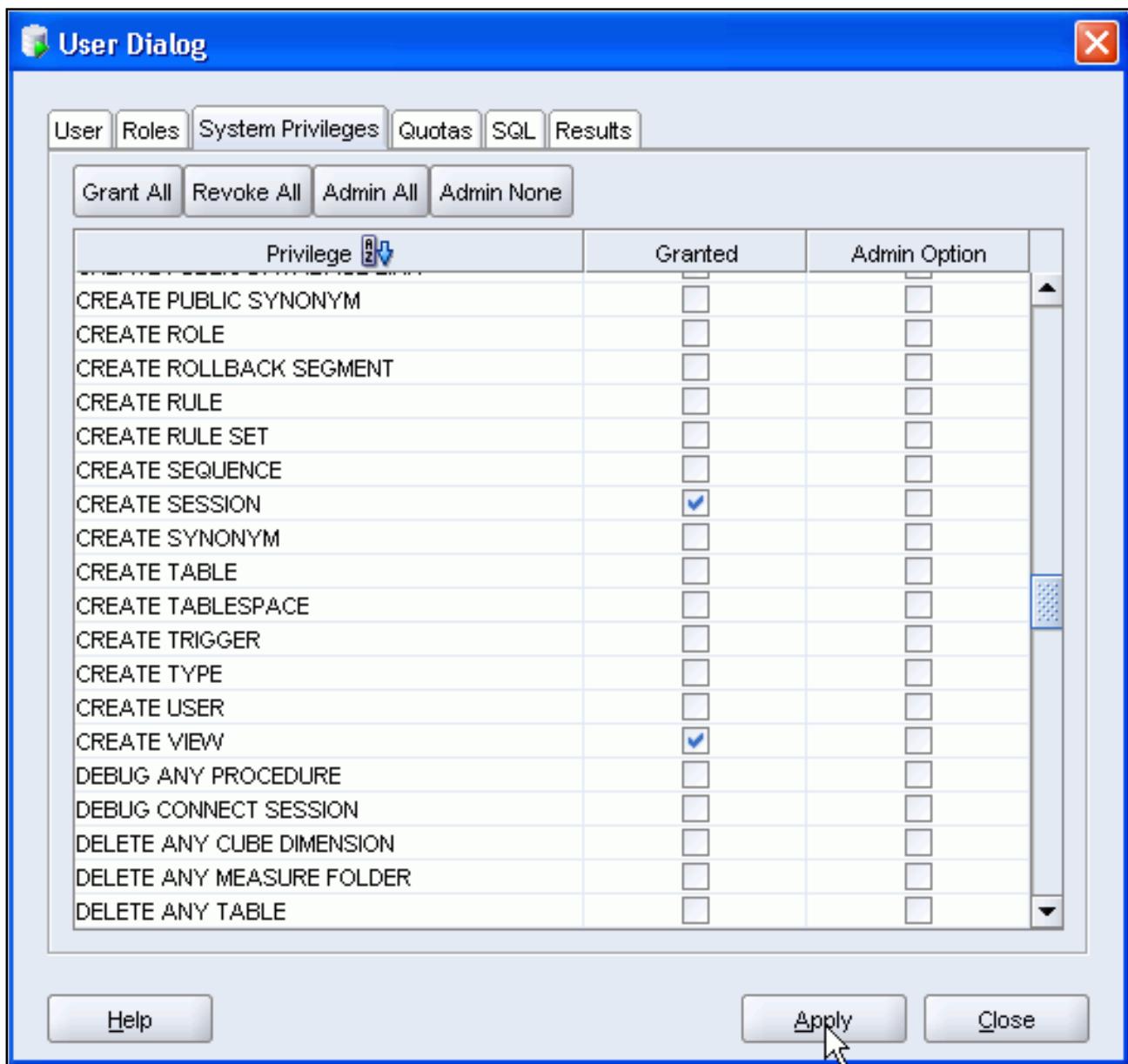
8. Select the Granted checkbox for the **CONNECT** role and scroll down in the list.



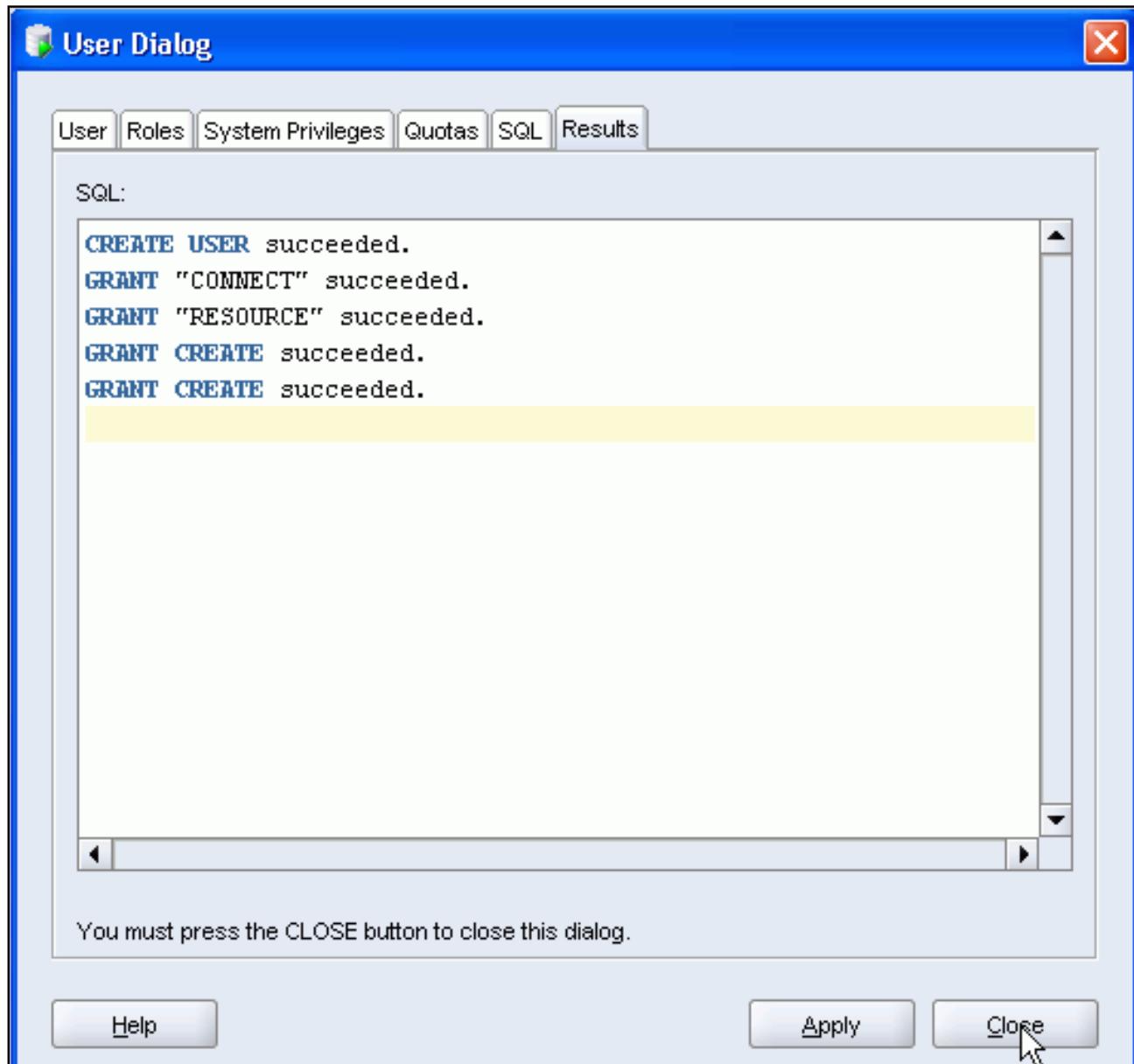
- Select the Granted checkbox for the **RESOURCE** role and click the **System Privileges** tab.



10. Select the Granted checkbox for the **CREATE SESSION** and **CREATE VIEW** privileges and click **Apply**.



11. The OMWBREP user was created successfully.

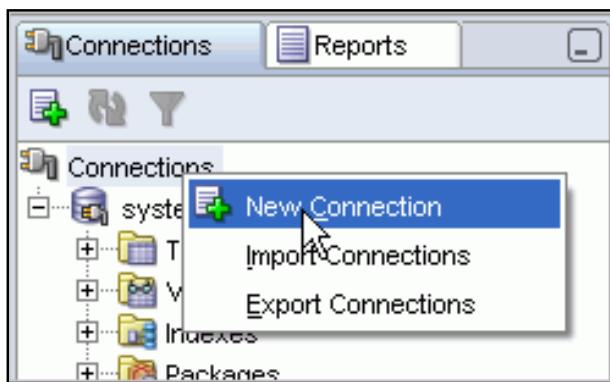


[Back to Topic List](#)

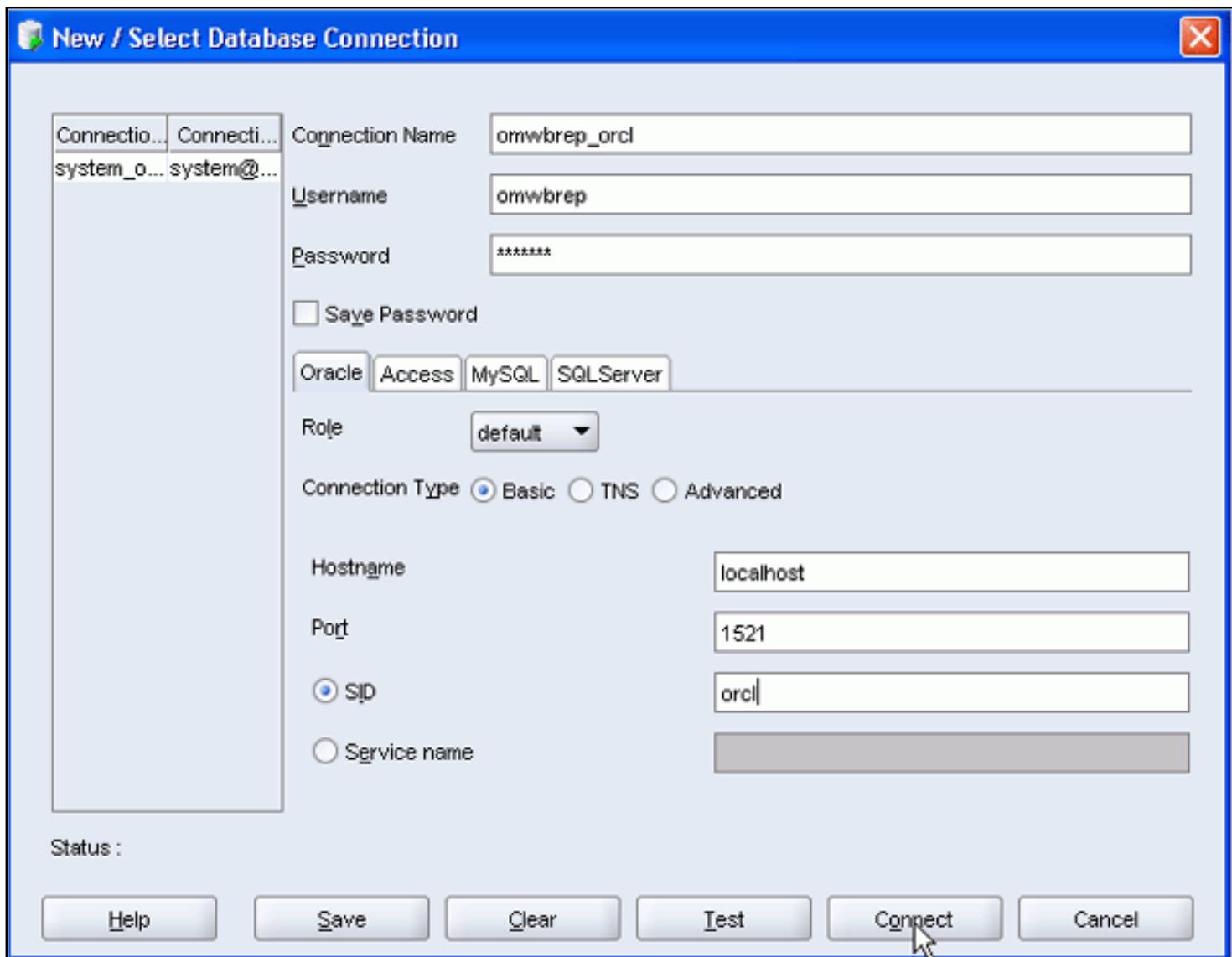
Creating the Migration Repository

In order for the Microsoft Access XML to be converted to Oracle, you need to create a repository to store the required repository tables and PL/SQL packages. Perform the following steps:

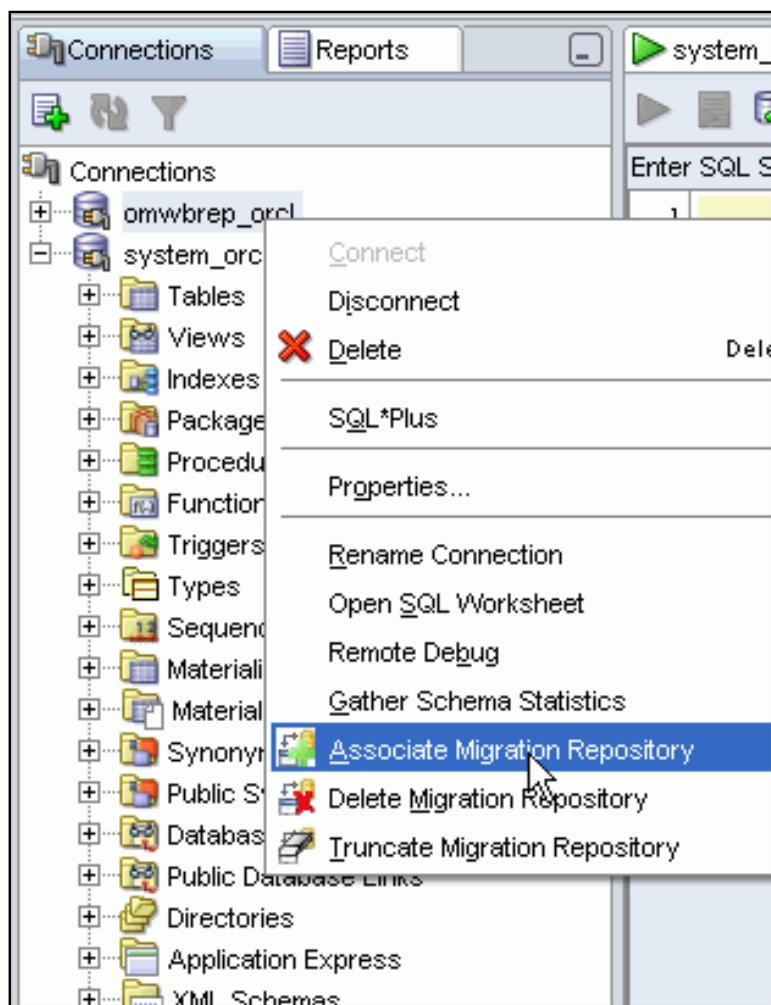
1. Before you create the repository, you need to create a connection to the OMWBREP user. Right click **Connection** and select **New Connection**.



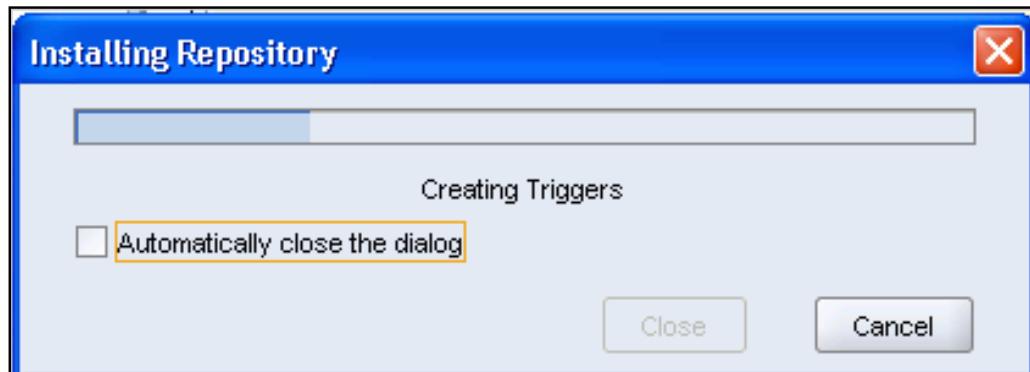
2. Enter **omwbrep_orcl** for the Connection Name (or any other name that identifies your connection), **omwbrep** for the Username and **omwbrep** for the Password and enter **orcl** for the SID. Then click **Connect**.



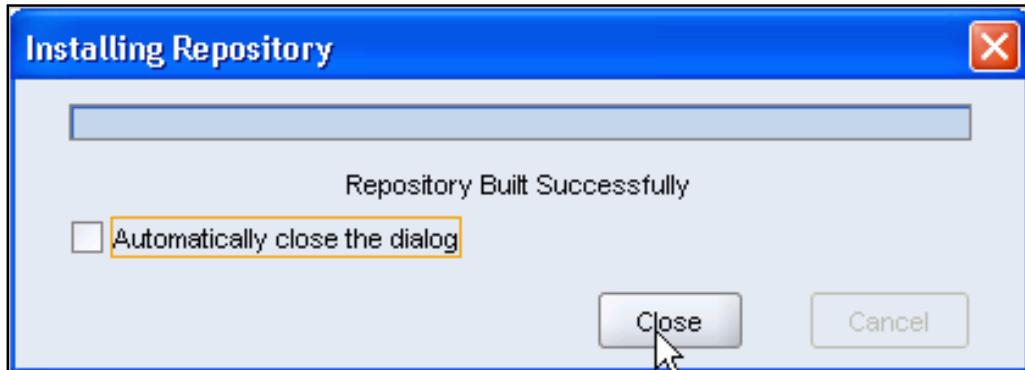
3. Now you can create the repository in this users connection. Right click on the **omwbrep_orcl** connection and select **Associate Migration Repository**.



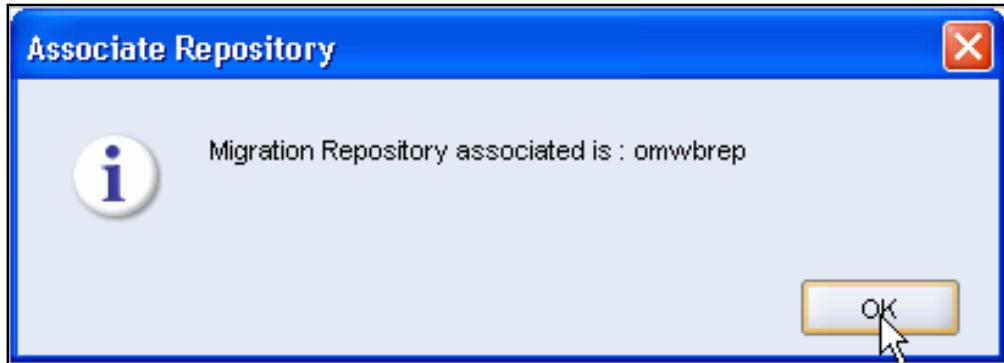
4. The progress window appears.



5. Once the repository is built successfully, click **Close**.



6. Click **OK**.

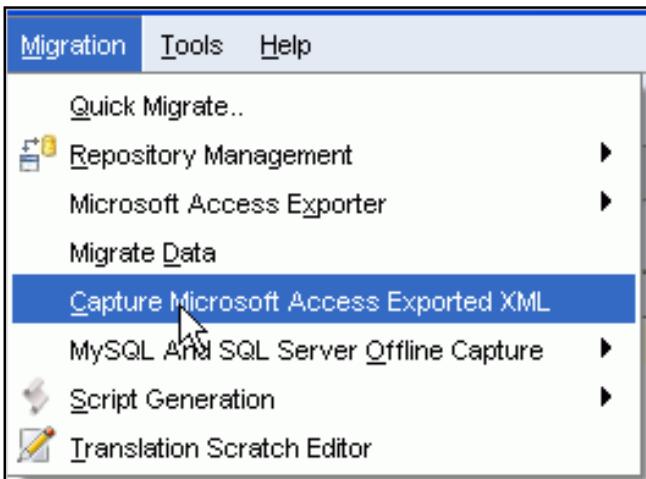


[Back to Topic List](#)

Capturing the Microsoft Access Exported XML

At this point, you can capture the Microsoft Access Exported XML into Oracle SQL Developer. Perform the following steps:

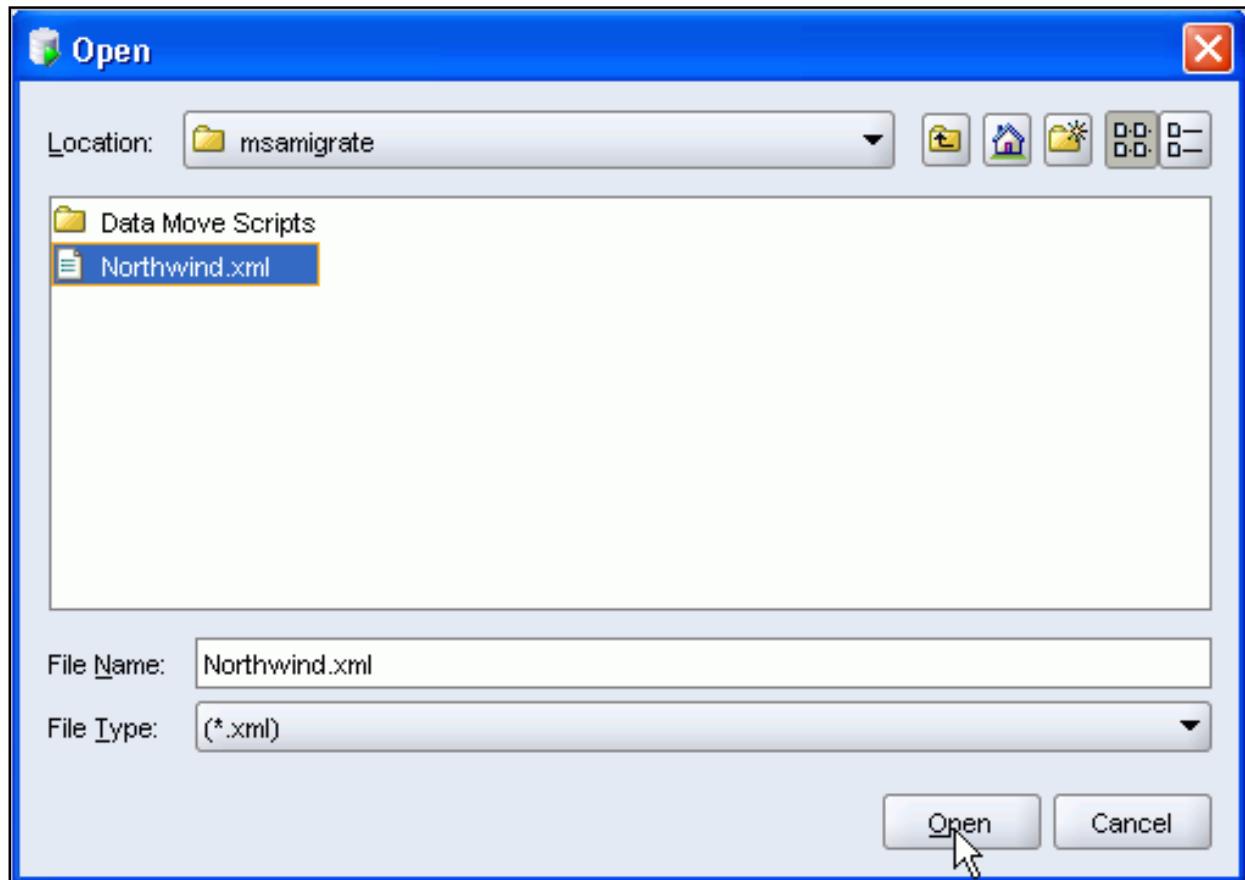
1. Select **Migration > Capture Microsoft Access Exported XML**.



2. Click **Browse...** to find the XML file. The files are in the **\SQLDev_HOS\Migration\files** directory.



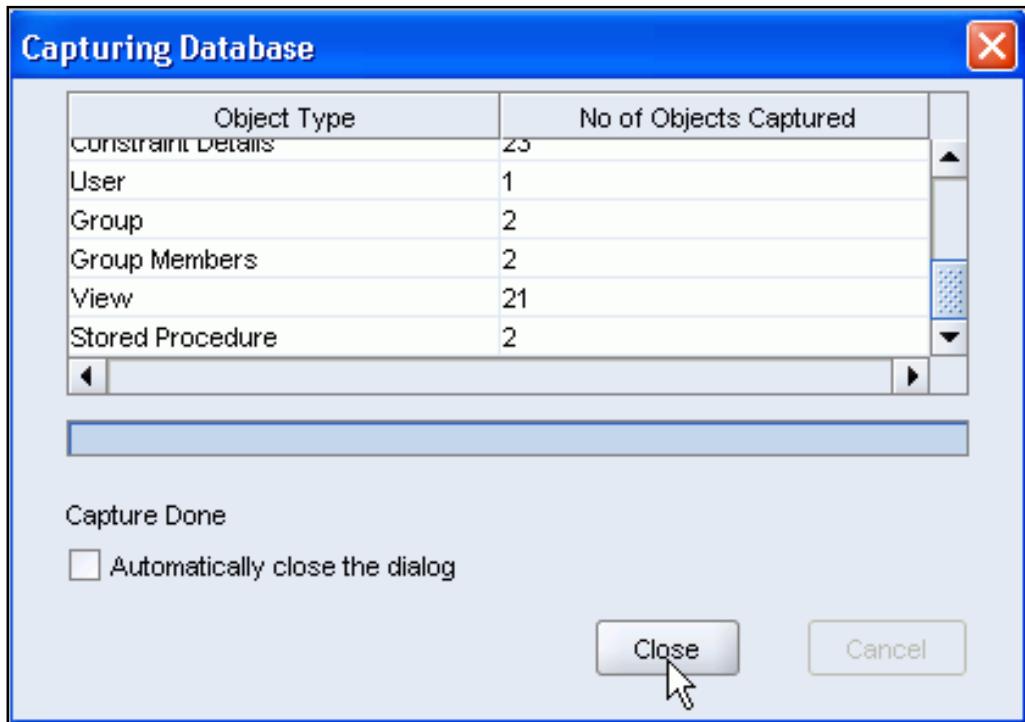
3. Select **Northwind.xml** from the list and click **Open**.



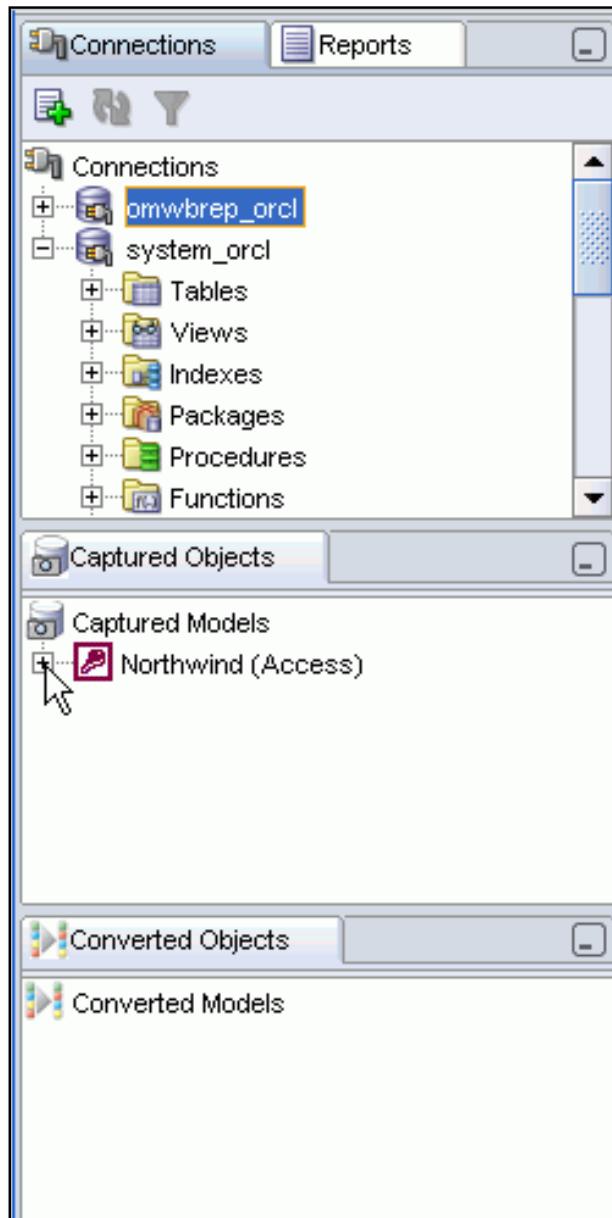
4. Click **OK**.



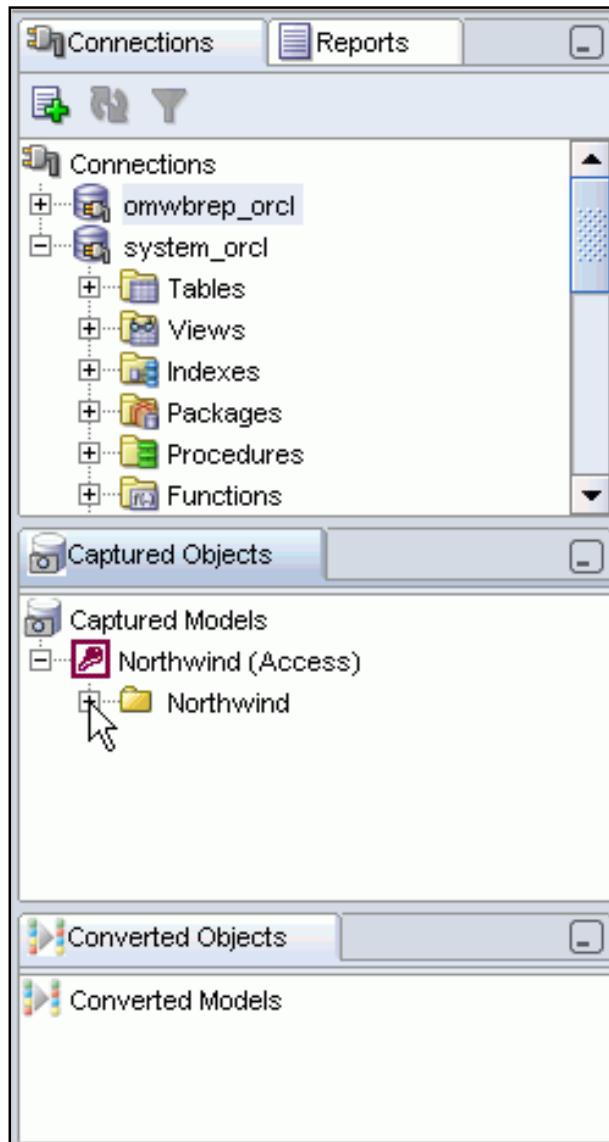
5. The objects are being captured. When done, click **Close**.



6. Under Captured Models, expand **Northwind (Access)**.



7. Expand **Northwind** to see the list of objects that were captured.

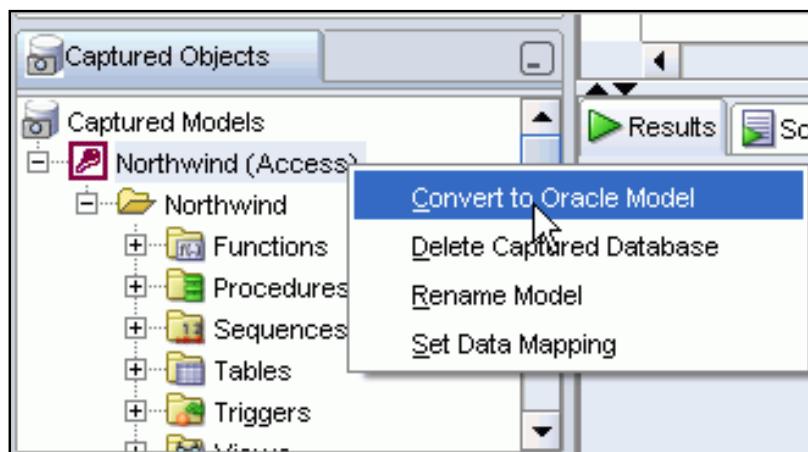


[Back to Topic List](#)

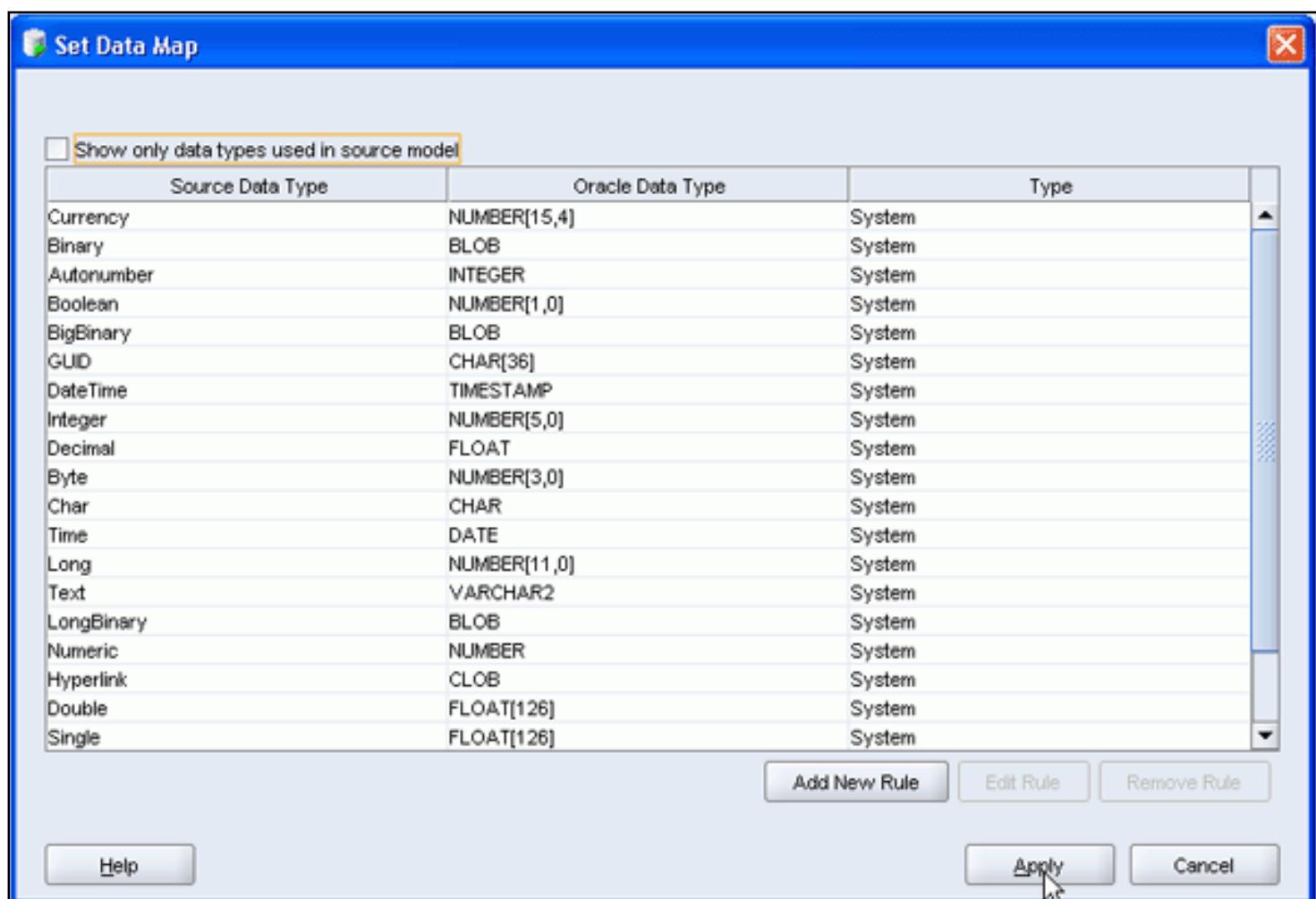
Converting to the Oracle Model

The next step is to convert the captured model to an Oracle model. Perform the following steps:

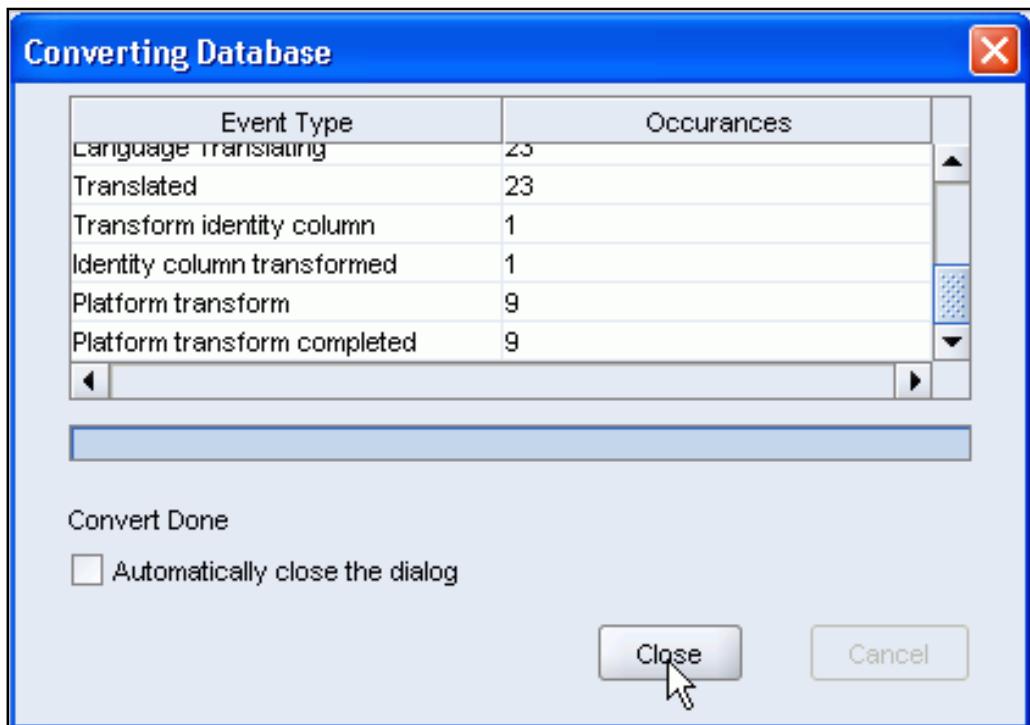
1. Right-click the captured model **Northwind (Access)** and select **Convert to Oracle Model**.



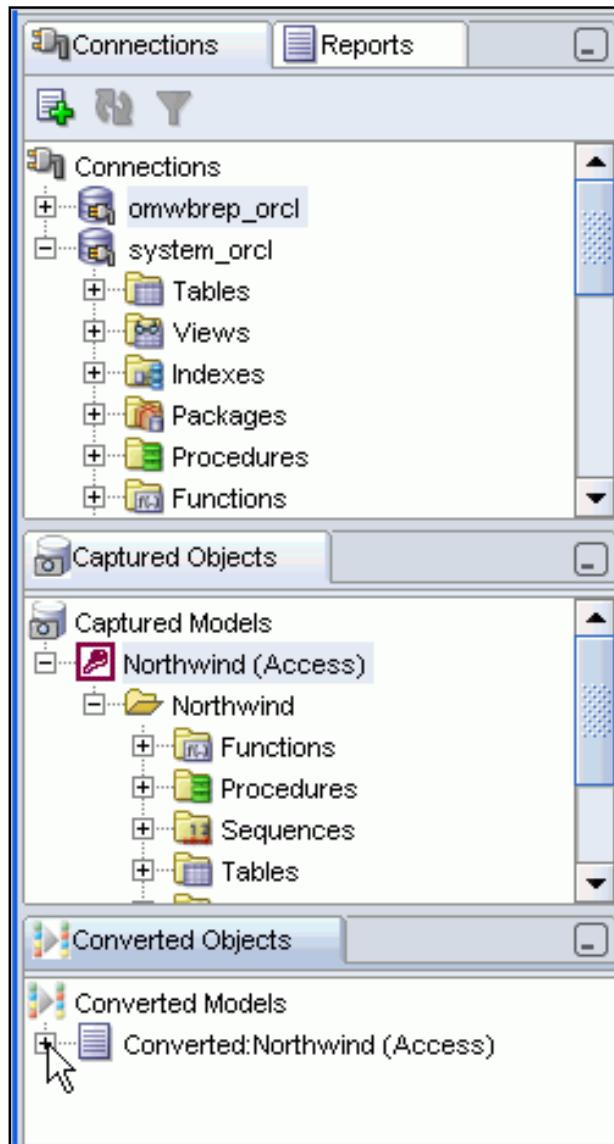
2. The Set Data Map window appears which shows you the Source Data Type and what it will be converted to in the Oracle Model. Click **Apply**.



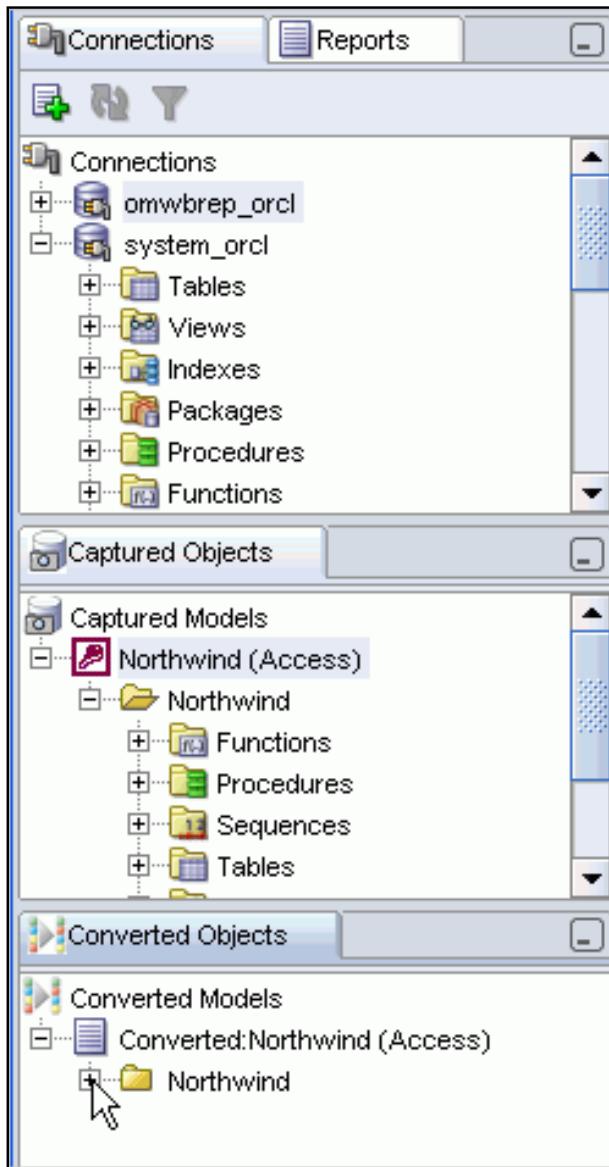
3. The conversion is performed. When done, click **Close**.



4. Expand **Converted Model (Northwind)**.



5. Expand **Northwind**.

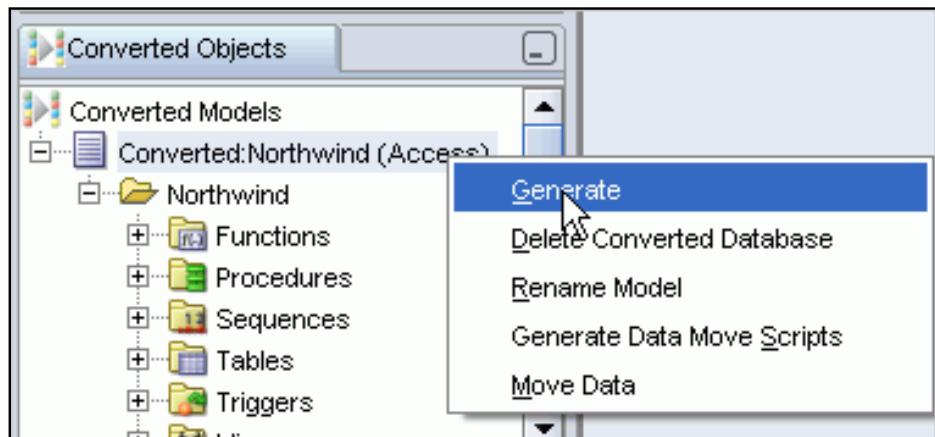


[Back to Topic List](#)

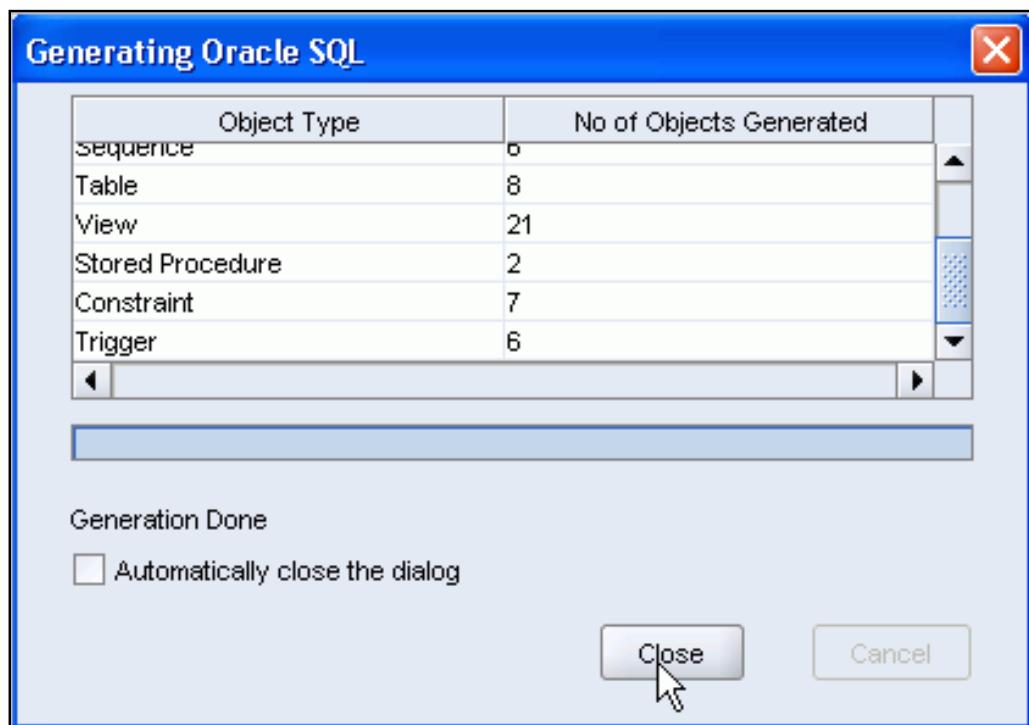
Generating and Executing the Script to Create the Oracle Database Objects

You can now generate the SQL script with DDL statements that can be run to create the objects in an Oracle Database. Perform the following steps:

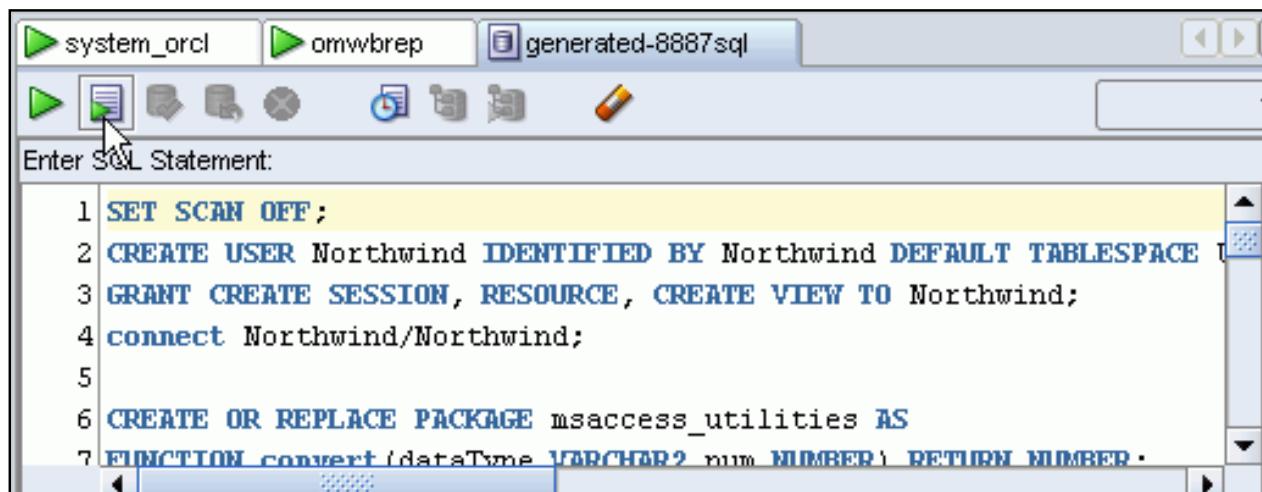
- Under Converted Objects, right click **Converted Northwind (Access)** and select **Generate**.



- The Oracle SQL is being generated. When done, click **Close**.



3. The SQL from the script is shown. Click the **Run Script** icon.



The screenshot shows the Oracle SQL Developer interface. A script editor window is open, titled 'generated-8887sql'. The toolbar at the top has several icons, including a green play button icon which is highlighted with a red box. Below the toolbar, the text 'Enter SQL Statement:' is displayed. The code area contains the following SQL script:

```
1 SET SCAN OFF;
2 CREATE USER Northwind IDENTIFIED BY Northwind DEFAULT TABLESPACE TBS_Northwind QUOTA 10M ON TBS_Northwind TEMPORARY TABLESPACE TBS_Northwind TEMP QUOTA 10M ON TBS_Northwind;
3 GRANT CREATE SESSION, RESOURCE, CREATE VIEW TO Northwind;
4 connect Northwind/Northwind;
5
6 CREATE OR REPLACE PACKAGE msaccess_utilities AS
7 FUNCTION convert (dateTime VARCHAR2) RETURN NUMBER;
```

4. Select the **system_orcl** connection and click **OK**.



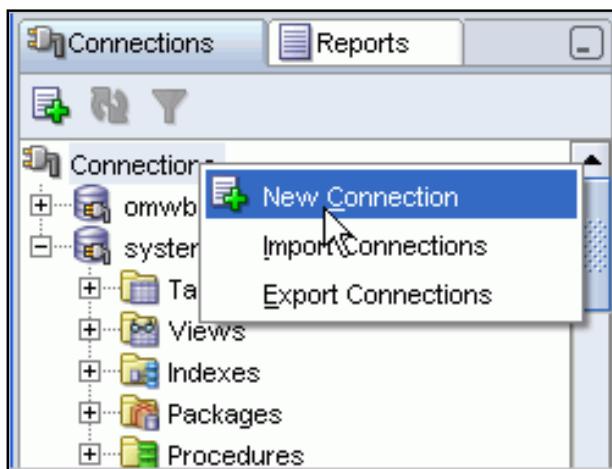
5. The results are displayed from the script execution.

The screenshot shows the Oracle SQL Developer interface. At the top, there are three tabs: 'system_orcl', 'omwbrep', and 'generated-8887sql'. The 'generated-8887sql' tab is active. Below the tabs is a toolbar with various icons. To the right of the toolbar, it says '4.59523678 seconds' and 'system_orcl'. The main area is a text editor labeled 'Enter SQL Statement:' containing several 'ALTER VIEW' statements. Below this is a message area displaying the results of the executed statements, all of which ended with 'succeeded.'.

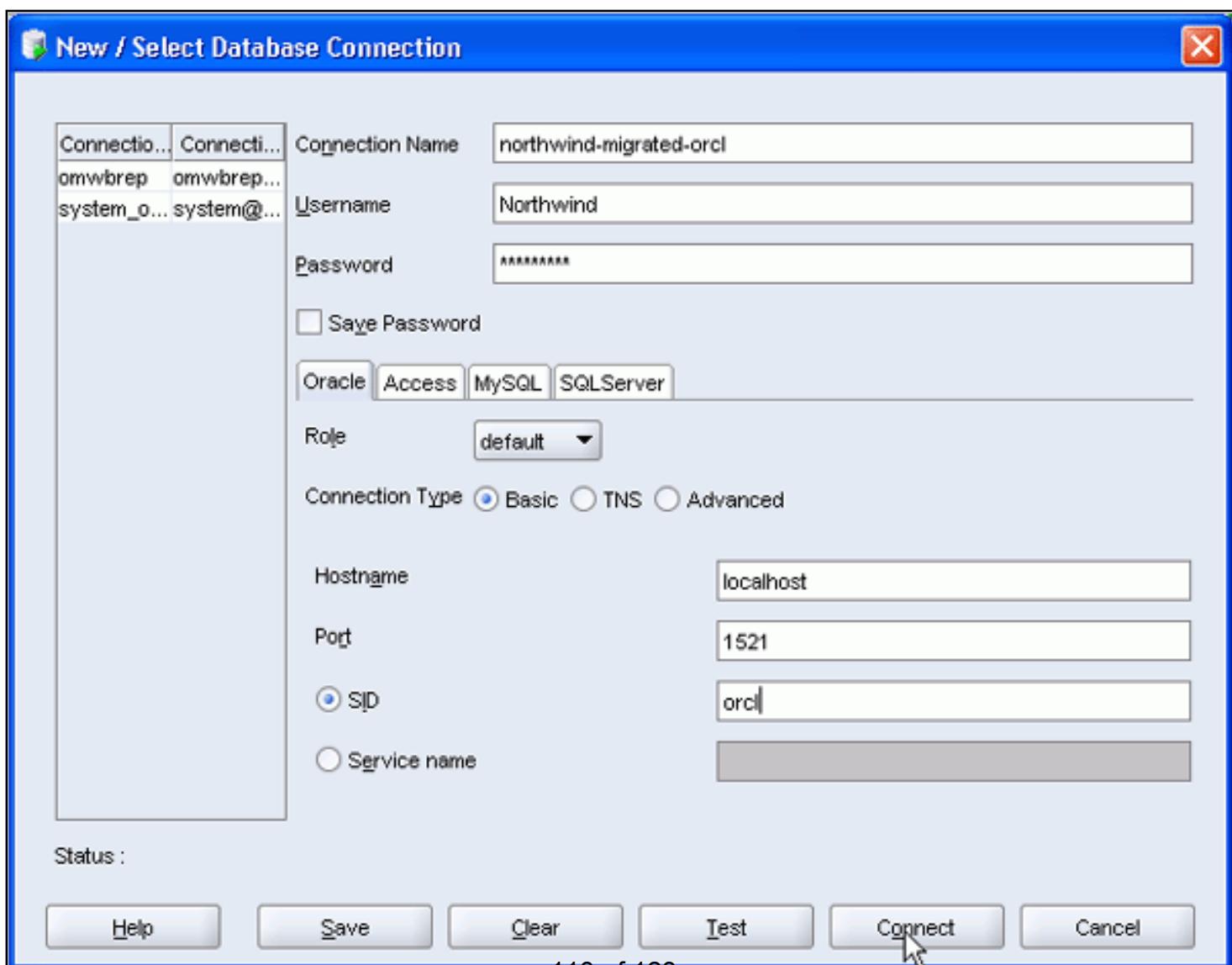
```
I291 ALTER VIEW Products_by_Category COMPILE;
I292
I293 ALTER VIEW Quarterly_Orders COMPILE;
I294
I295 ALTER VIEW Quarterly_Orders_by_Product COMPILE;
I296
I297
```

ALTER VIEW Summary_of_Sales_by_Year succeeded.
ALTER VIEW Ten_Most_Expensive_Products succeeded.
ALTER VIEW Alphabetical_List_of_Products succeeded.
ALTER VIEW Product_Sales_for_1997 succeeded.
ALTER VIEW Category_Sales_for_1997 succeeded.
ALTER VIEW Current_Product_List succeeded.
ALTER VIEW Customers_and_Suppliers_by_Cit succeeded.
ALTER VIEW Employee_Sales_by_Country succeeded.
ALTER VIEW Invoices succeeded.
ALTER VIEW Invoices_Filter succeeded.
ALTER VIEW Orders_Qry succeeded.
ALTER VIEW Products_Above_Average_Price succeeded.
ALTER VIEW Products_by_Category succeeded.
ALTER VIEW Quarterly_Orders succeeded.
ALTER VIEW Quarterly_Orders_by_Product succeeded.

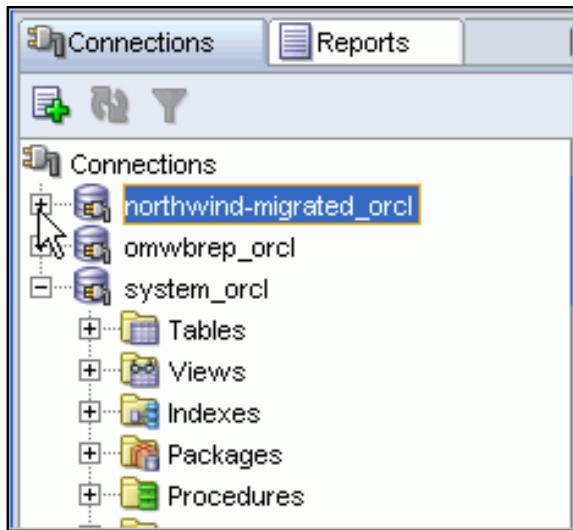
6. Now that your scripts have run successfully, you can create a connection for the Northwind user. Right click **Connection** then select **New Connection**.



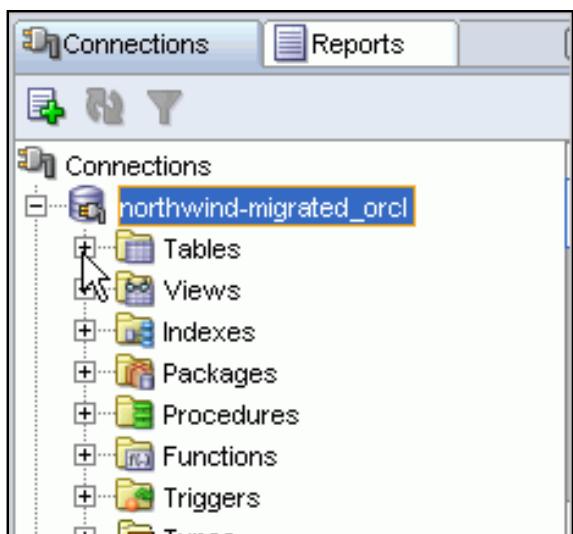
7. Enter **northwind-migrated_orcl** for the Connection Name (or any other name that identifies your connection), **Northwind** for the Username and **northwind** for the Password and enter **orcl** for the SID. Then click **Connect**.



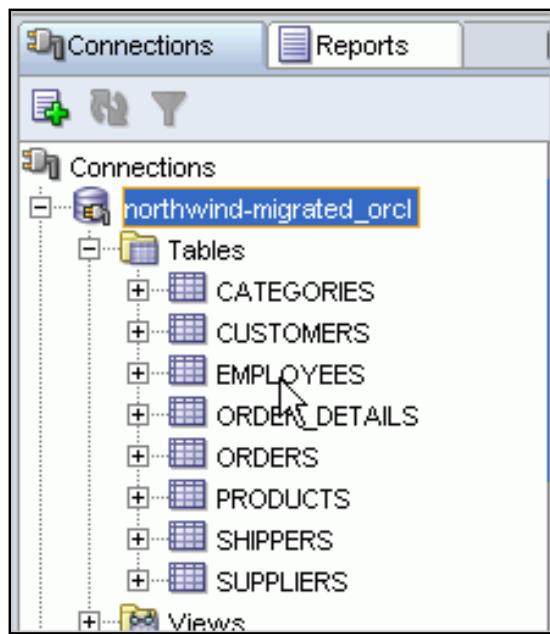
8. Expand **northwind-migrated_orcl**.



9. Expand **Tables**.



10. Notice that the database tables that were converted to Oracle are listed. Select the **EMPLOYEES** table.



11. Select the **Data** tab.

Column Name	Data Type	Nullable	Data Default	COLUMN ID	Primary Key	COMMENT
EMPLOYEEID	NUMBER(11,0)	No	(null)	1	1	Number automatically generated
LASTNAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)	(null) (null)
FIRSTNAME	VARCHAR2(10 BYTE)	No	(null)	3	(null)	(null) (null)
TITLE	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)	Employee's title
TITLEOFCOURTEY	VARCHAR2(25 BYTE)	Yes	(null)	5	(null)	Title used in salutation
BIRTHDATE	DATE	Yes	(null)	6	(null)	(null) (null)
HIREDATE	DATE	Yes	(null)	7	(null)	(null) (null)
ADDRESS	VARCHAR2(60 BYTE)	Yes	(null)	8	(null)	Street or post office address
CITY	VARCHAR2(15 BYTE)	Yes	(null)	9	(null)	(null) (null)
REGION	VARCHAR2(15 BYTE)	Yes	(null)	10	(null)	State or province
POSTALCODE	VARCHAR2(10 BYTE)	Yes	(null)	11	(null)	(null) (null)
COUNTRY	VARCHAR2(15 BYTE)	Yes	(null)	12	(null)	(null) (null)
HOMEPHONE	VARCHAR2(24 BYTE)	Yes	(null)	13	(null)	Phone number at home
EXTENSION	VARCHAR2(4 BYTE)	Yes	(null)	14	(null)	Internal telephone extension
PHOTO	VARCHAR2(255 BYTES)	Yes	(null)	15	(null)	Picture of employee
NOTES	CLOB	Yes	(null)	16	(null)	General information
REPORTSTO	NUMBER(11,0)	Yes	(null)	17	(null)	Employee's supervisor

12. Notice that there currently is no data in the table. You will migrate the data in the next topic of this tutorial.

EMPLOYEEID	LASTNAME	FIRSTNAME	TITLE	TITLEOFCOURTESY	BIRTHDATE	HIREDATE	ADDRESS	CITY	REGION	POSTALCODE	COUNTRY	HOMEPHONE	EXTENSION	PHOTO	NOTES	REPORTSTO
1	Sampath	Chinthaka	Software Engineer	Mr.	1981-07-15	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	2
2	King	Mark	Software Engineer	Mr.	1973-12-09	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	1
3	Allen	Andrea	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	5
4	Wingard	Bob	Software Engineer	Mr.	1973-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	6
5	Tucker	Robert	Software Engineer	Mr.	1970-09-13	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	7
6	Spencer	Karen	Software Engineer	Mr.	1975-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	8
7	Adams	Janet	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	9
8	Peacock	Loyd	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	10
9	Feeney	Patricia	Software Engineer	Mr.	1973-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	11
10	Mavrovi	Michael	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	12
11	Fuller	Tim	Software Engineer	Mr.	1973-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	13
12	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	14
13	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	15
14	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	16
15	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	17
16	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	18
17	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	19
20	King	King	Software Engineer	Mr.	1972-01-12	2005-01-01	123 Main St	Anytown	CA	95111-4212	US	555-555-1234	1234	Photo placeholder	Employee details	20

[Back to Topic List](#)

Migrating the Data

The data has already been exported into scripts from Microsoft Access. You will import the data using the scripts

provided. Perform the following steps:

1. Open a DOS command prompt and execute the following commands:

```
cd <location where files are>  
oracle_ctl
```

The location of the files is \SQLDev_HOS\Migration\files\data

oracle_ctl is a bat file that contains statements to load the data. It uses sqlldr to load the data.

2. Switch to Oracle SQL Developer and click **Refresh** to see that the data has been loaded for the CUSTOMERS table.



3. The data for the table has been loaded successfully.

This screenshot shows the same Oracle SQL Developer interface as above, but now the EMPLOYEES table is populated with data. The data consists of 9 rows, each representing an employee with their Employee ID, Last Name, First Name, Title, Title of Courtesy, and Birth Date. The data is as follows:

EMPLOYEEID	LASTNAME	FIRSTNAME	TITLE	TITLEOFCOURTESY	BIRTHDATE
1	Davolio	Nancy	Sales Representative	Ms.	08-DEC-1958
2	Fuller	Andrew	Vice President, Sales	Dr.	19-FEB-1952
3	Leverling	Janet	Sales Representative	Ms.	30-AUG-1963
4	Peacock	Margaret	Sales Representative	Mrs.	19-SEP-1965
5	Buchanan	Steven	Sales Manager	Mr.	04-MAR-1960
6	Suyama	Michael	Sales Representative	Mr.	02-JUL-1962
7	King	Robert	Sales Representative	Mr.	29-MAJ-1963
8	Callahan	Laura	Inside Sales Coordin...	Ms.	09-JAN-1966
9	Dodsworth	Anne	Sales Representative	Ms.	02-JUL-1962

Summary

In this tutorial, you learned how to:

- ☒ Create the OMWBREP User
- ☒ Create the Migration Repository
- ☒ Capture the Microsoft Access Exported XML
- ☒ Convert to the Oracle Model
- ☒ Generate and Execute the Script to Create the Oracle Database Objects
- ☒ Migrate the Data

[Back to Topic List](#)

 **Move your mouse over this icon to hide all screenshots.**