

Oracle SQL Developer 使用教程

2010.12.19 atyu30 纪念

内容目录

创建数据库连接	2
使用 Table 对话框添加新表	8
更改表定义	18
添加表约束	22
向表中添加数据.....	27
访问数据.....	41
使用可用的报表.....	47
添加用户定义的报表	52
创建并编译 PL/SQL 过程.....	55
运行 PL/SQL 过程.....	66
调试 PL/SQL 过程.....	70
使用 SQL Worksheet.....	79
导出数据.....	83
session 功能.....	89
观察 TableSpace 功能.....	93
Sql Developer 中 code snippets 工具栏.....	94
连接 MySQL.....	96

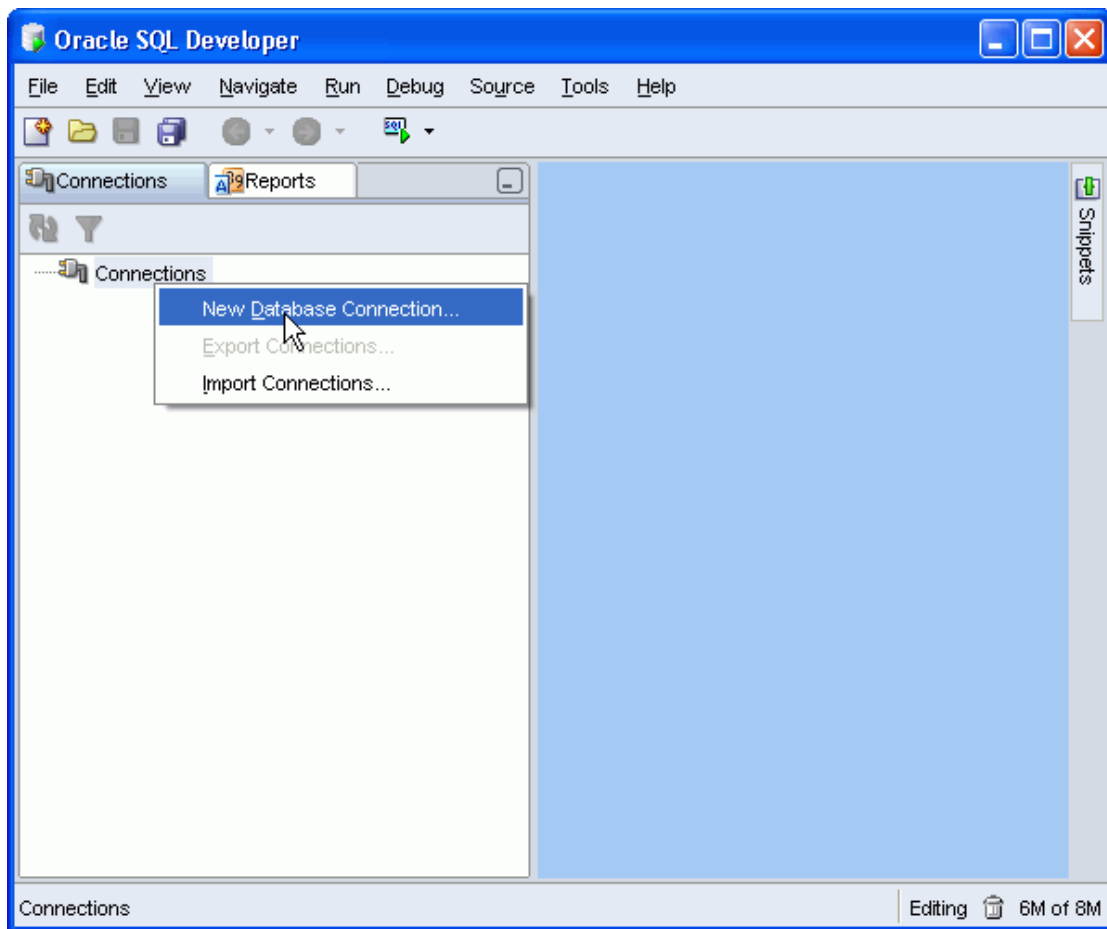
创建数据库连接

使用 Oracle SQL Developer 管理数据库对象首先要创建数据库连接。执行以下步骤：

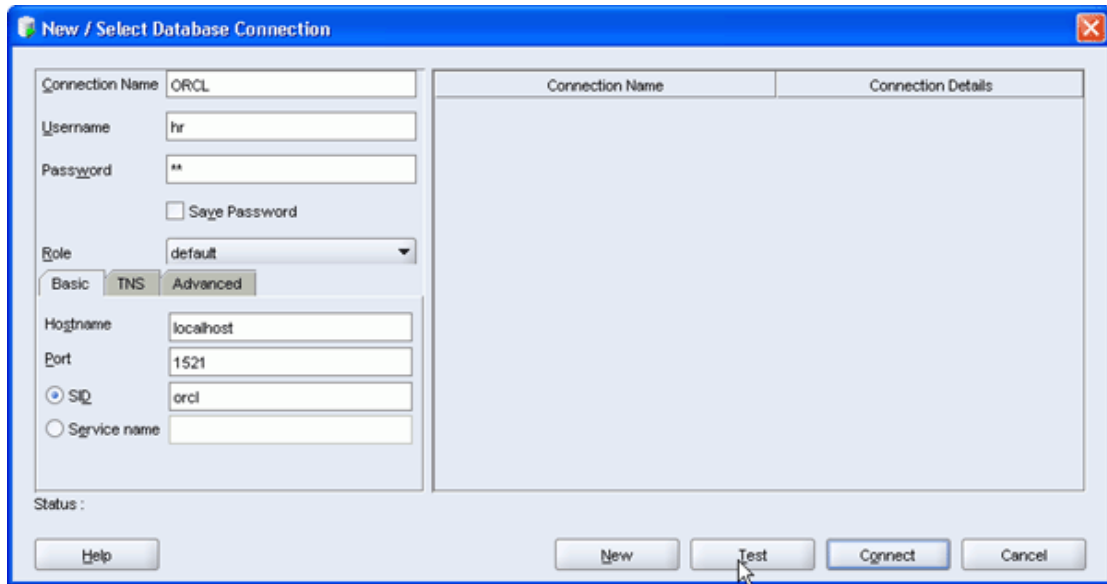
1. 打开 Windows 资源管理器，双击 <path>\sqldeveloper\sqldeveloper.exe。

注：也可以在非 Windows 环境中执行 sqldeveloper。

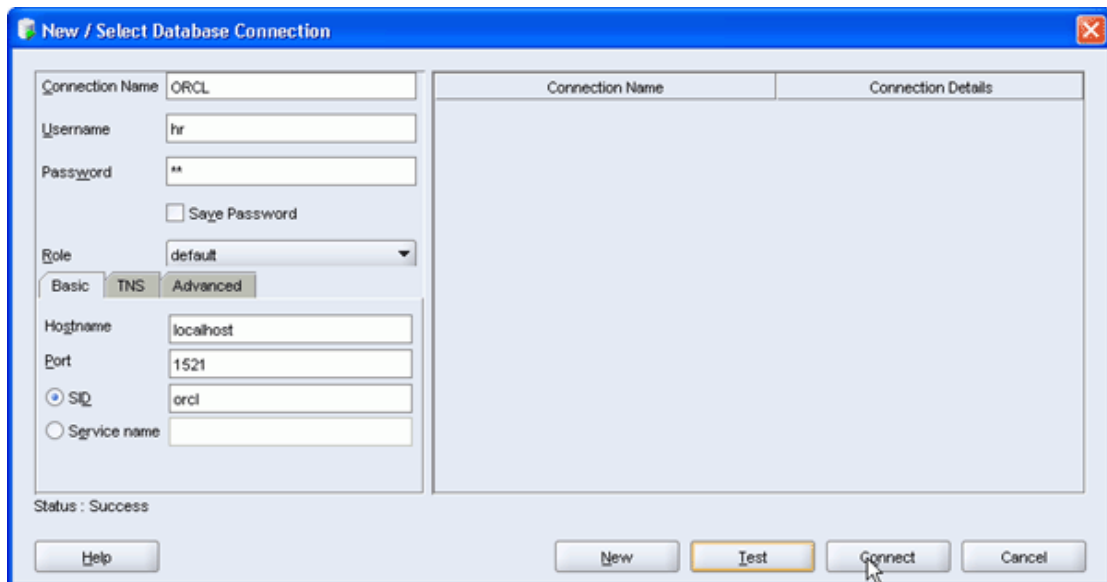
2. 在 Connections 选项卡中，右键单击 Connections 并选择 New Database Connection。



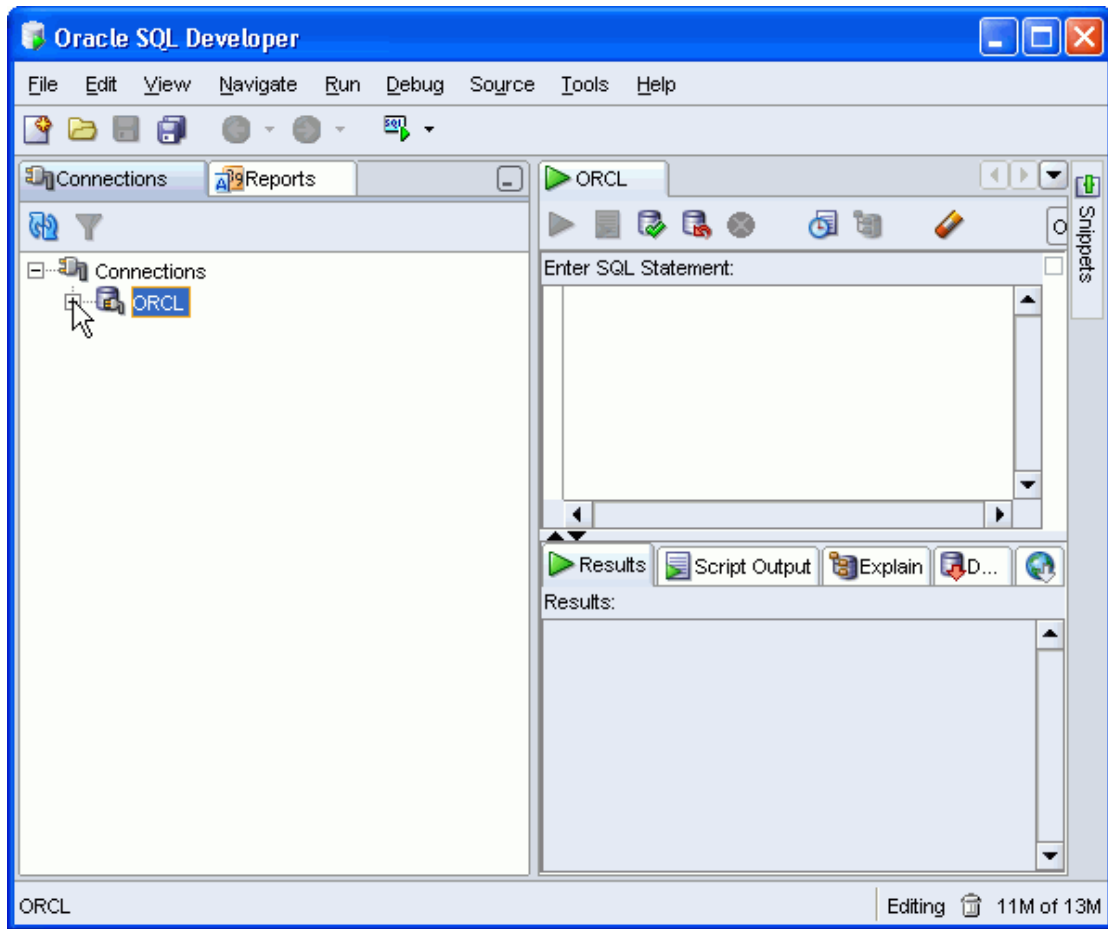
3. 在 Connection Name 域中输入 <your_db_sid_name>，在 Username 和 Password 域中输入 hr，指定 <hostname> 作为 Hostname，在 SID 域中输入 <your_db_sid>。然后单击 Test。



4. 连接状态已成功测试。但没有保存该连接。要保存该连接，请单击 Connect。

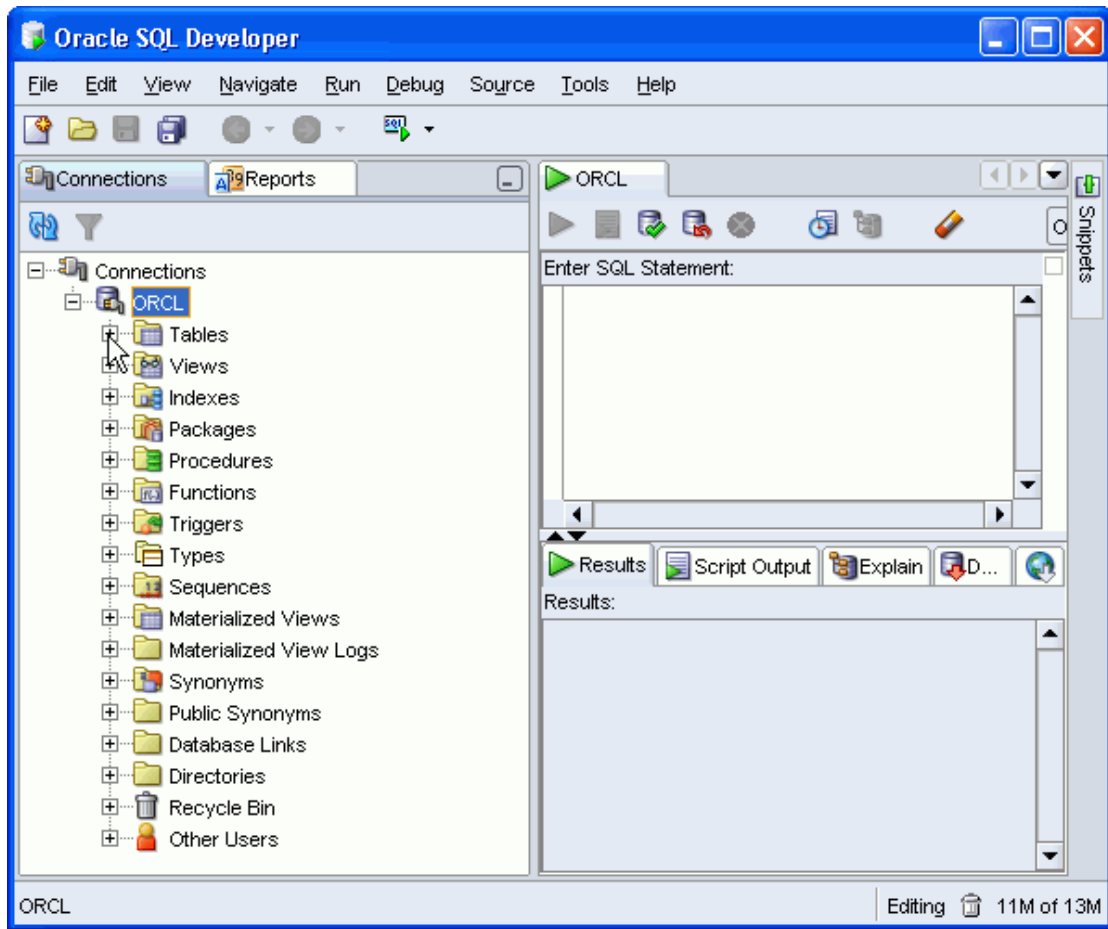


5. 保存了该连接，可以在列表中看到该数据库。展开 ORCL。

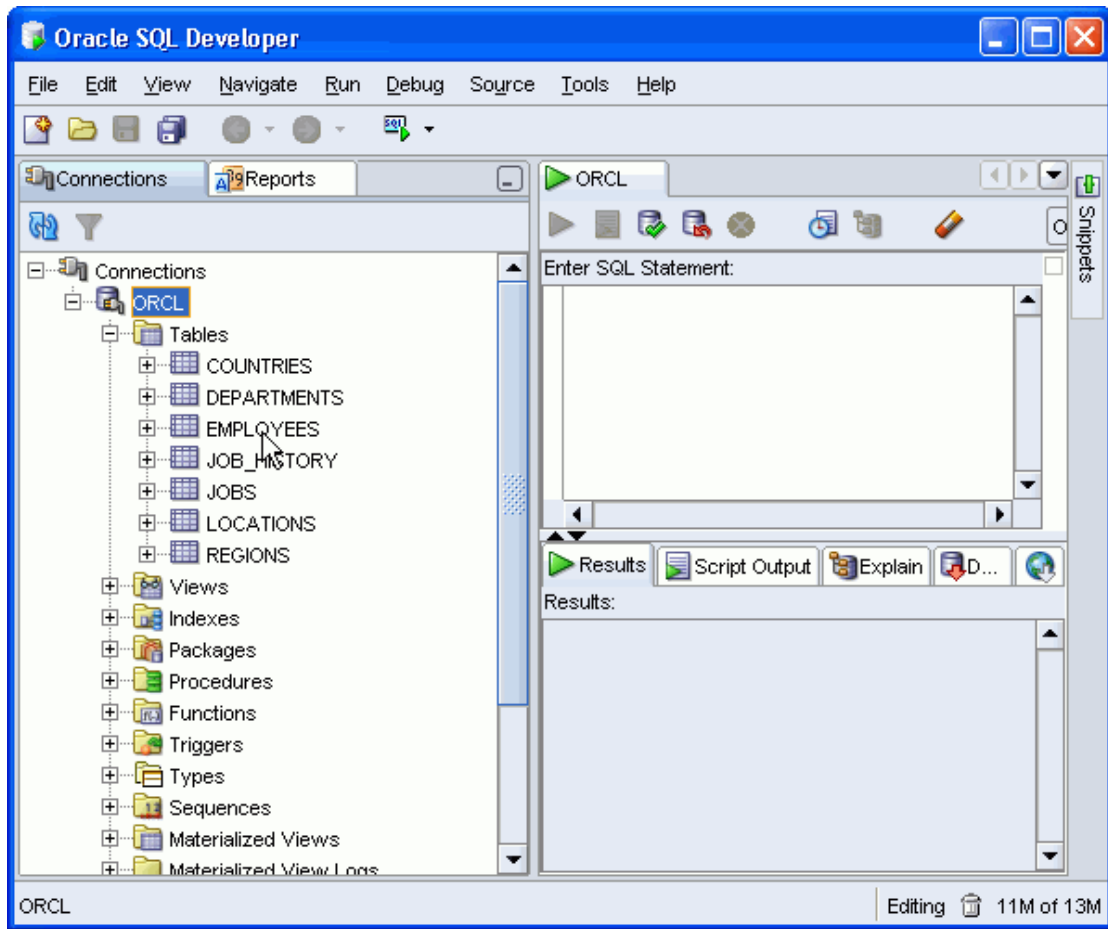


注：打开连接时，自动打开 SQL Worksheet。SQL Worksheet 允许针对刚创建的连接执行 SQL。

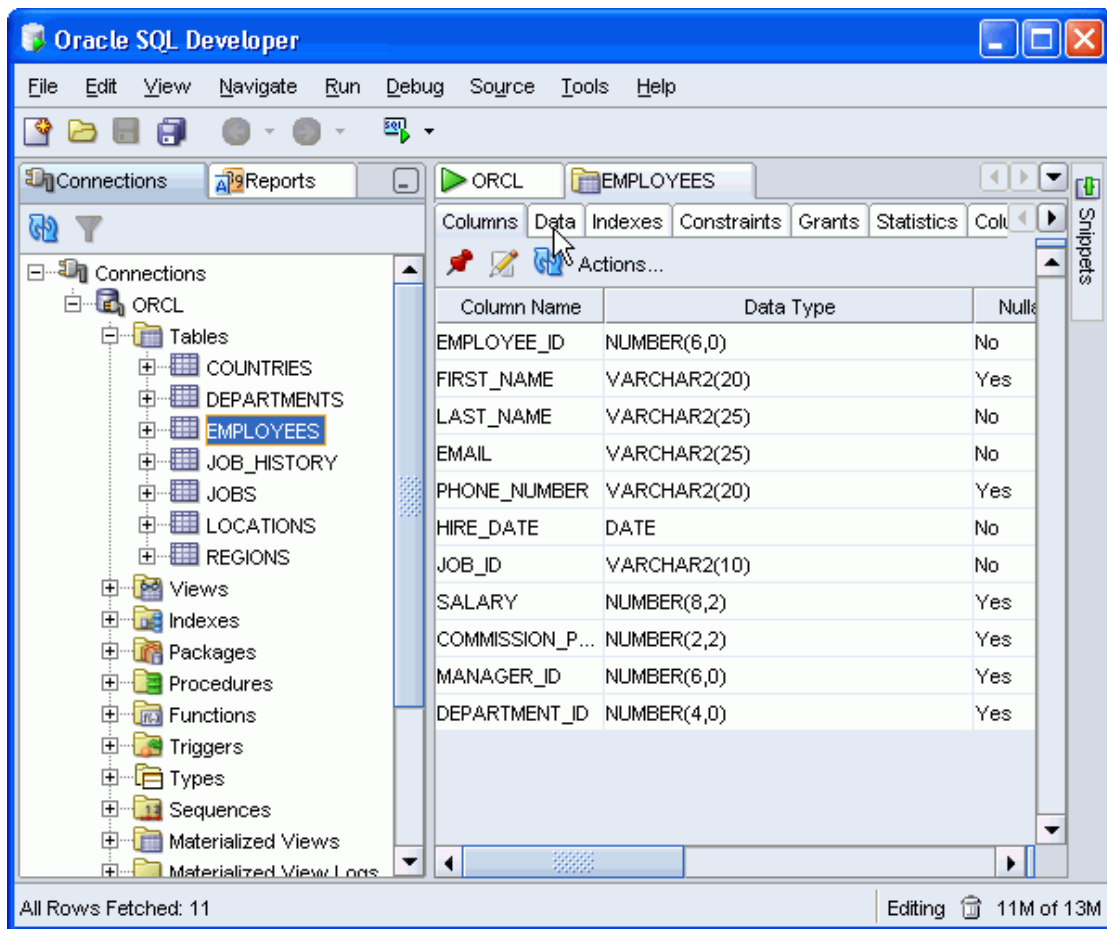
6. 展开 Tables。



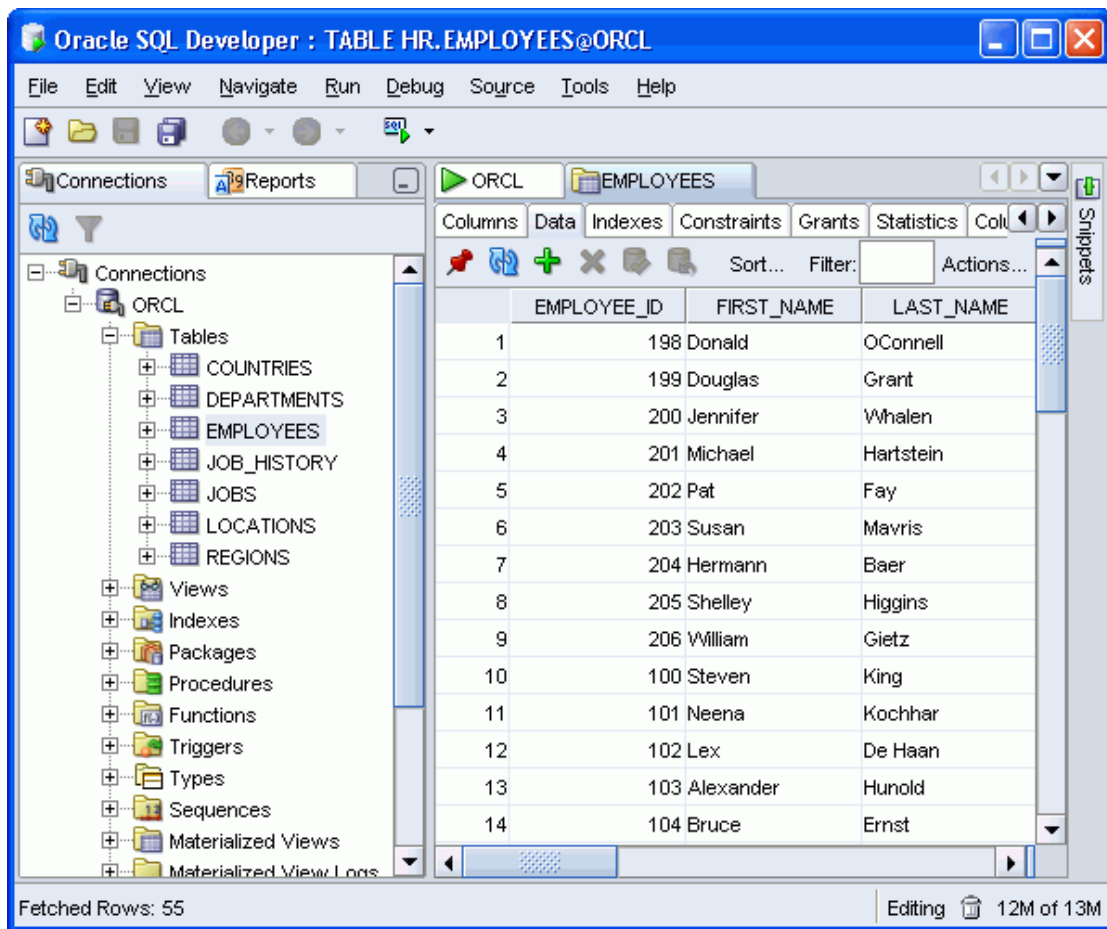
7. 选择 EMPLOYEES 表，查看表定义。



8. 要查看该数据，请单击 Data 选项卡。



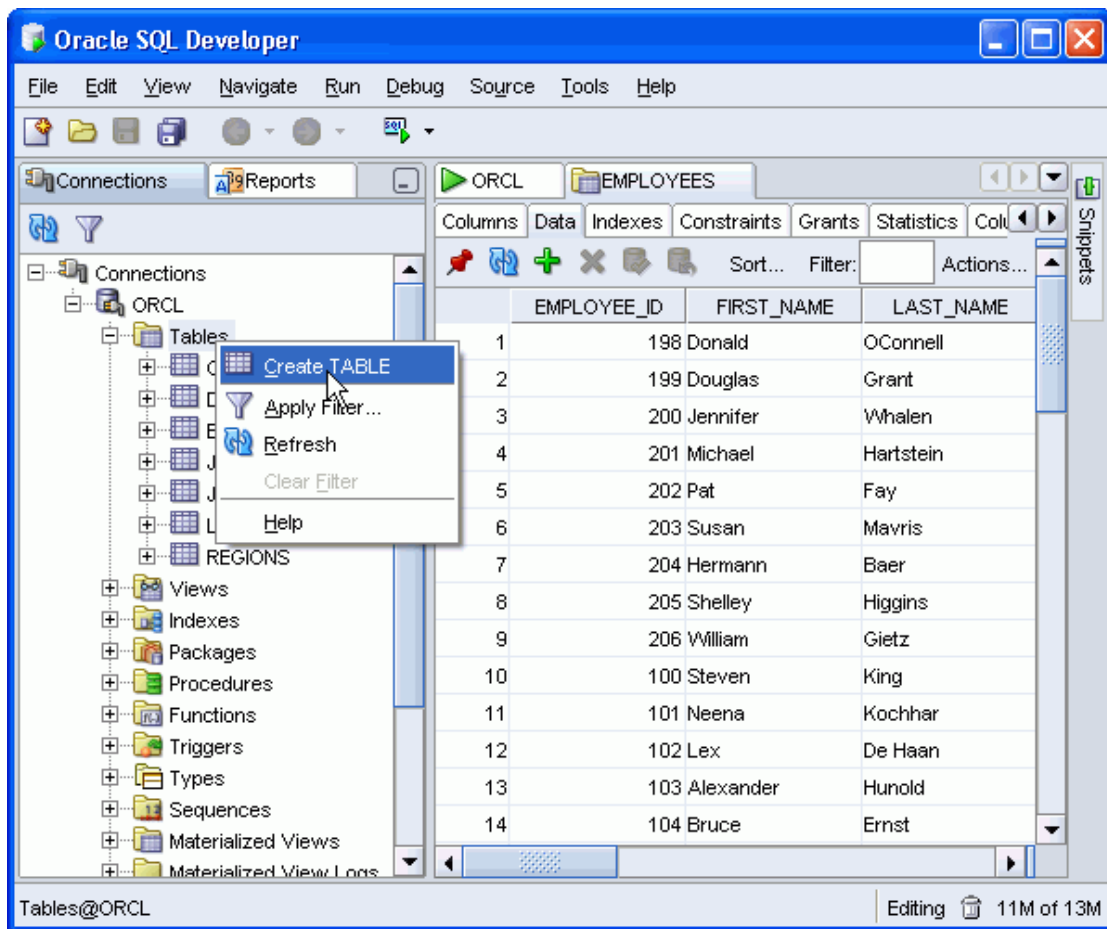
9. 数据随即显示。



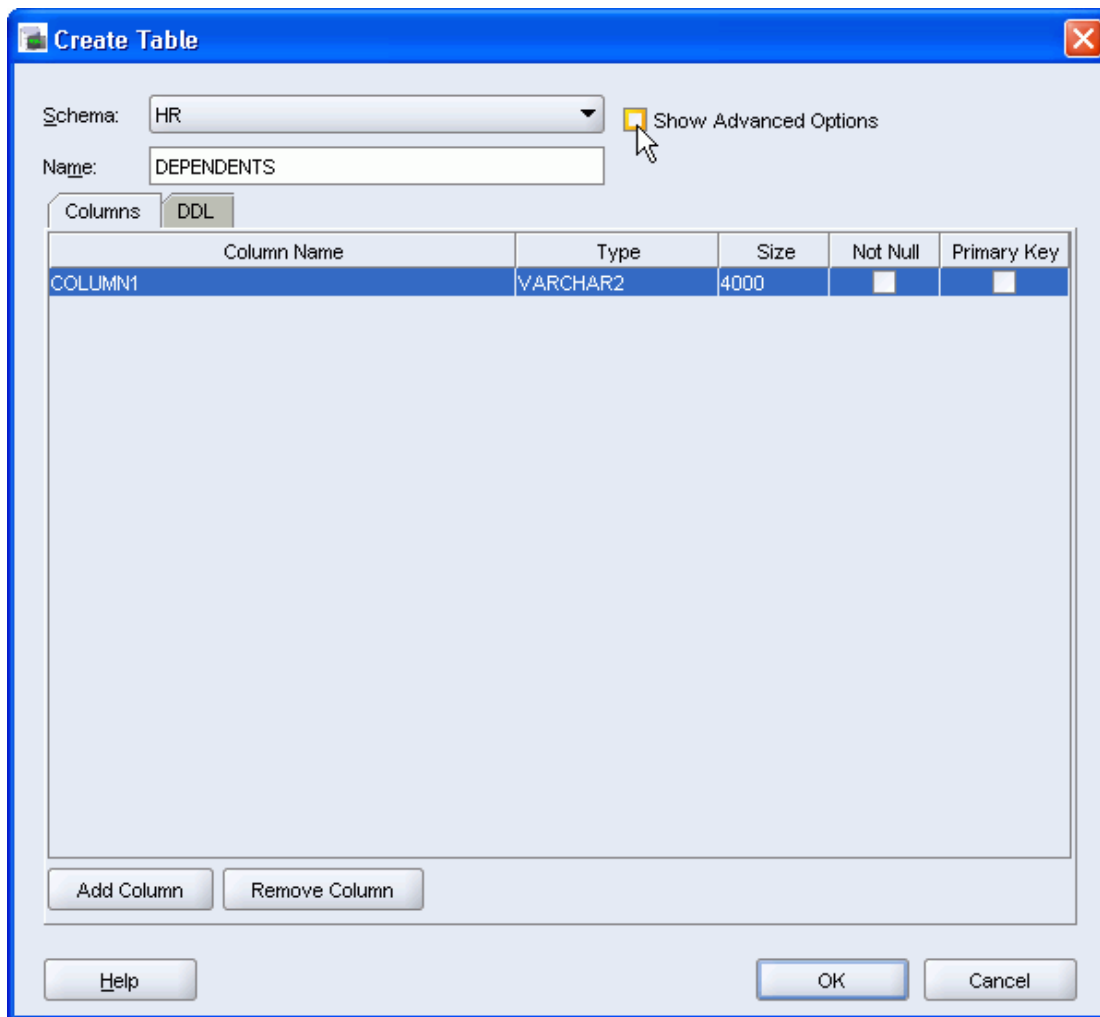
使用 **Table** 对话框添加新表

新建一个名为 DEPENDENTS 的表，它包含一个带有 EMPLOYEES 表的外键。执行以下步骤：

1. 右键单击 Tables 并选择 Create TABLE。



2. 输入 DEPENDENTS 作为 Table Name, 然后单击 Show Advanced Options 复选框。

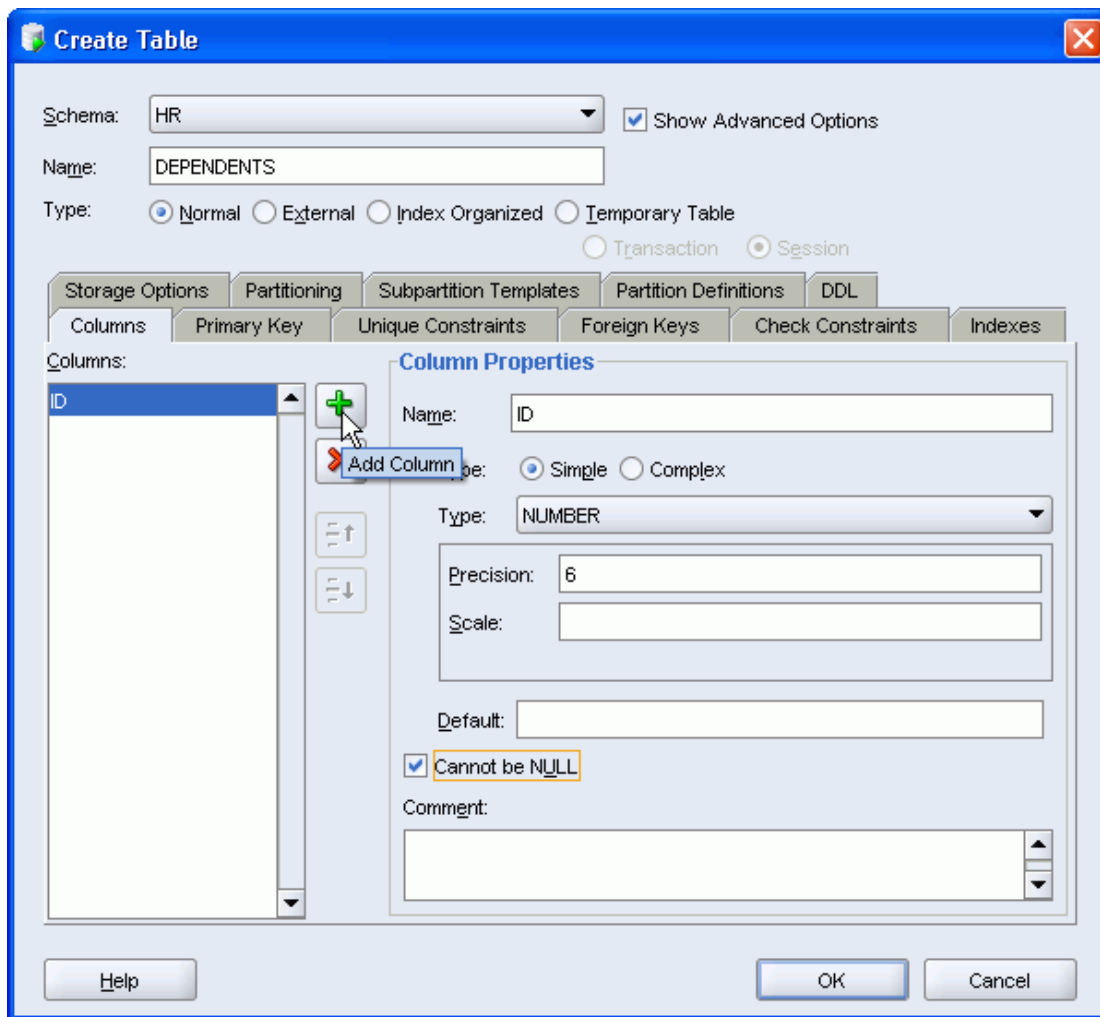


The image shows a 'Create Table' dialog box with a blue title bar. It contains fields for 'Schema' (set to 'HR') and 'Name' (set to 'DEPENDENTS'). A 'Show Advanced Options' checkbox is checked, with a mouse cursor pointing to it. Below these fields are two tabs: 'Columns' and 'DDL'. The 'Columns' tab is active, displaying a table with the following data:

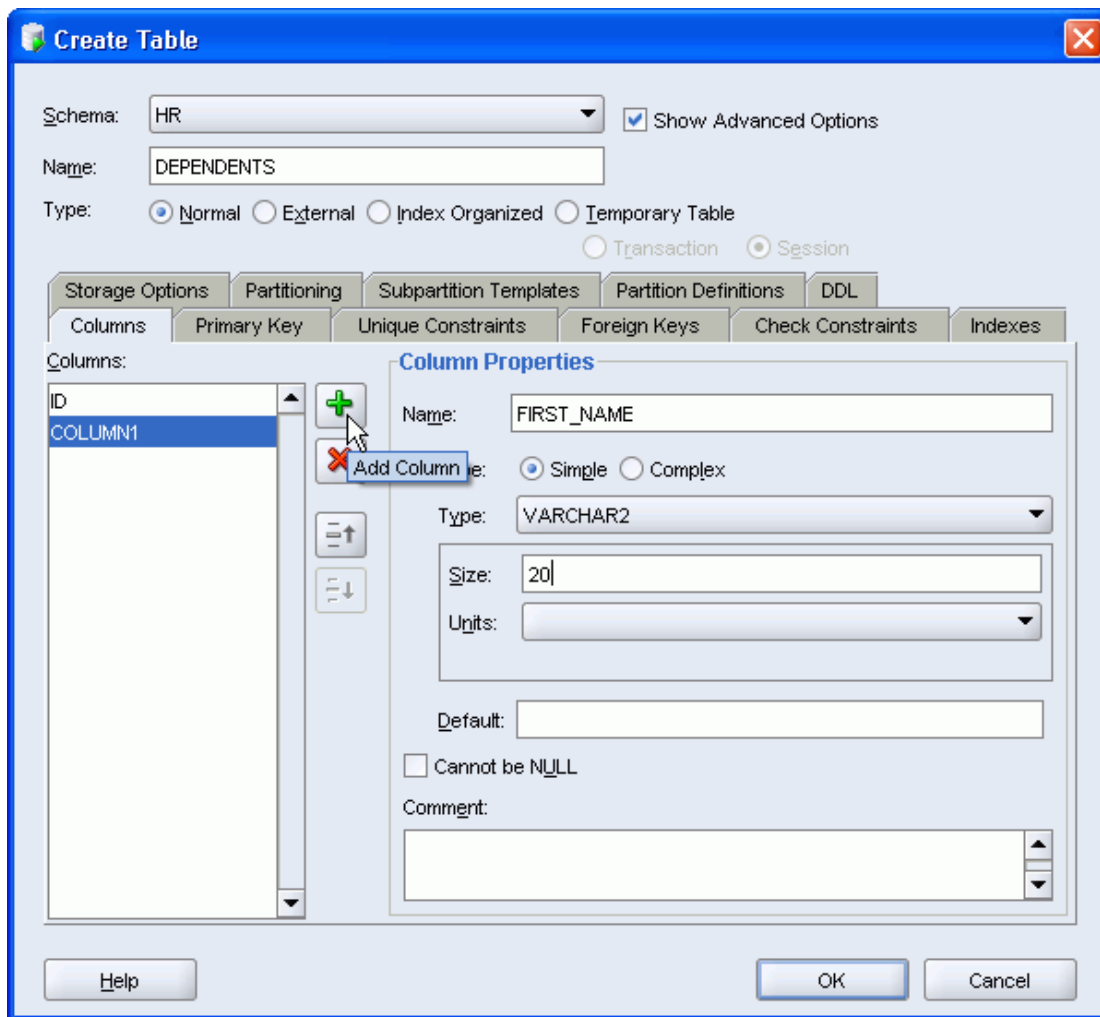
Column Name	Type	Size	Not Null	Primary Key
COLUMN1	VARCHAR2	4000	<input type="checkbox"/>	<input type="checkbox"/>

Below the table are 'Add Column' and 'Remove Column' buttons. At the bottom of the dialog are 'Help', 'OK', and 'Cancel' buttons.

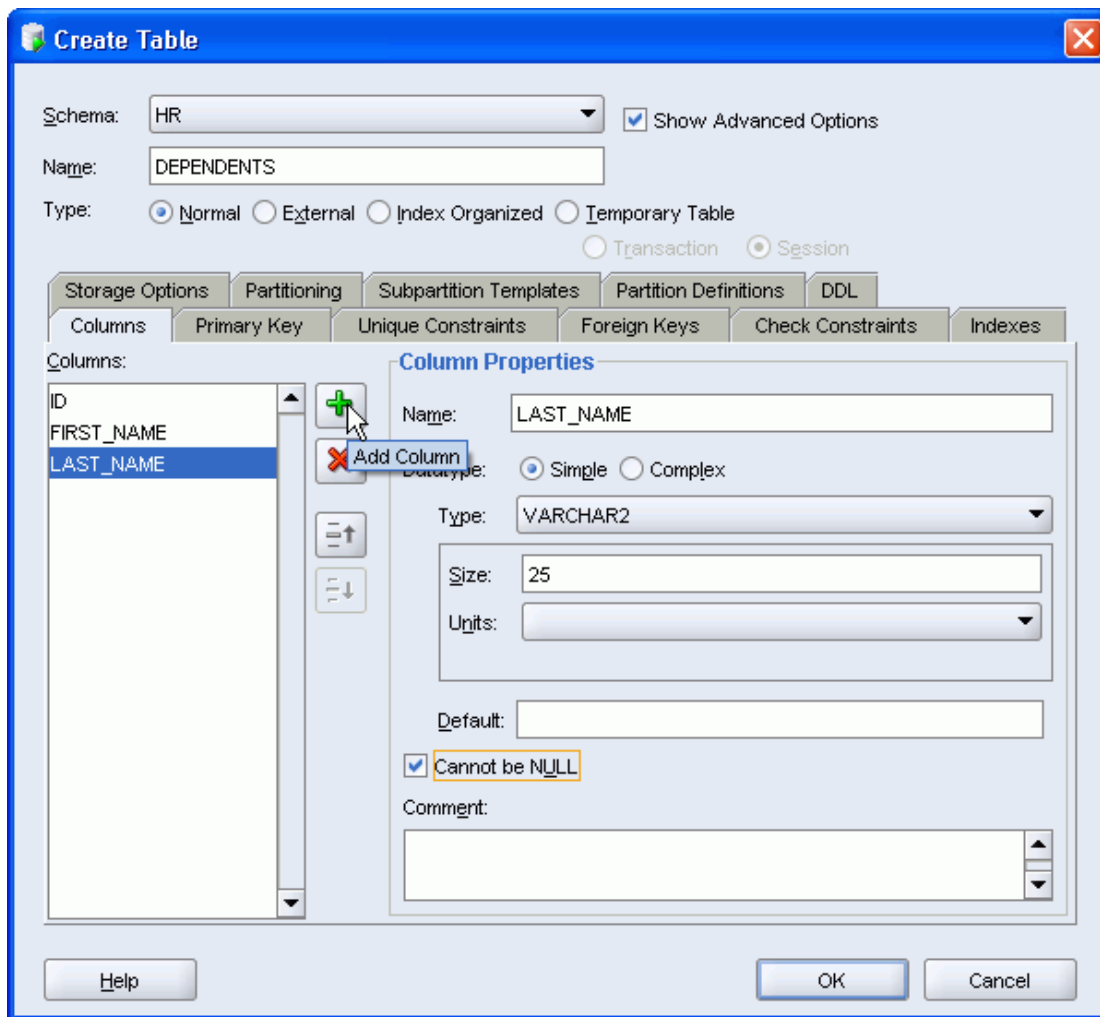
3. 输入 ID 作为 Name，选择 NUMBER 作为数据类型，并输入 6 作为 Precision。选择 Cannot be NULL 复选框。然后单击 Add Column 图标。



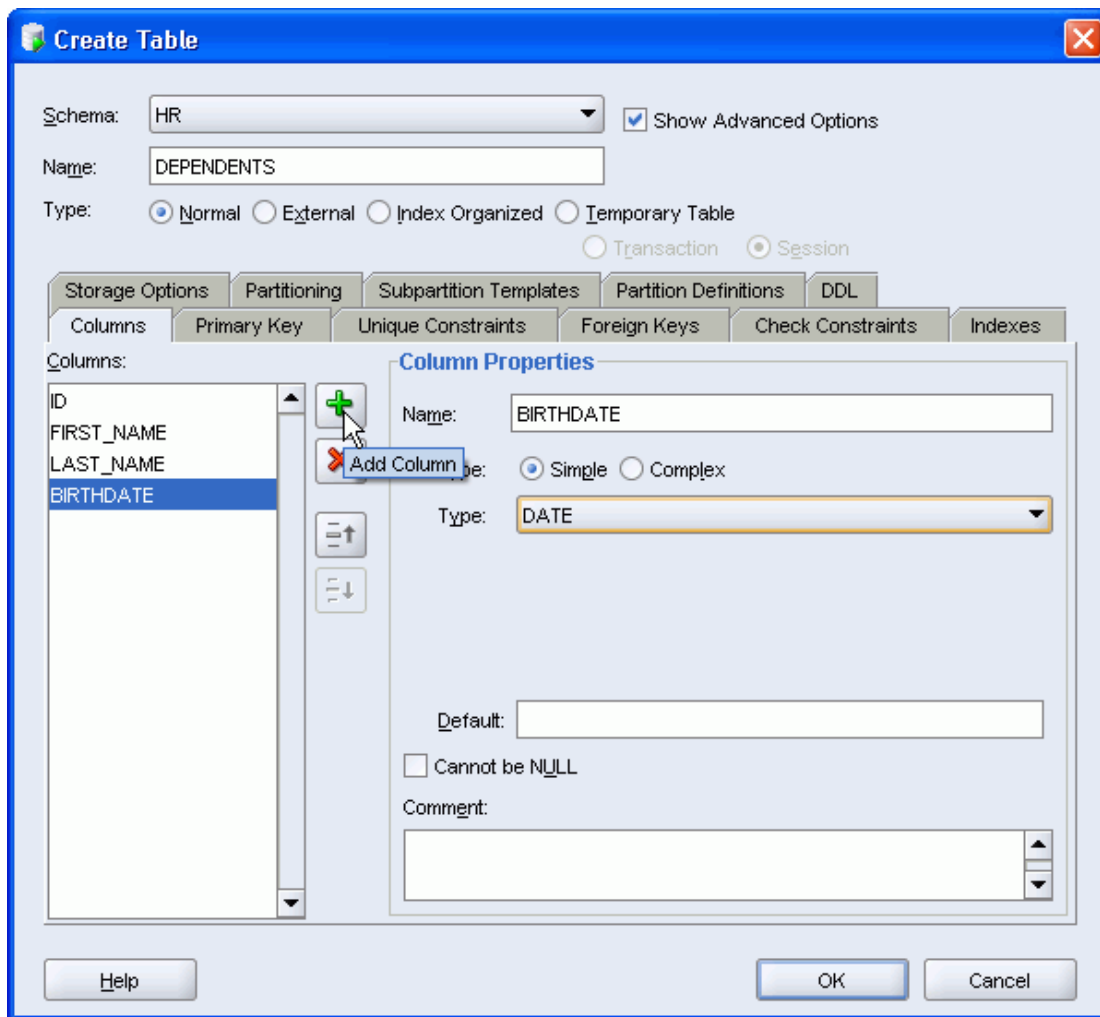
4. 输入 FIRST_NAME 作为 Name 并输入 20 作为 Size。然后单击 Add Column 图标。



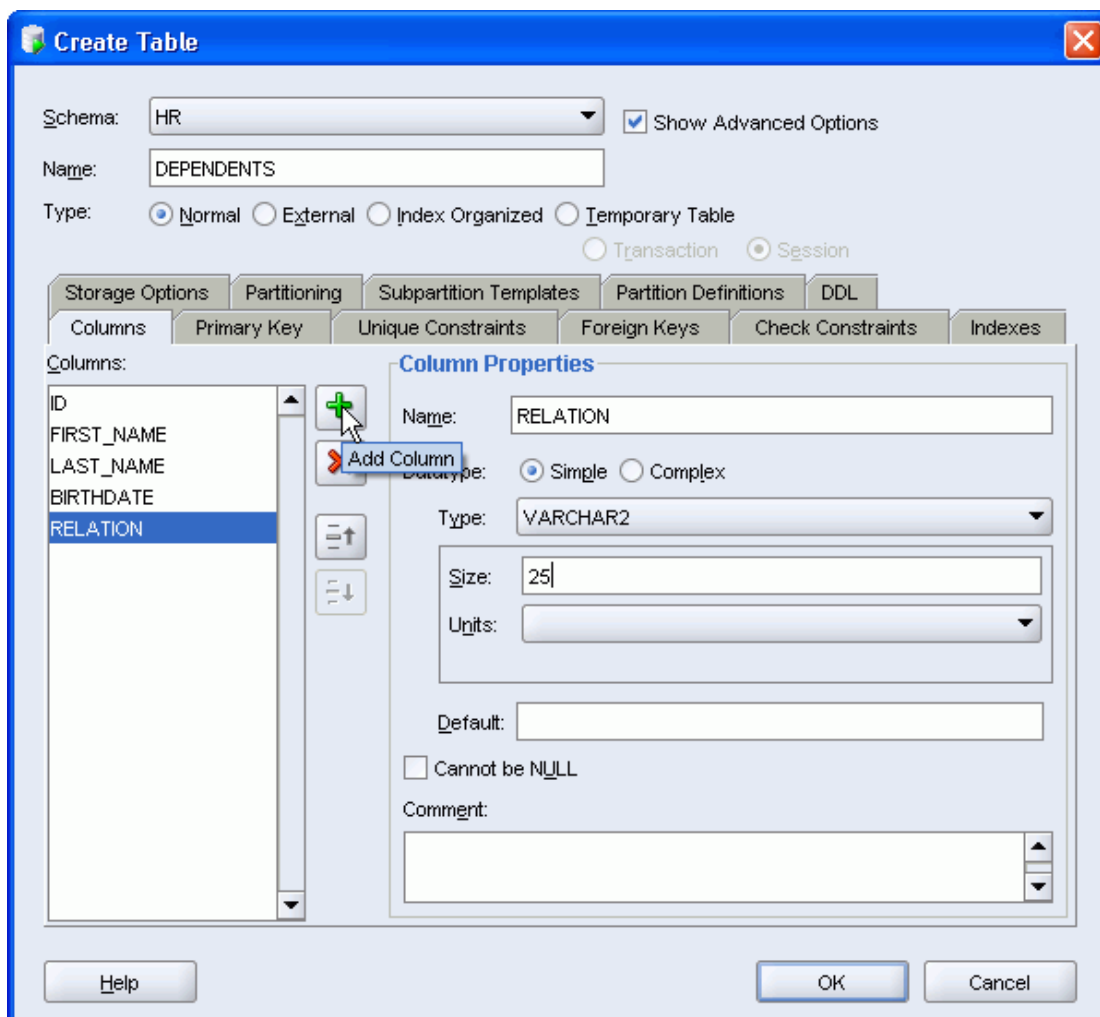
5. 输入 LAST_NAME 作为 Name 并输入 25 作为 Size。选择 Cannot be NULL 复选框。然后单击 Add Column 图标。



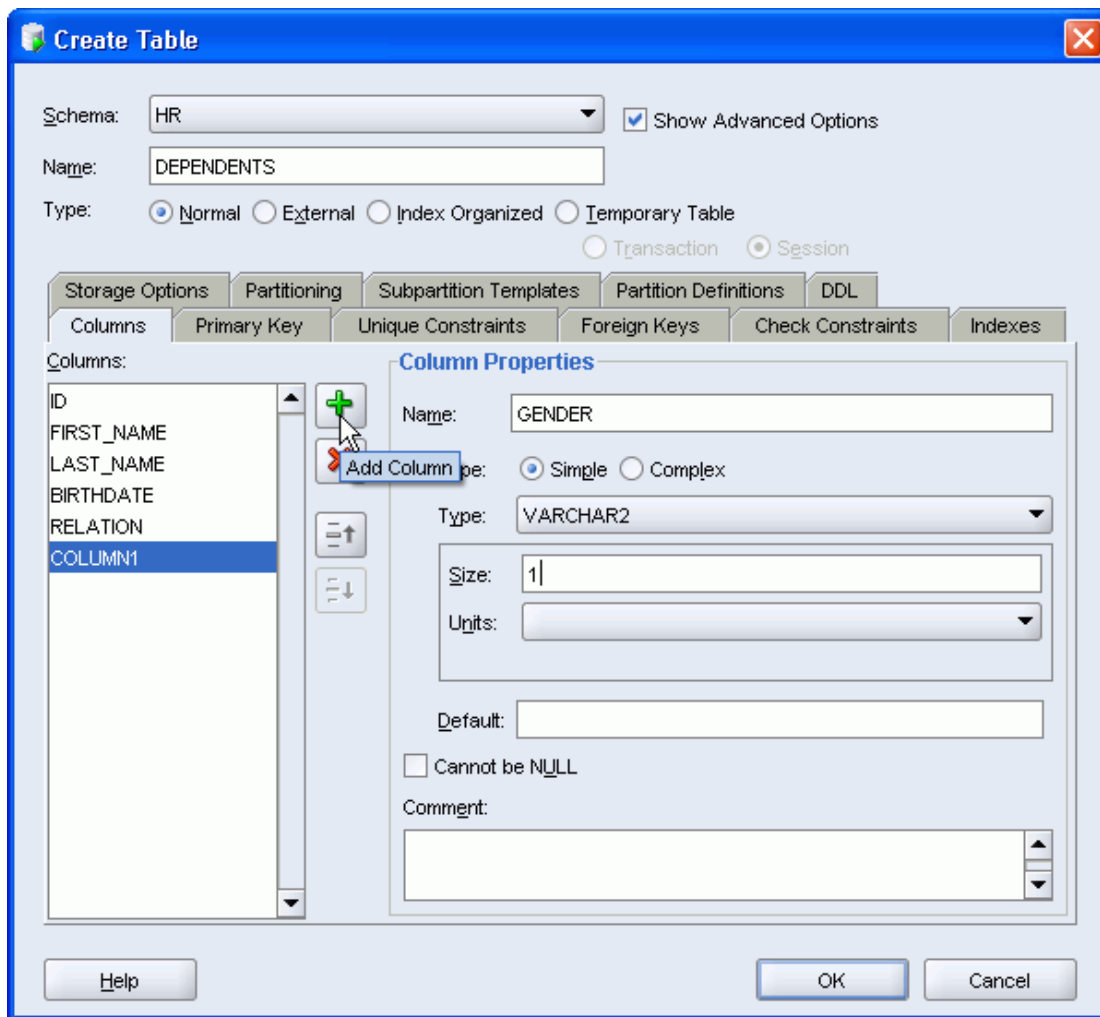
6. 输入 BIRTHDATE 作为名称，然后选择 DATE 作为数据类型。然后单击 Add Column 图标。



7. 输入 RELATION 作为 Name，并输入 25 作为 Size。然后单击 Add Column 图标。



8. 输入 GENDER 作为 Name 并输入 1 作为 Size。然后单击 Add Column 图标。



9. 输入 `RELATIVE_ID` 作为 Name，选择 `NUMBER` 作为数据类型，并输入 6 作为 Precision。选择 `Cannot be NULL` 复选框。然后单击 `OK` 创建该表。

Create Table

Schema: ☒ Show Advanced Options

Name:

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

Storage Options Partitioning Subpartition Templates Partition Definitions DDL

Columns Primary Key Unique Constraints Foreign Keys Check Constraints Indexes

Columns:

- ID
- FIRST_NAME
- LAST_NAME
- BIRTHDATE
- RELATION
- GENDER
- RELATIVE_ID

Column Properties

Name:

Datatype: ☒ Simple ☐ Complex

Type:

Precision:

Scale:

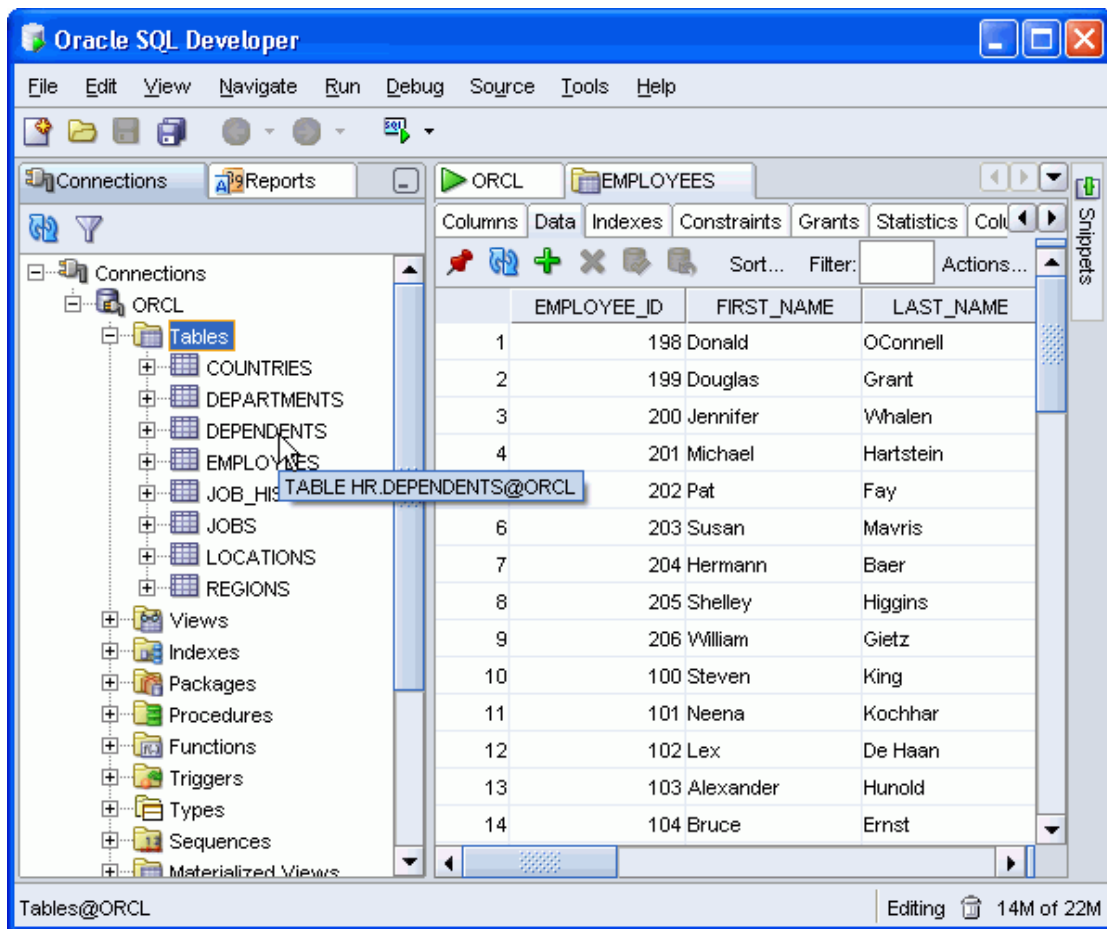
Default:

☒ Cannot be NULL

Comment:

Help OK Cancel

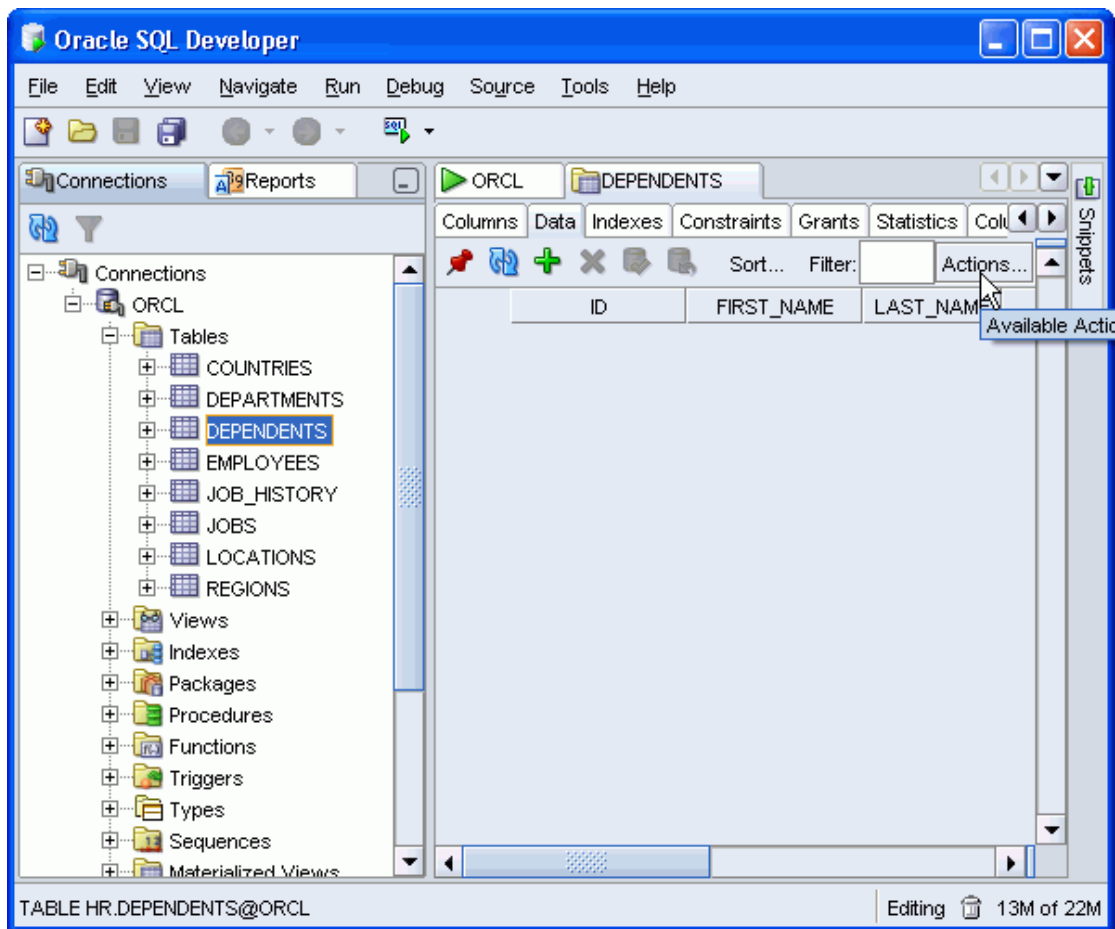
10. 新表出现在表列表中。从列表中选择 DEPENDENTS。



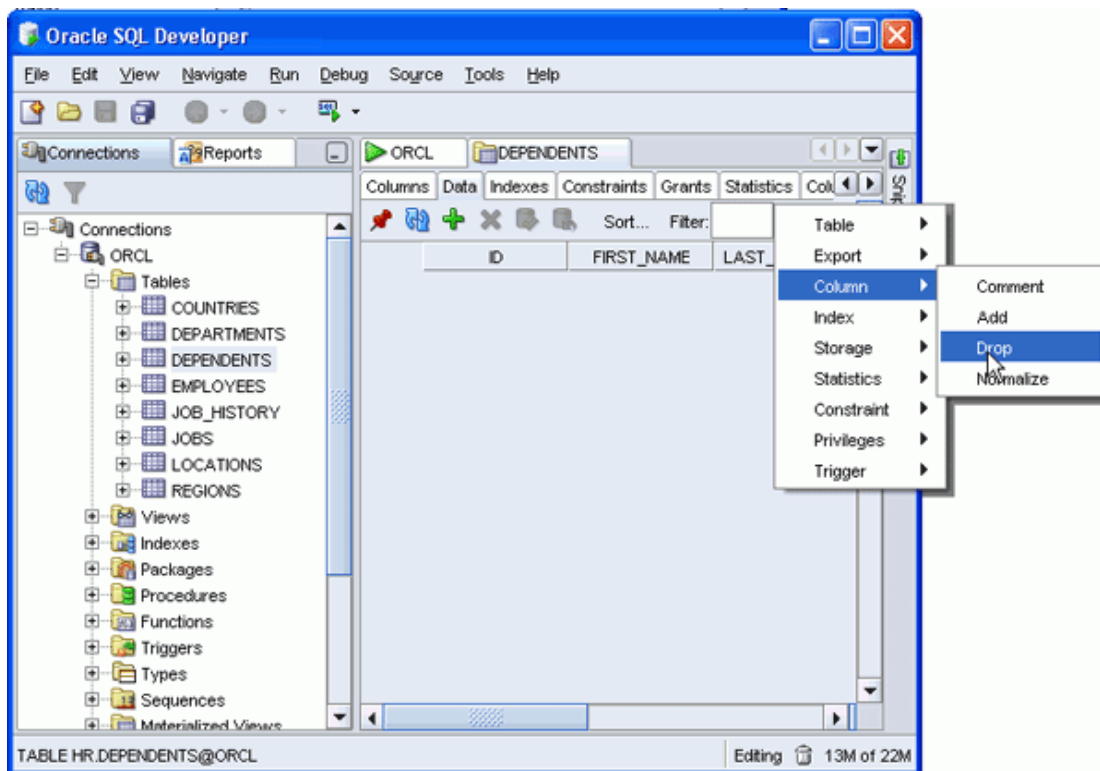
更改表定义

Oracle SQL Developer 使得更改数据库对象变得非常简单。在刚创建的 DEPENDENTS 表中删除一列。执行以下步骤：

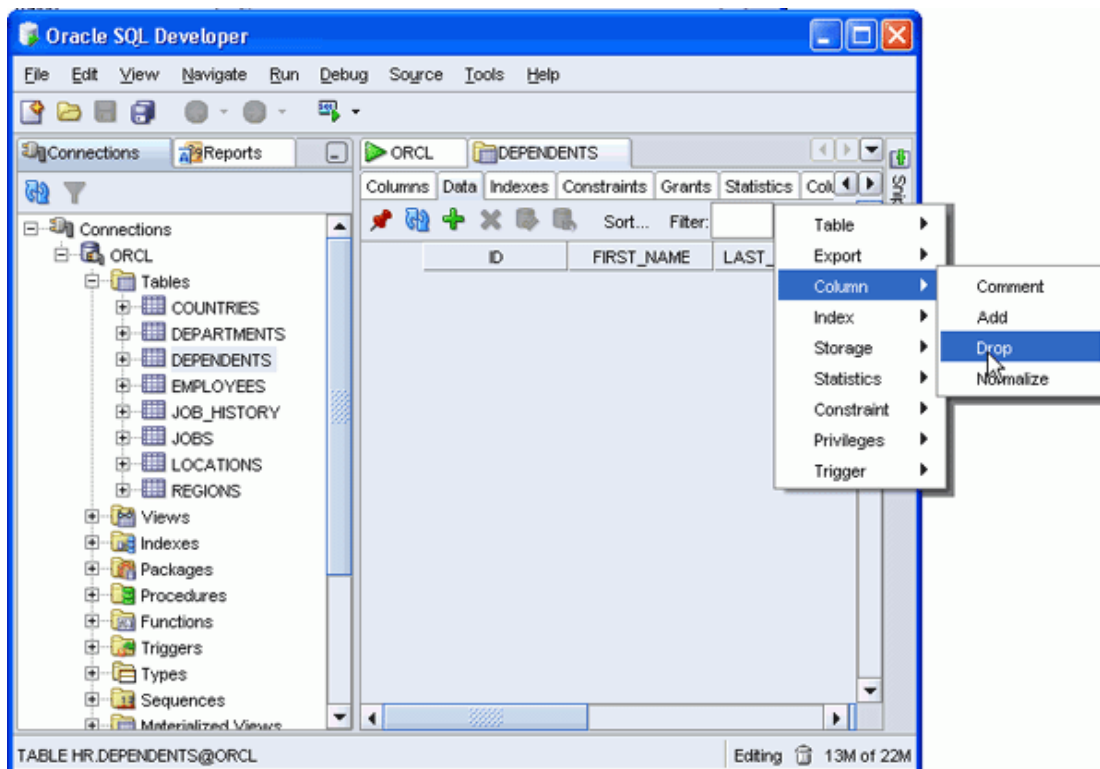
1. 单击 Actions... 按钮。



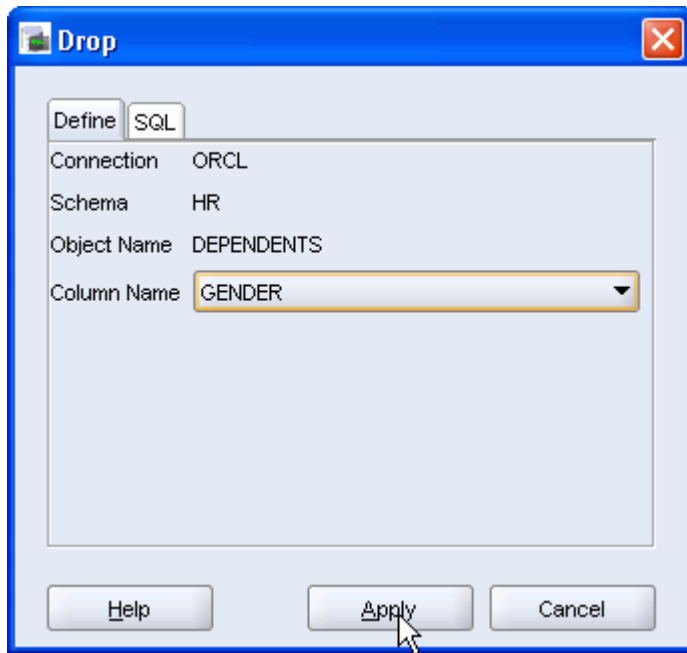
2. 选择 Column，然后选择 Drop。



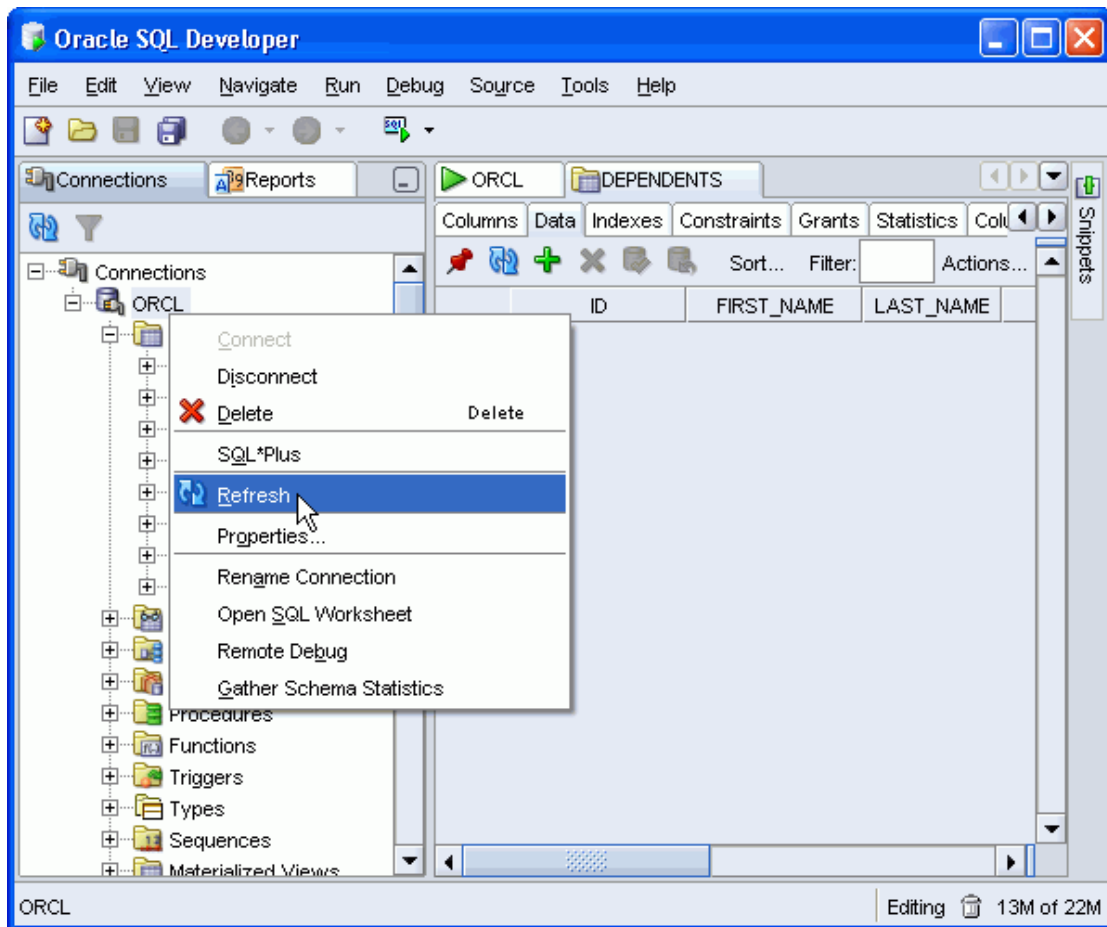
3. 选择 Column Name GENDER 并单击 Apply。



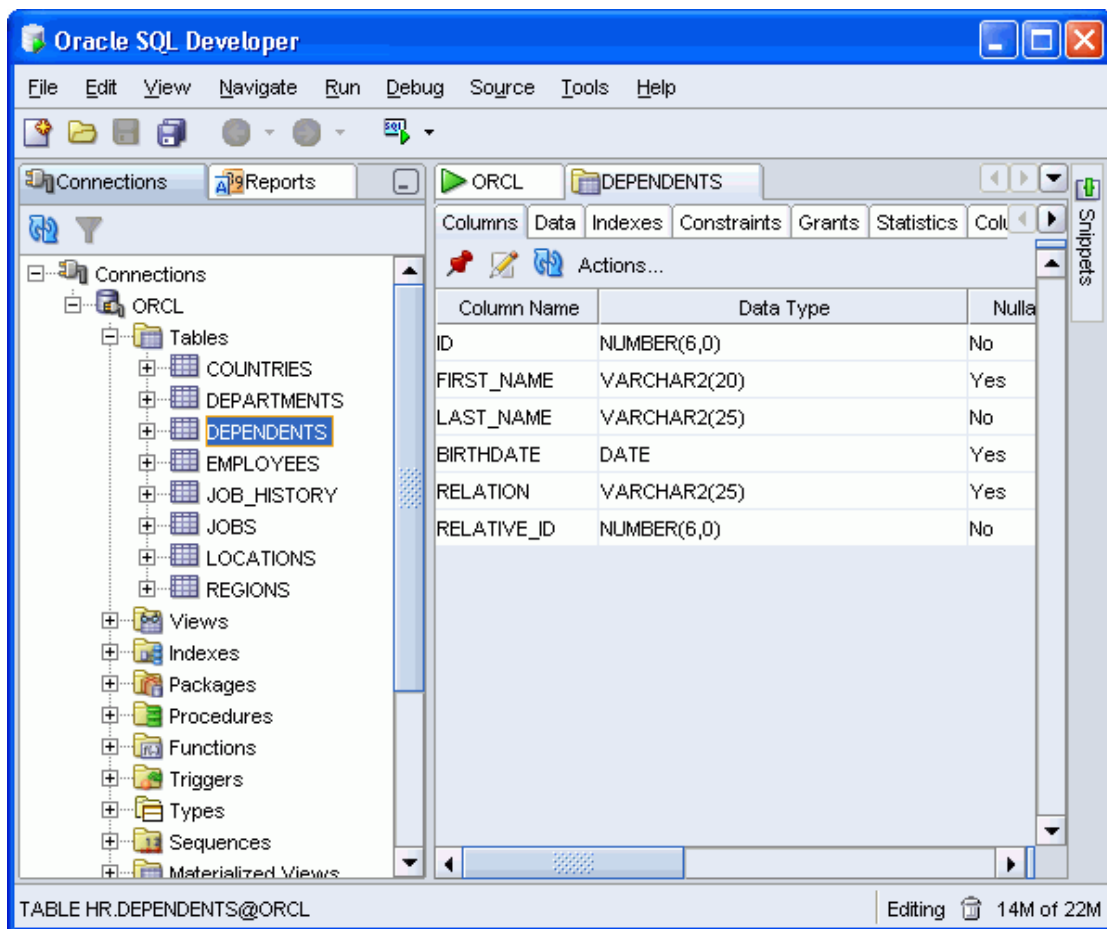
4. 该列已经删除。单击 OK。



5. 右键单击 ORCL 数据库连接并单击 Refresh，以刷新数据库连接。



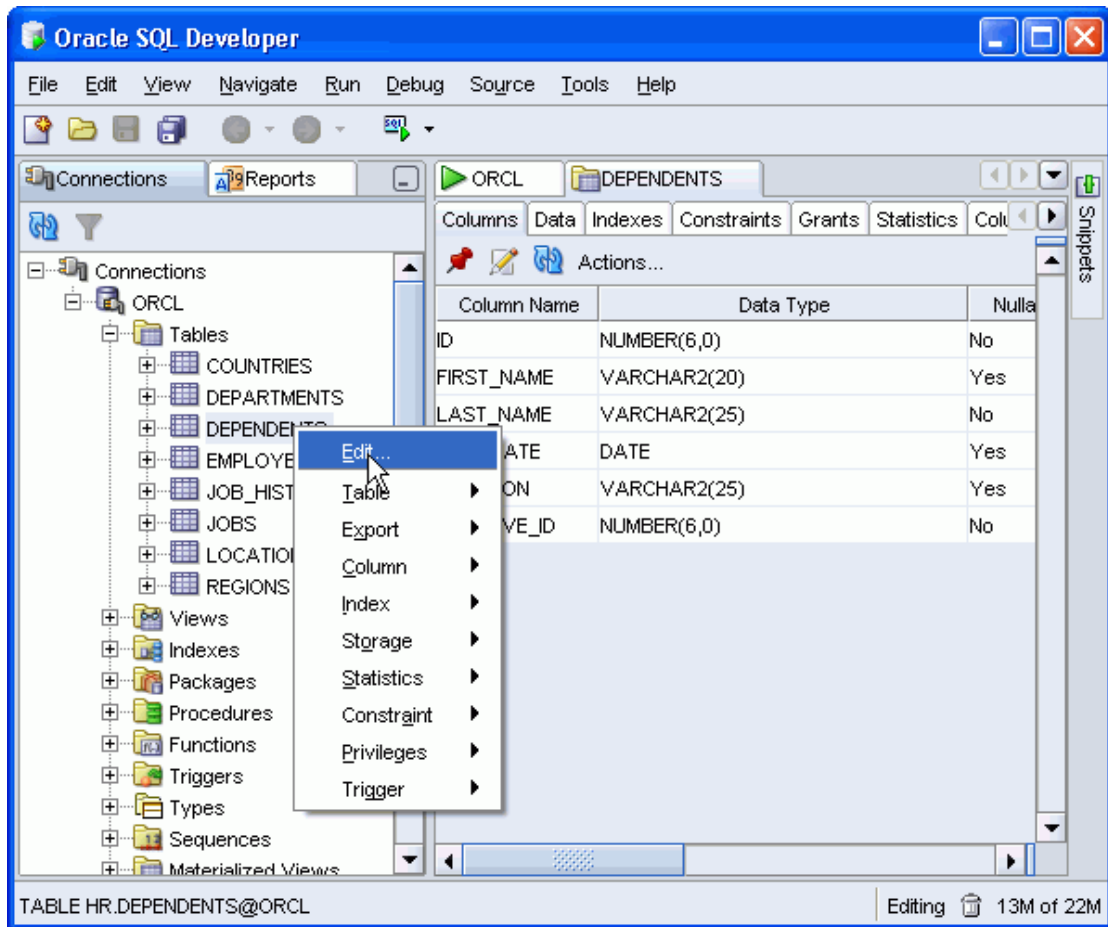
6. 展开 ORCL > Tables。然后选择 DEPENDENTS 表。



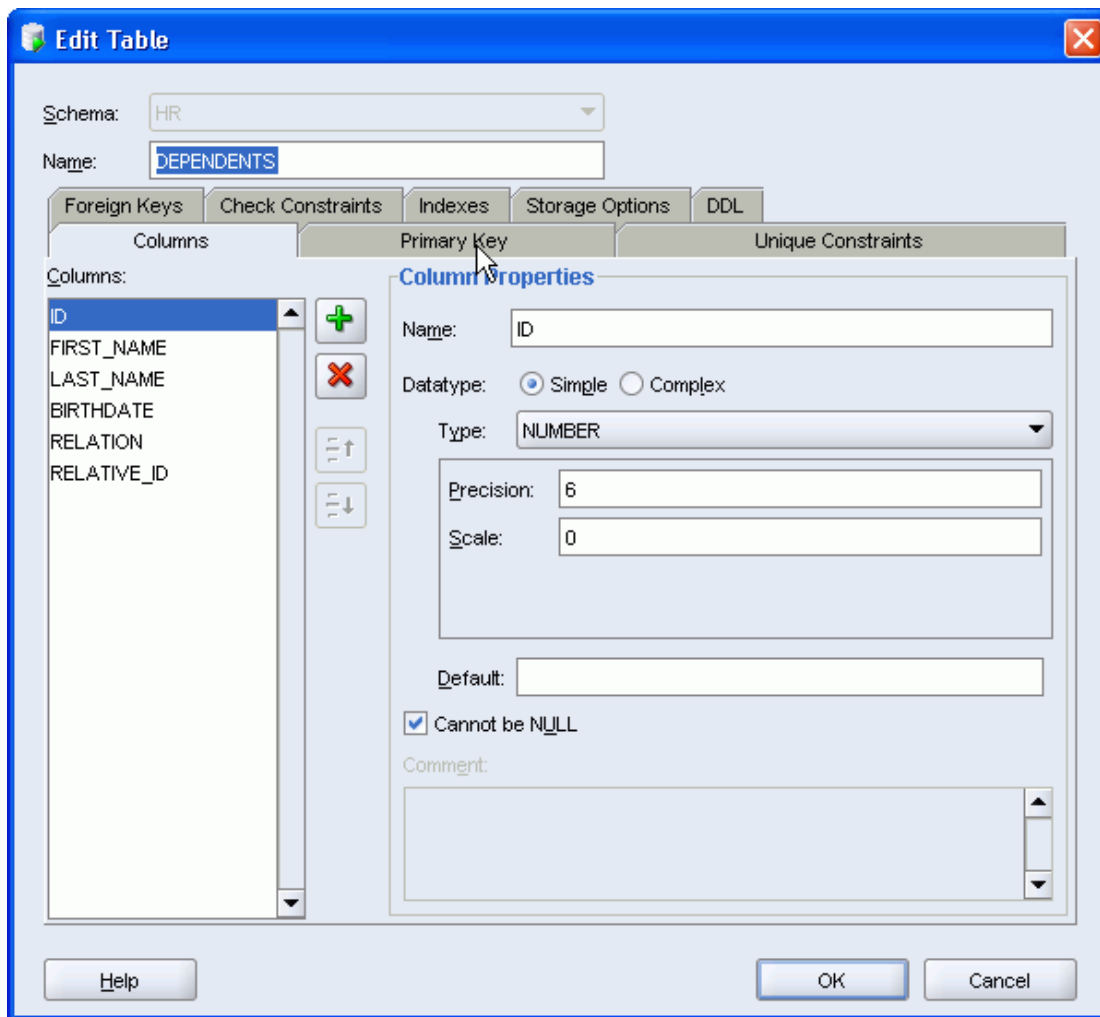
添加表约束

在这一节中，为 DEPENDENTS 表创建主键和外键约束。执行以下步骤：

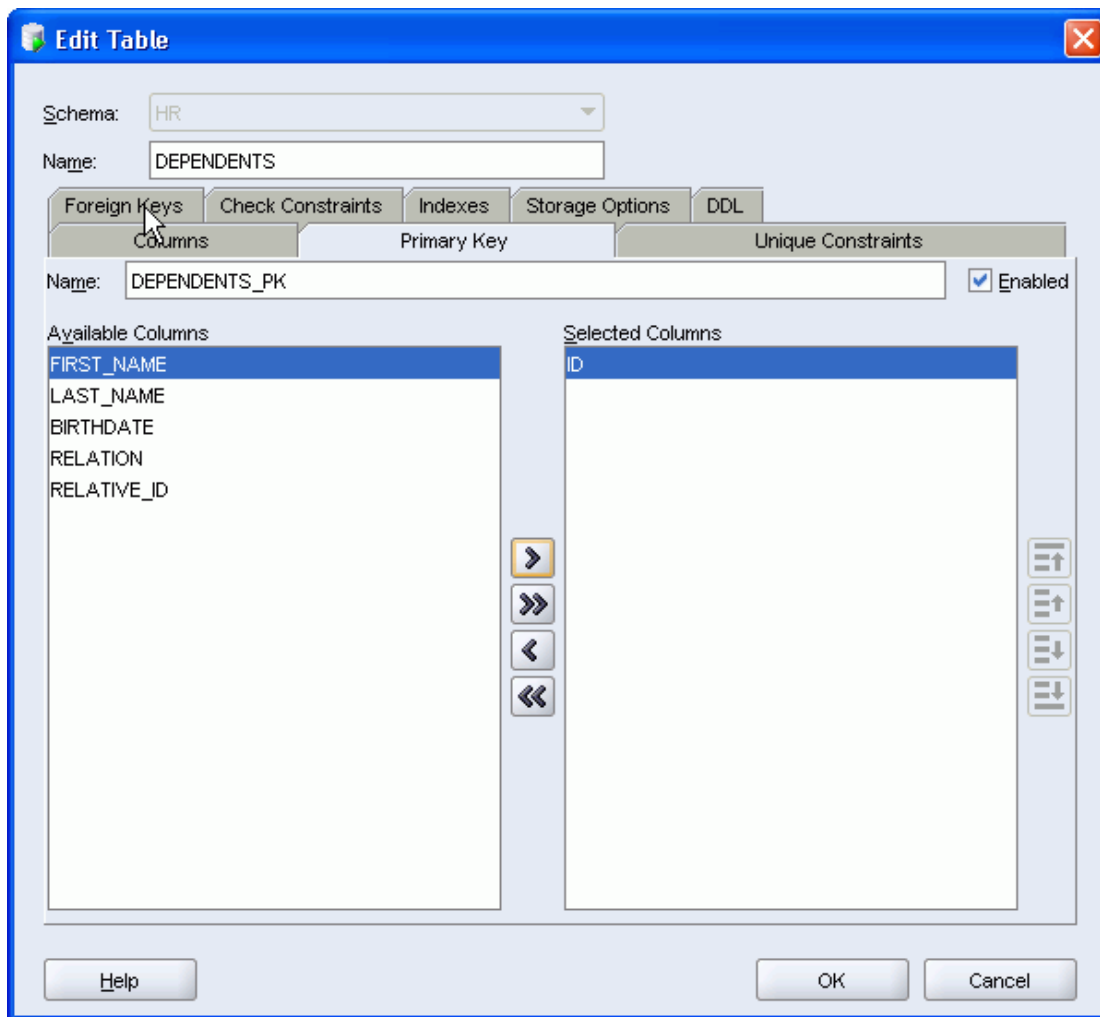
1. 右键单击 DEPENDENTS 并选择 Edit。



2. 单击 Primary Key 选项卡。



3. 选择 ID 列并单击 ➤，然后单击 Foreign Key 选项卡。



4. 单击 Add。

Edit Table

Schema:

Name:

Foreign Keys | Check Constraints | Indexes | Storage Options | DDL

Columns | Primary Key | Unique Constraints

Foreign Keys:

Name:

☐ Enabled

Referenced Schema:

Referenced Table:

Referenced Constraint:

Associations:

Local Column	Referenced Column

On Delete:

5. 选择 EMPLOYEES 作为 Referenced Table 并选择 RELATIVE_ID 作为 Local Column, 然后单击 OK。

Edit Table

Schema:

Name:

Foreign Keys | Check Constraints | Indexes | Storage Options | DDL

Columns | Primary Key | Unique Constraints

Foreign Keys:

Foreign Key	Action
DEPENDENTS_FK1	<input type="button" value="Add"/> <input type="button" value="Remove"/>

Name: ☒ Enabled

Referenced Schema:

Referenced Table:

Referenced Constraint:

Associations:

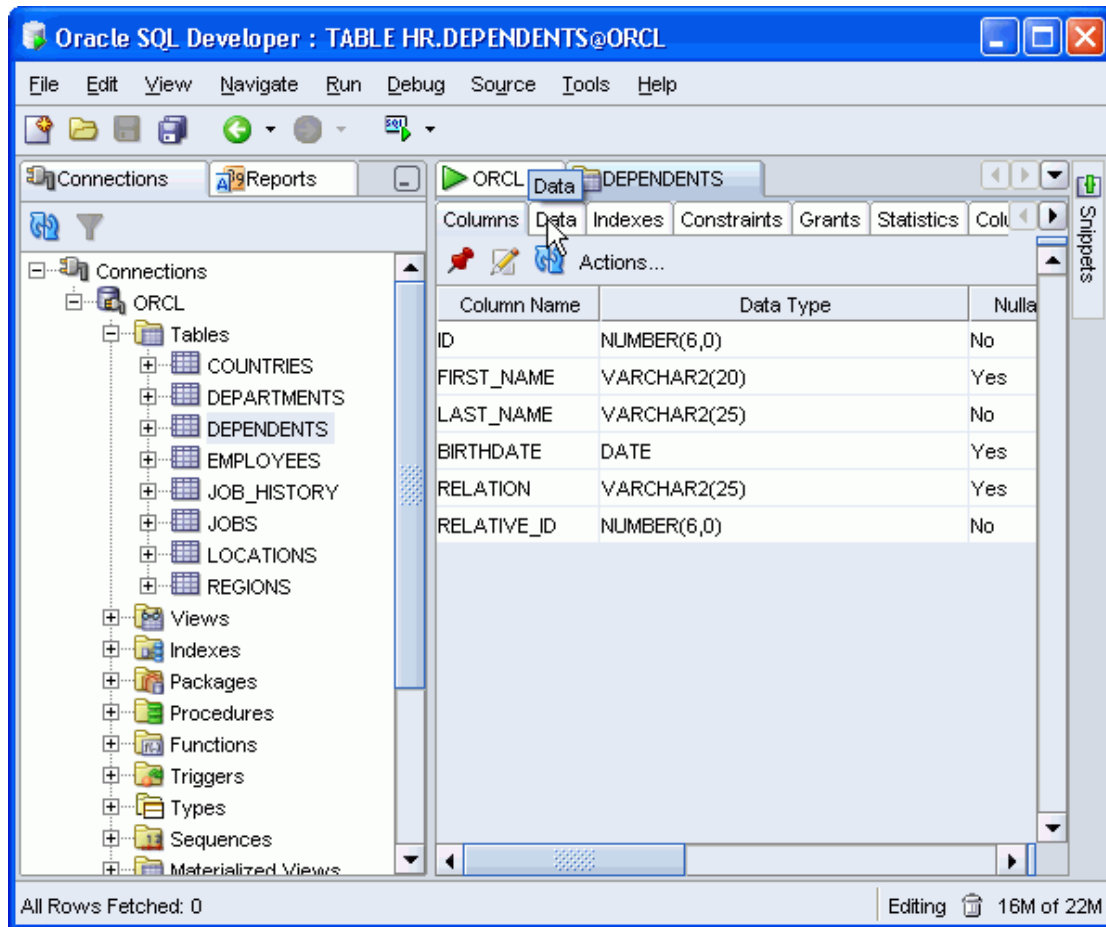
Local Column	Referenced Column on EMPLOYEES
RELATIVE_ID	EMPLOYEE_ID

On Delete:

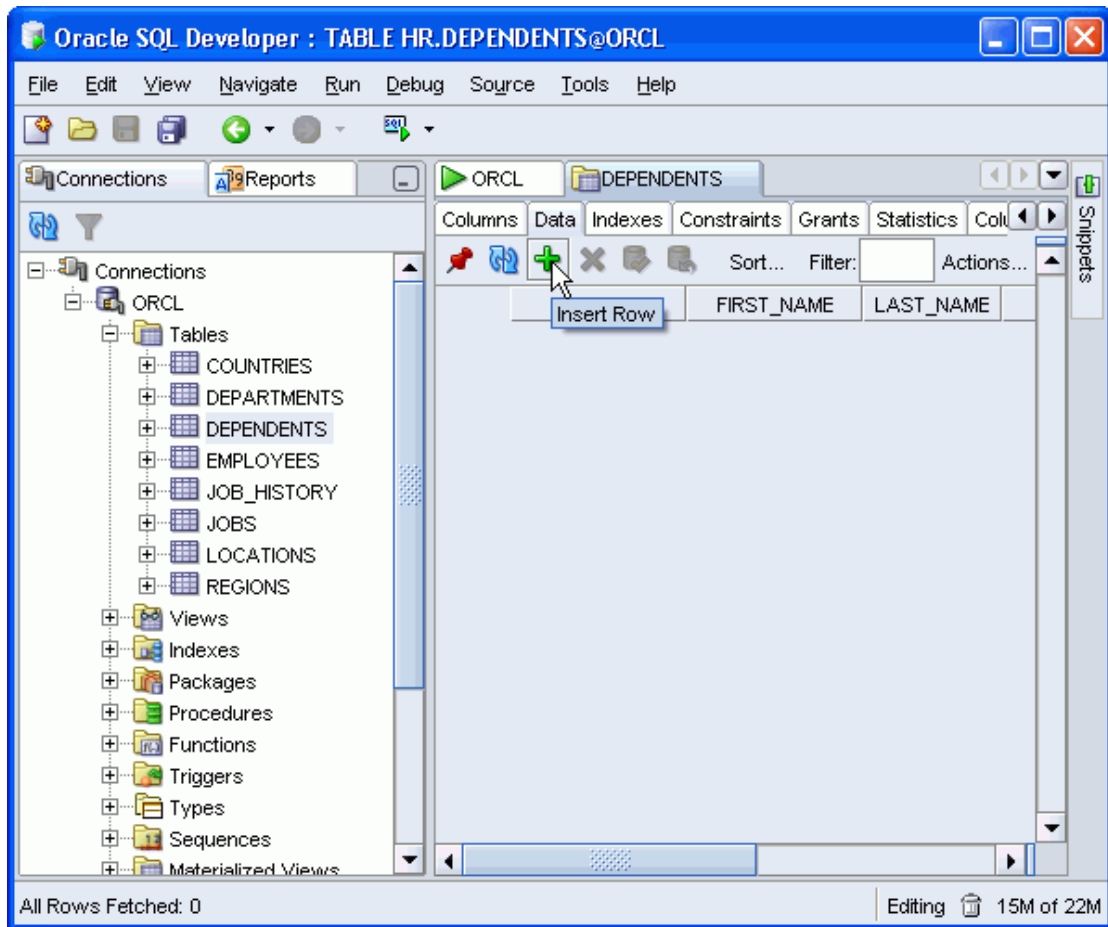
向表中添加数据

通过执行以下步骤，可以向表中添加数据：

1. 单击 Data 选项卡。

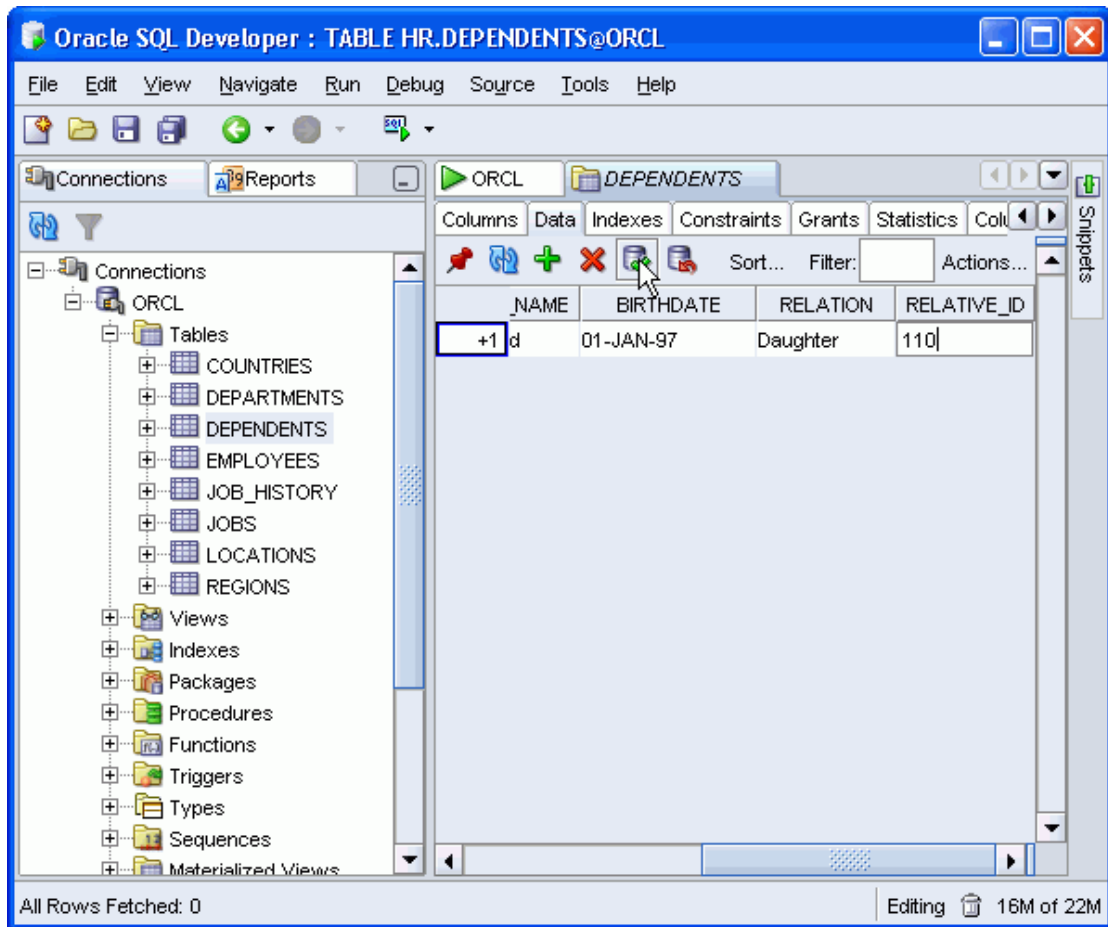


2. 然后单击 Insert Row 图标。

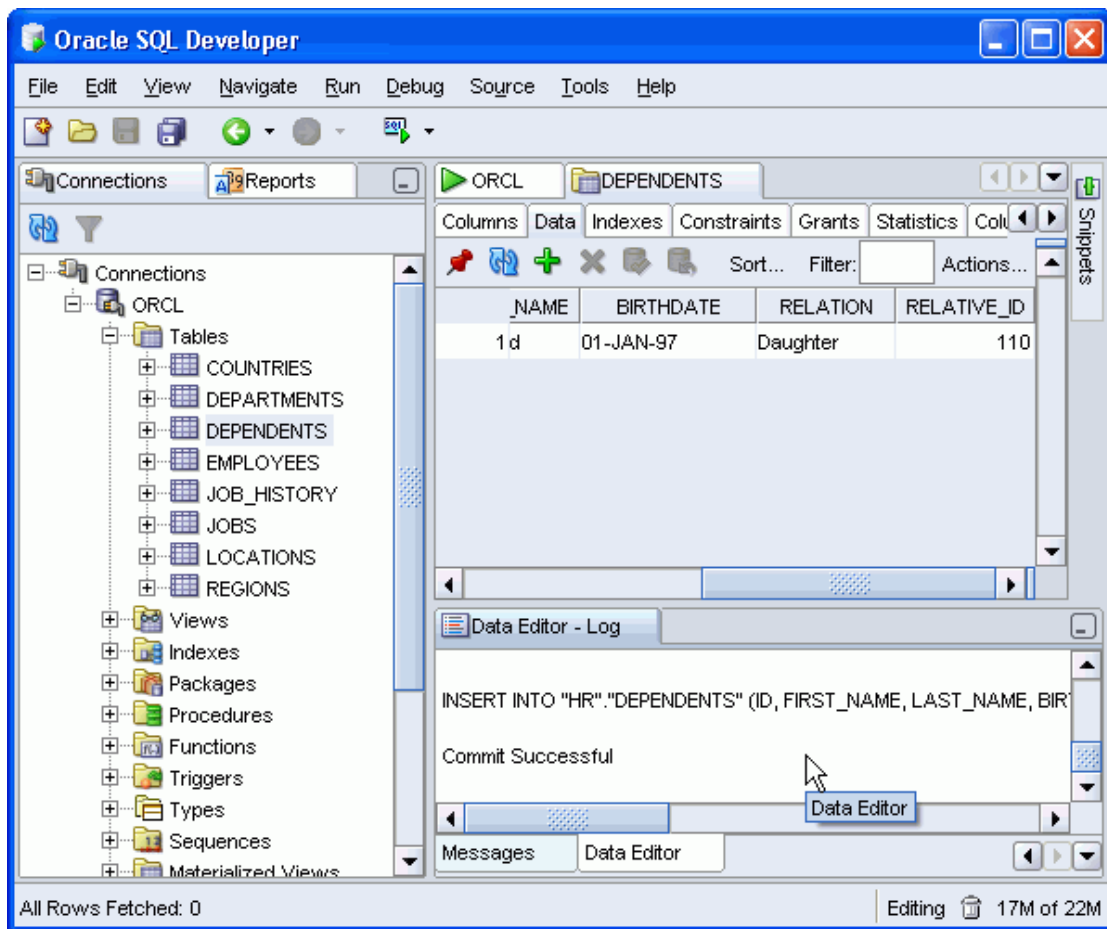


3. 输入以下数据并单击 Commit Changes 图标。

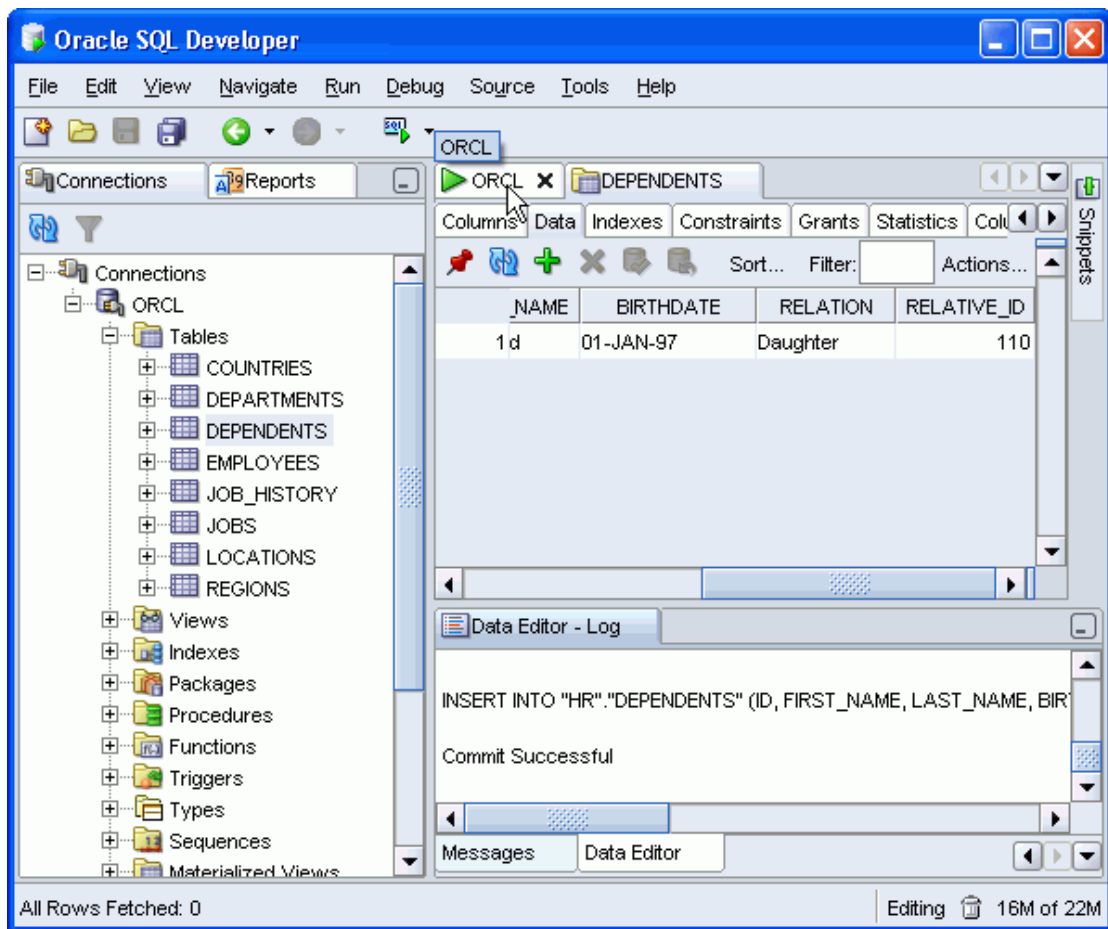
I	209
D	
FIRST_NAME	Sue
LAST_NAME	Littlefield
BIRTHDATE	2003 年 10 月 9 日
RELATION	Daughter
RELATIVE_ID	110



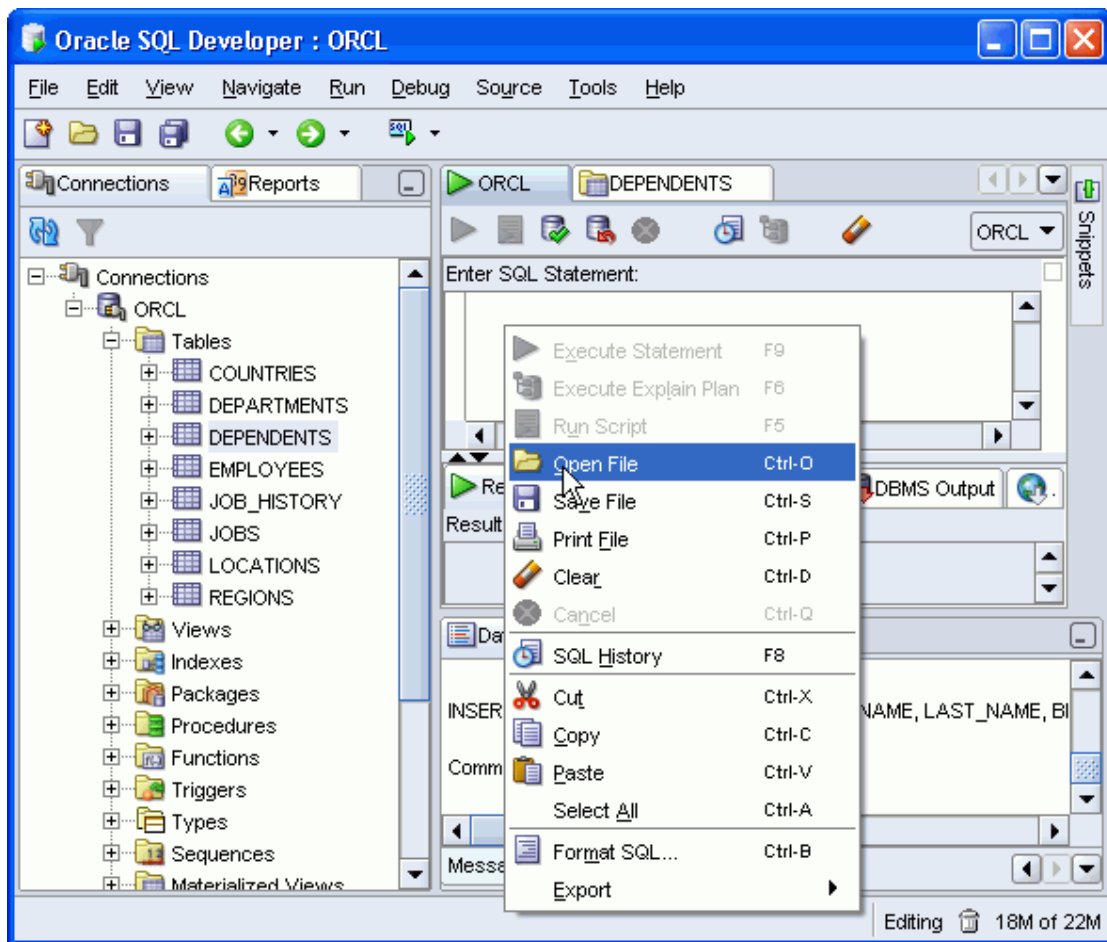
4. 刚添加的行被提交到数据库。



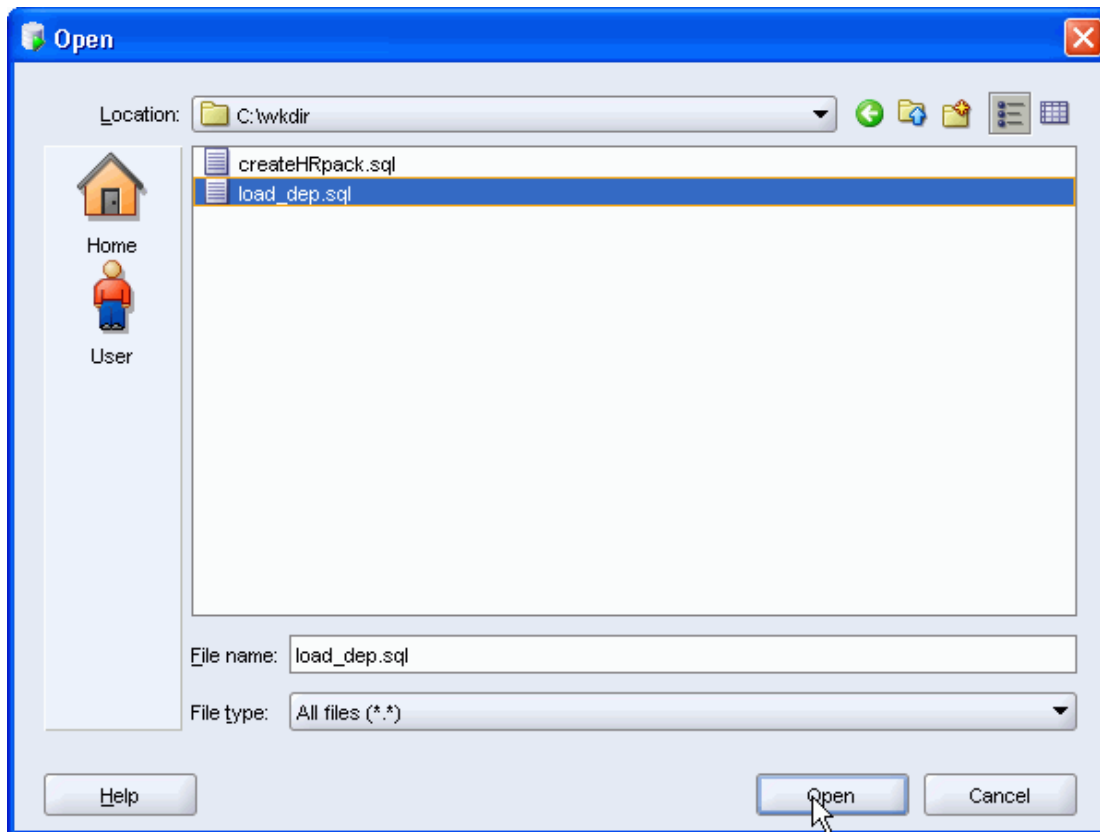
5. 还可以使用脚本一次添加多行。单击 SQL Worksheet ORCL 选项卡。



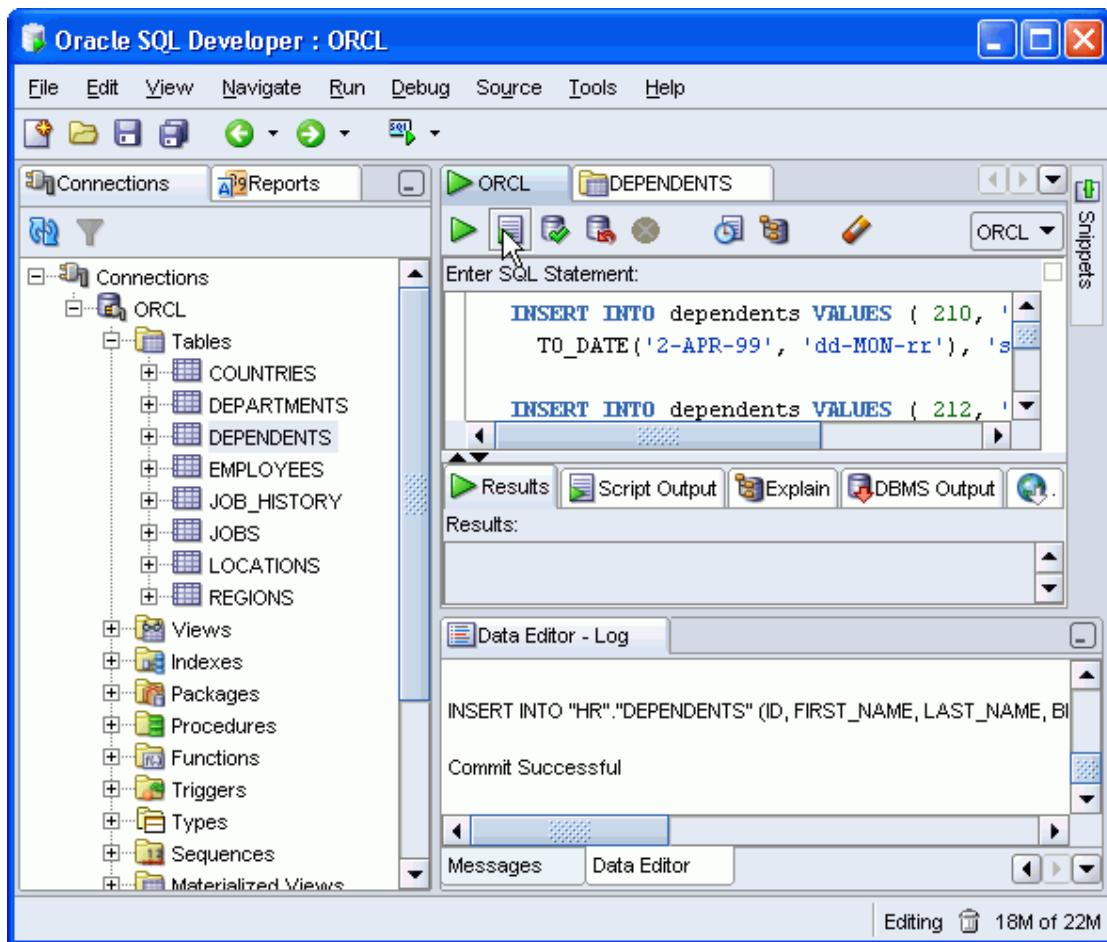
6. 右键单击 SQL Statement 区域并选择 Open File。



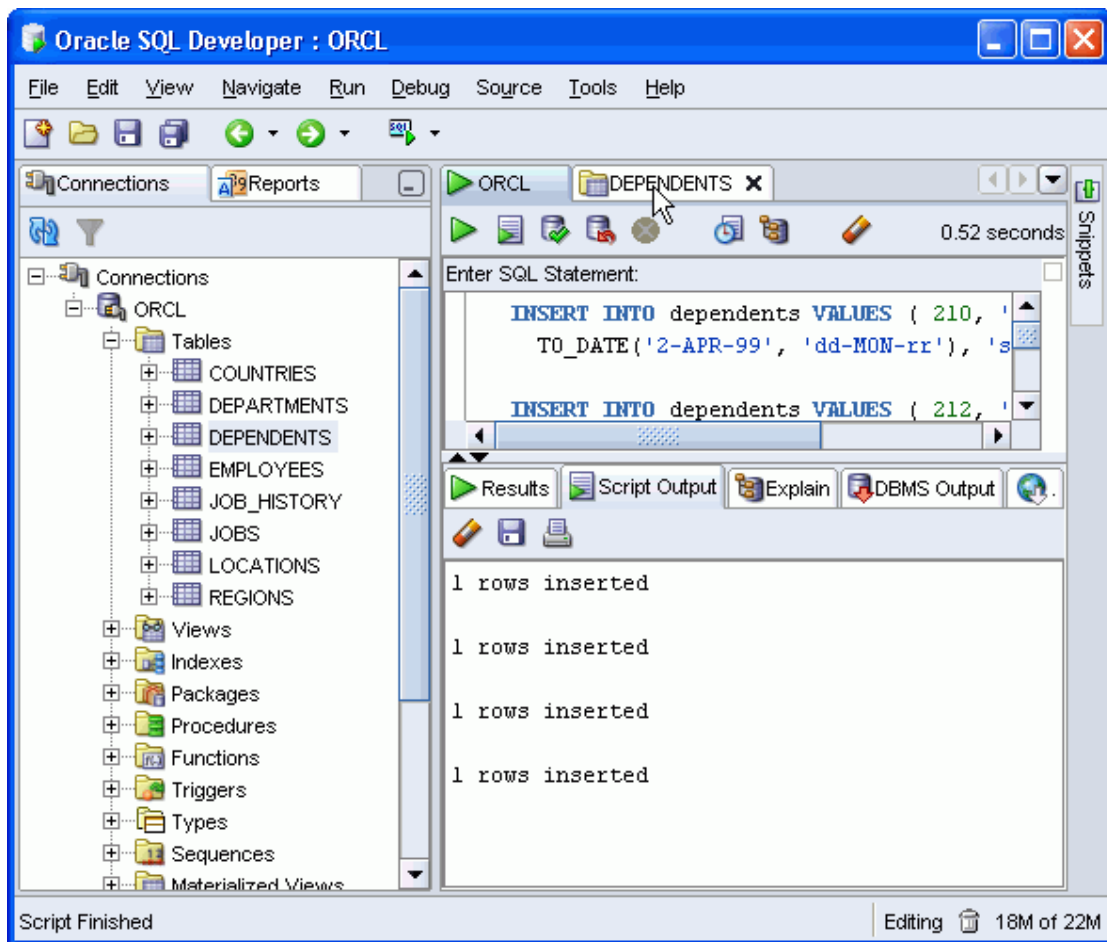
7. 导航到工作目录（例如，c:\wkdir）并选择 load_dep.sql 文件，然后单击 Open。



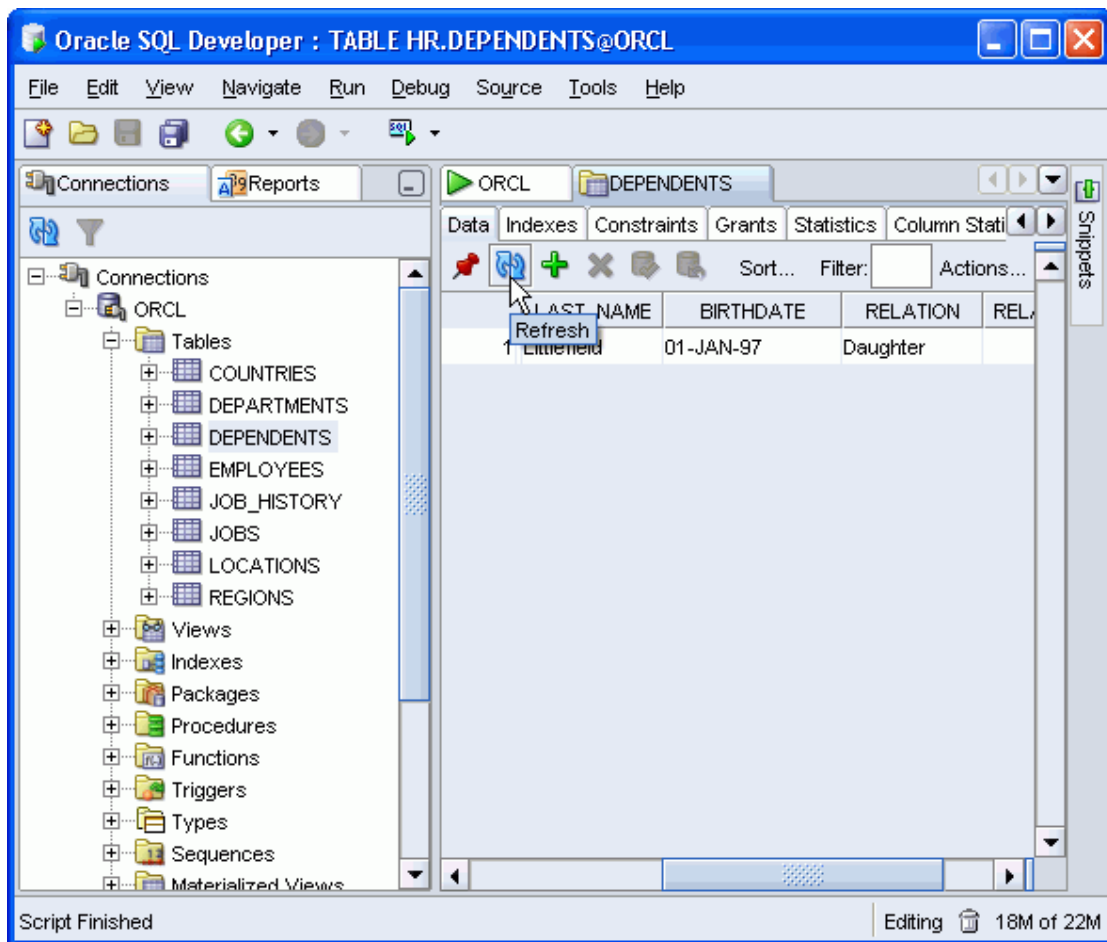
8. 随即显示该脚本的 SQL。单击 Run Script 图标。



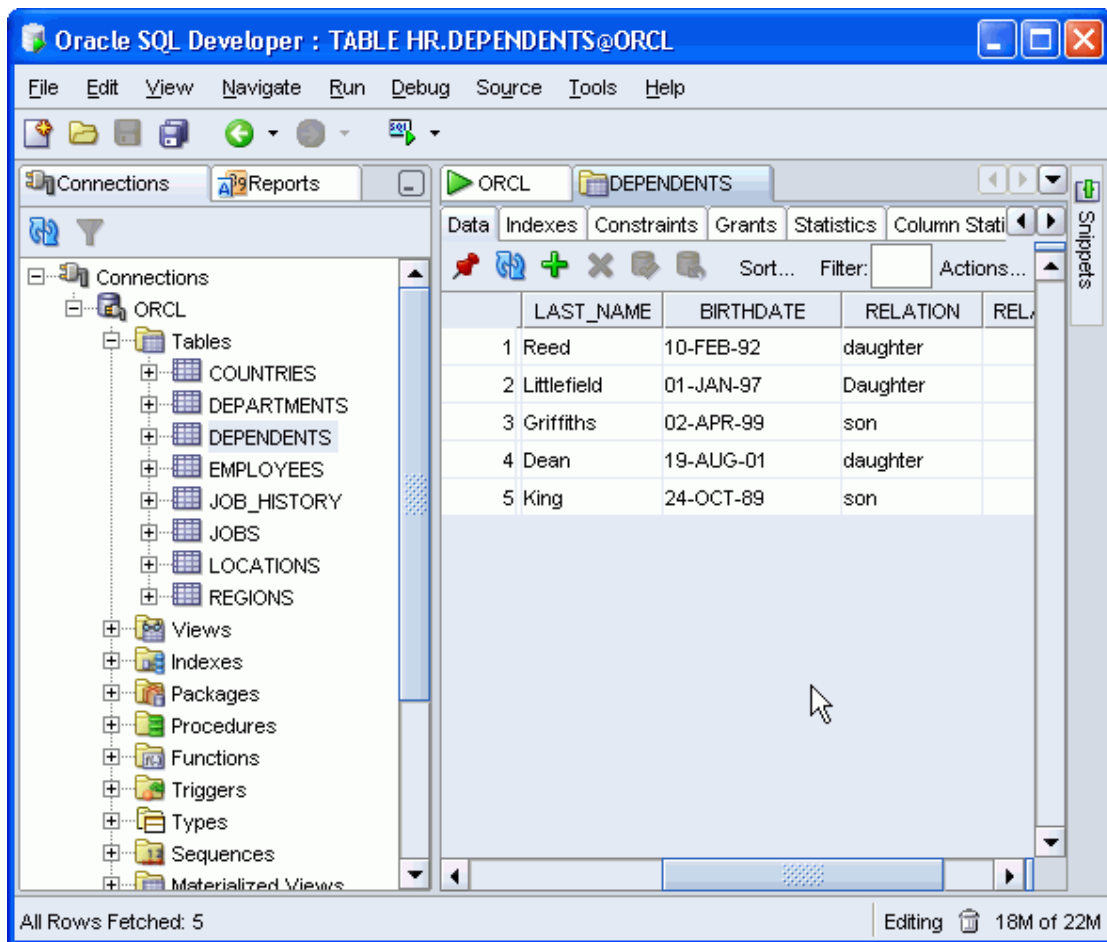
9. 该数据已插入。要查看该数据，请单击 DEPENDENTS 选项。



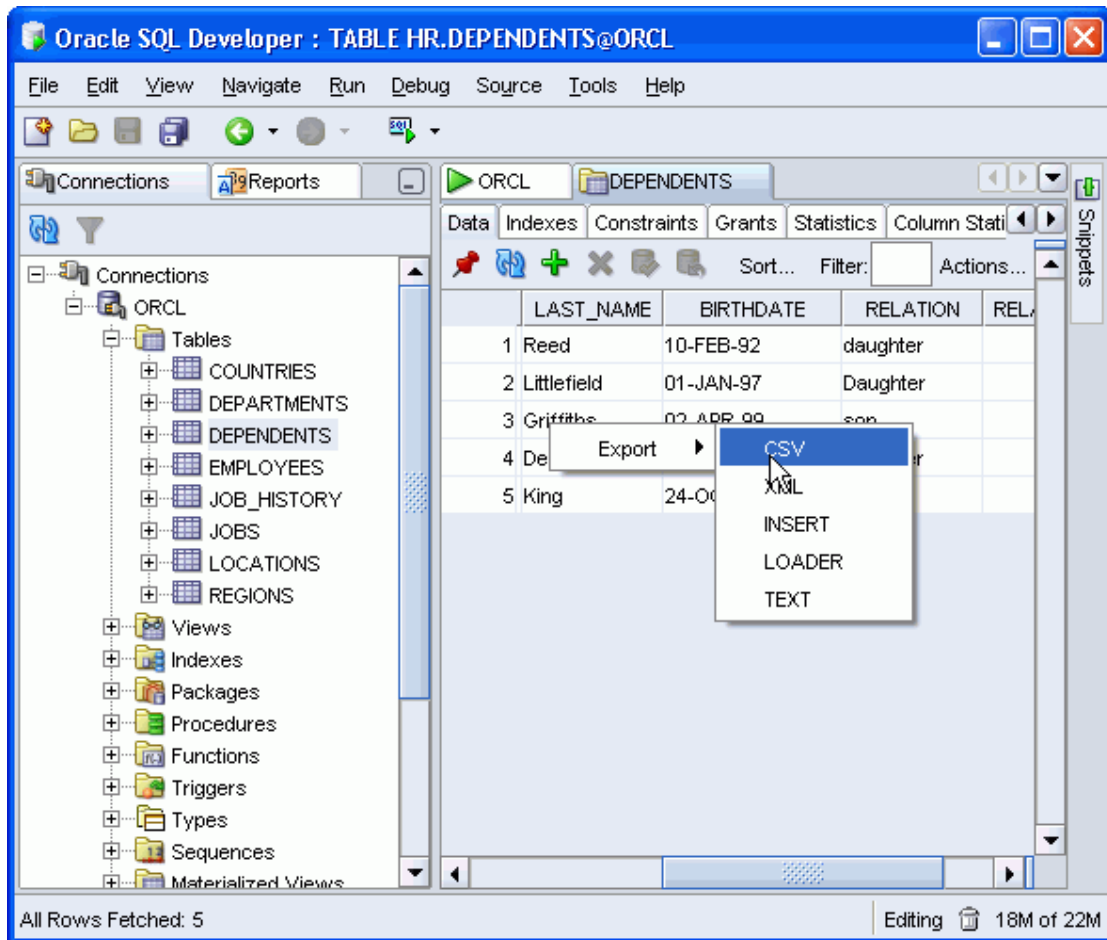
10. 单击 Refresh 显示所有数据。



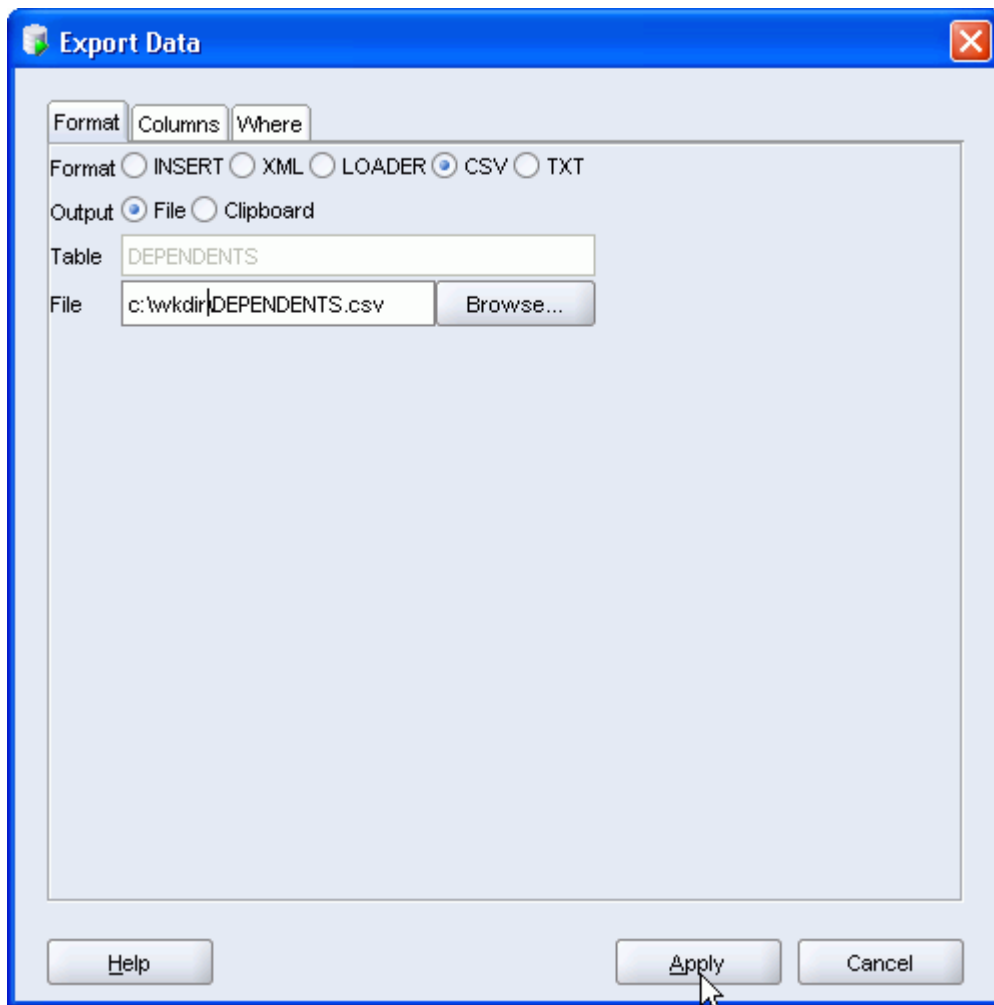
11. 系统将列出表中的所有数据。



12. 可以导出数据以便在 Excel 中使用。右键单击任何列中的一个值，选择 Export，然后选择一个文件类型。



13. 指定文件目录和名称，然后单击 Apply。



14. 如果查看 DEPENDENTS.CSV 文件，将看到以下内容。

Microsoft Excel - DEPENDENTS.csv

File Edit View Insert Format Tools Data Window Help Adobe PDF

A1 = ID

	A	B	C	D	E	F	G	H
1	ID	FIRST_NAME	LAST_NAME	BIRTHDATE	RELATION	RELATIVE_ID		
2	212	Jill	Reed	10-Feb-92	daughter	110		
3	209	Sue	Littlefield	1-Jan-97	Daughter	110		
4	210	David	Griffiths	2-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								
12								
13								

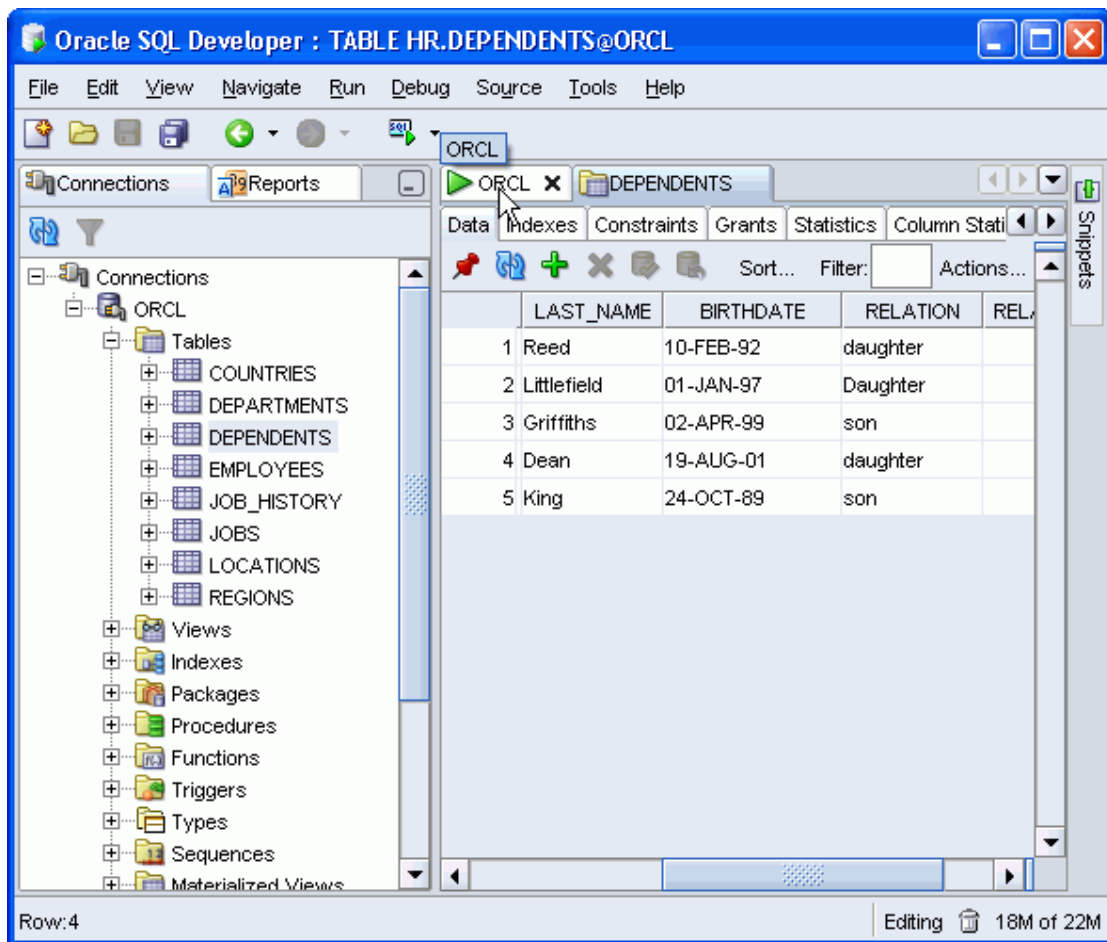
DEPENDENTS

Ready

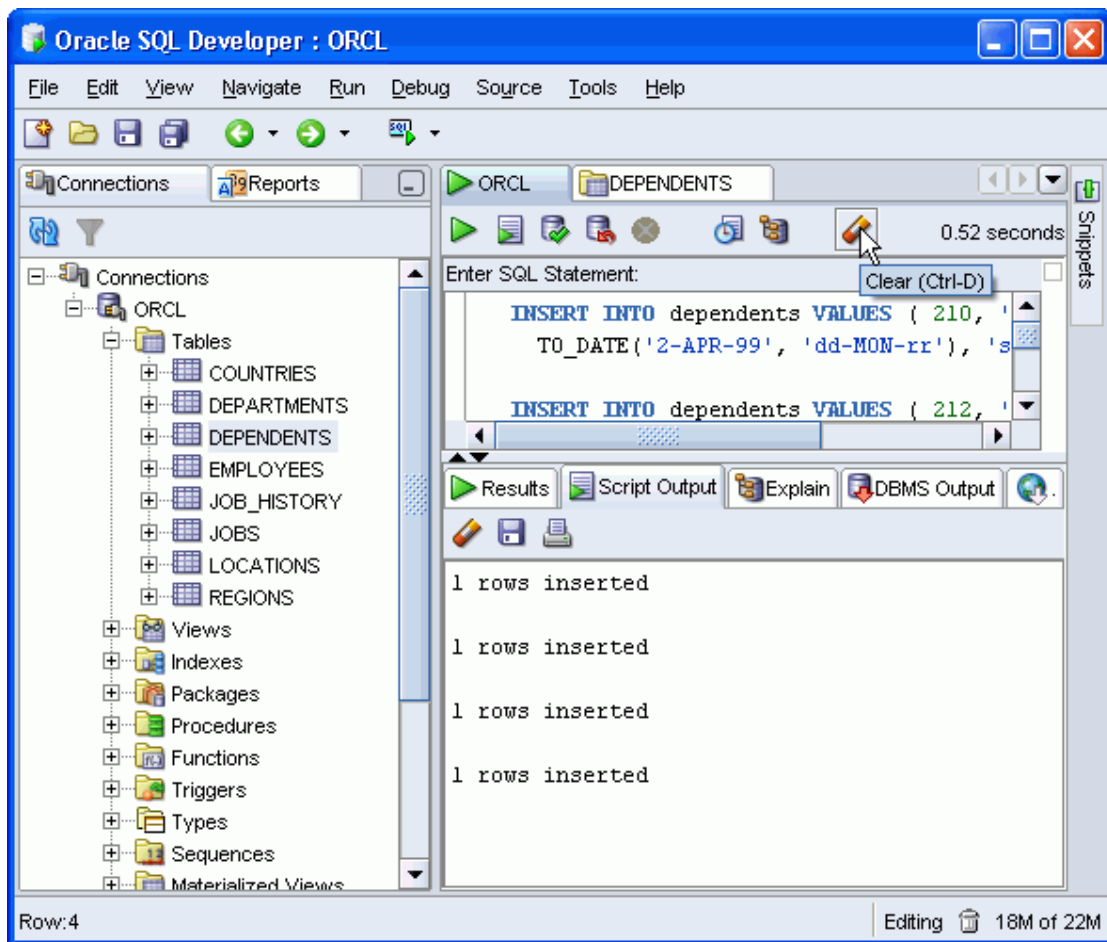
访问数据

访问 DEPENDENTS 数据的一种方式是在 DEPENDENTS 表上生成一条 SELECT 语句，然后添加一个 WHERE 子句。执行以下步骤：

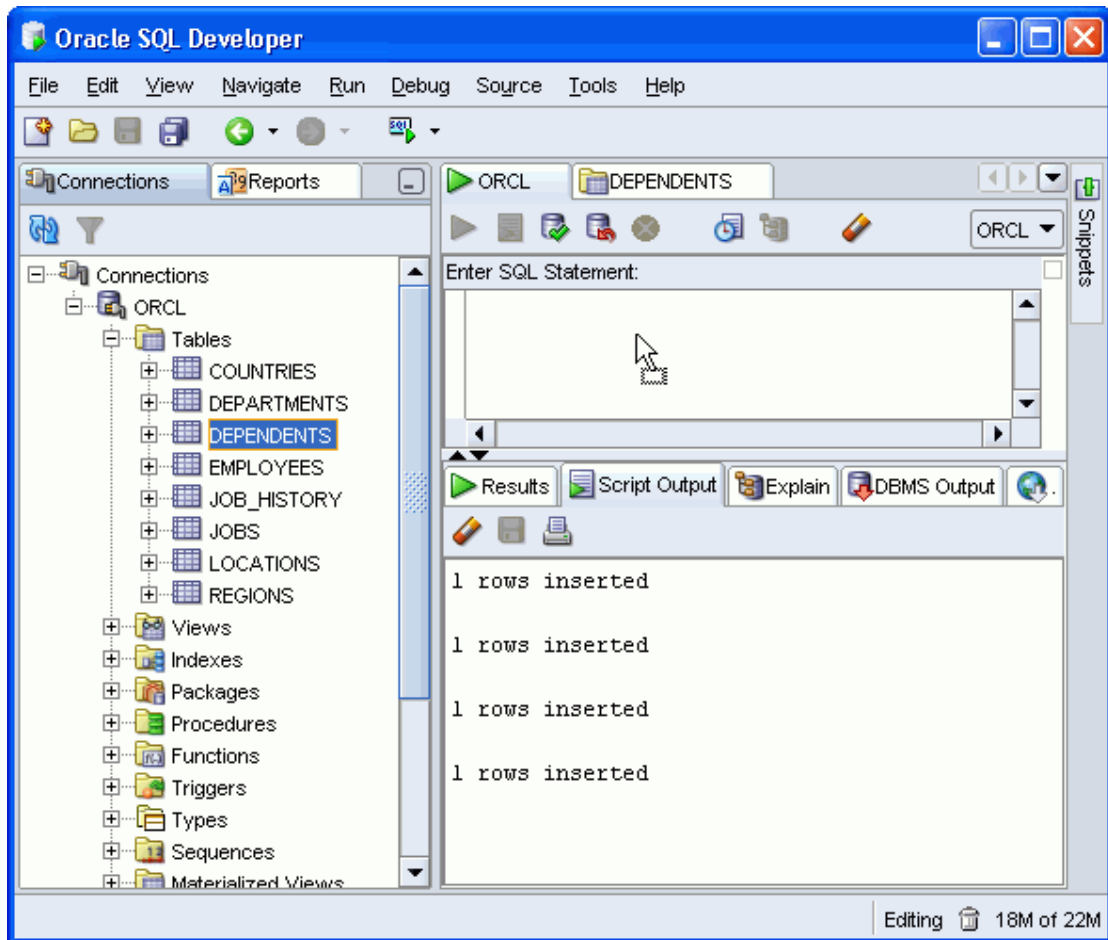
1. 单击 ORCL SQL Worksheet 选项卡。



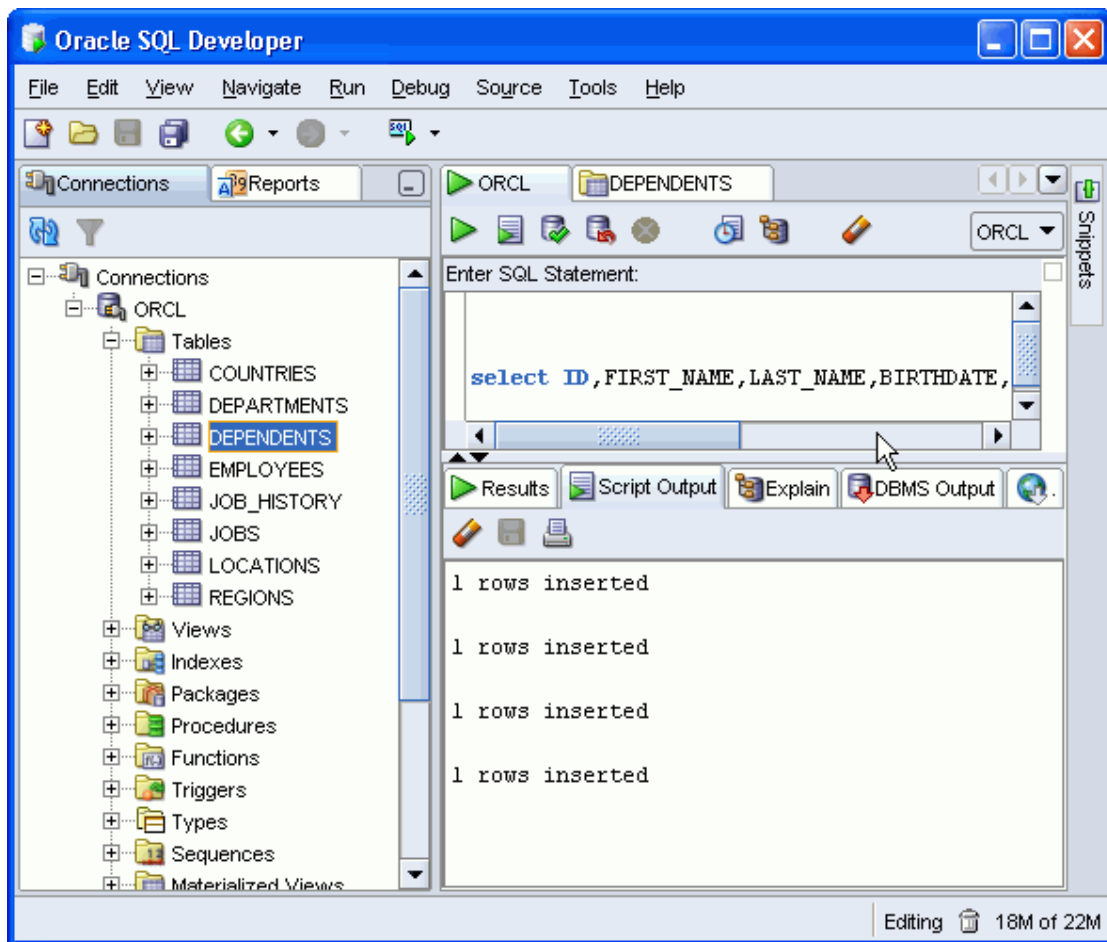
2. 单击 clear（铅笔）图标移除 SQL Statement 窗口的内容。



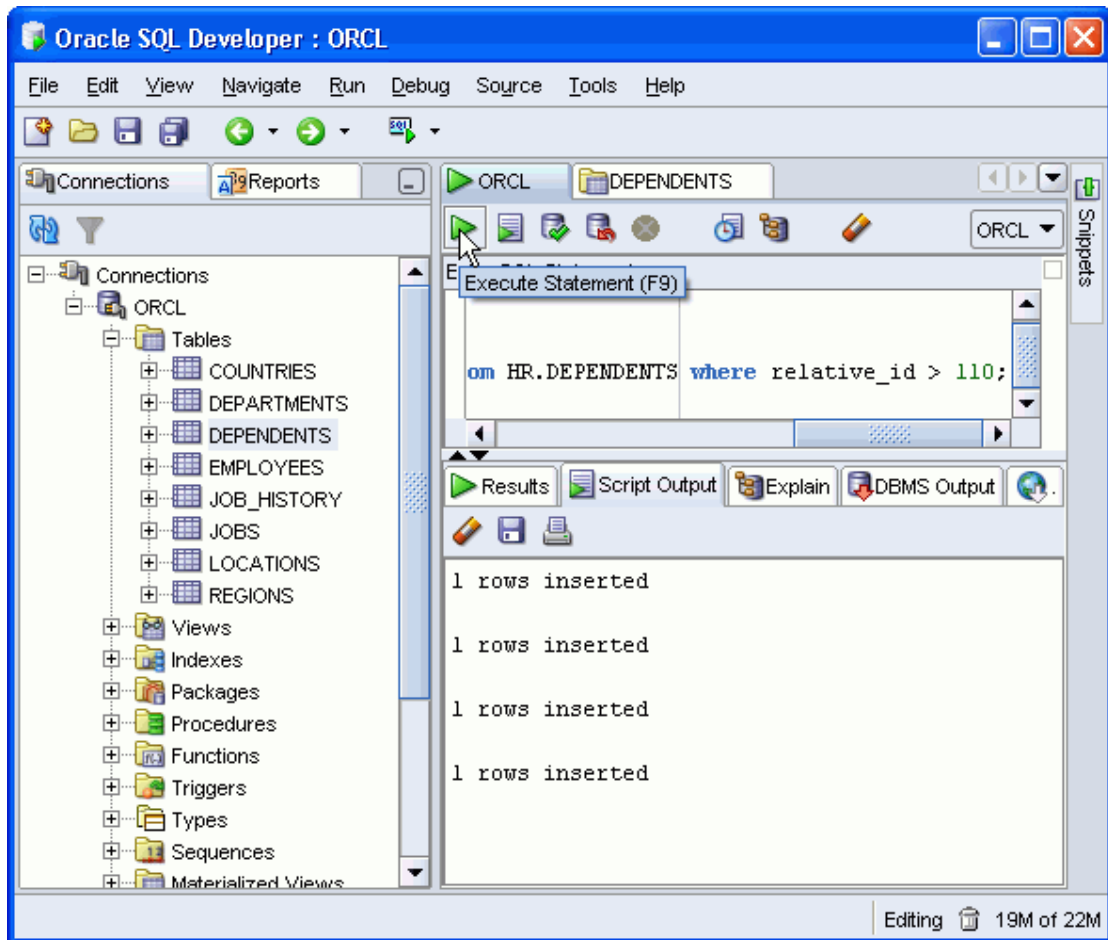
3. 将 DEPENDENTS 表从数据库对象列表拖放到 SQL 语句区域。



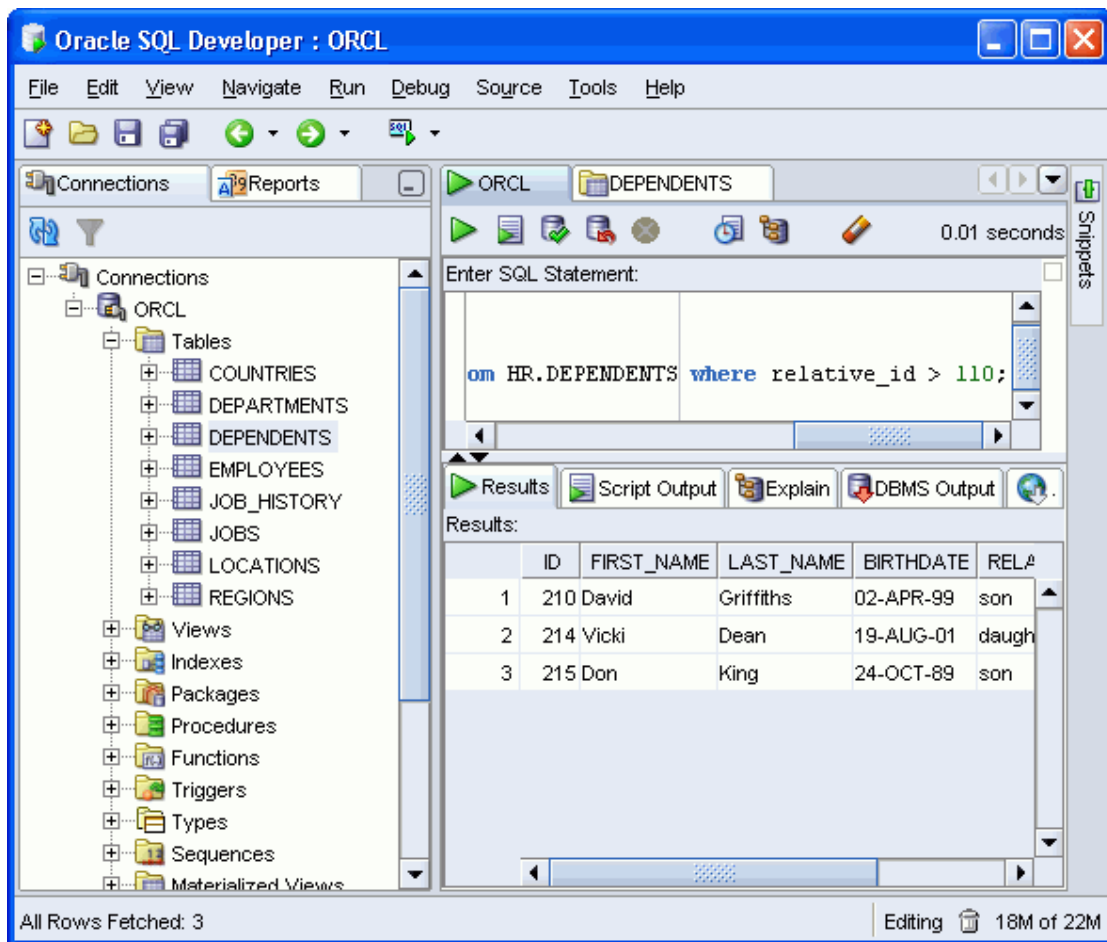
4. 显示 SELECT 语句以及 DEPENDENTS 表中的所有列。向右滚到到 SELECT 语句结尾。



5. 将 WHERE 子句 `where relative_id > 110` 添加到 SELECT 语句末尾 “;” 之前。然后单击 Execute Statement 图标。



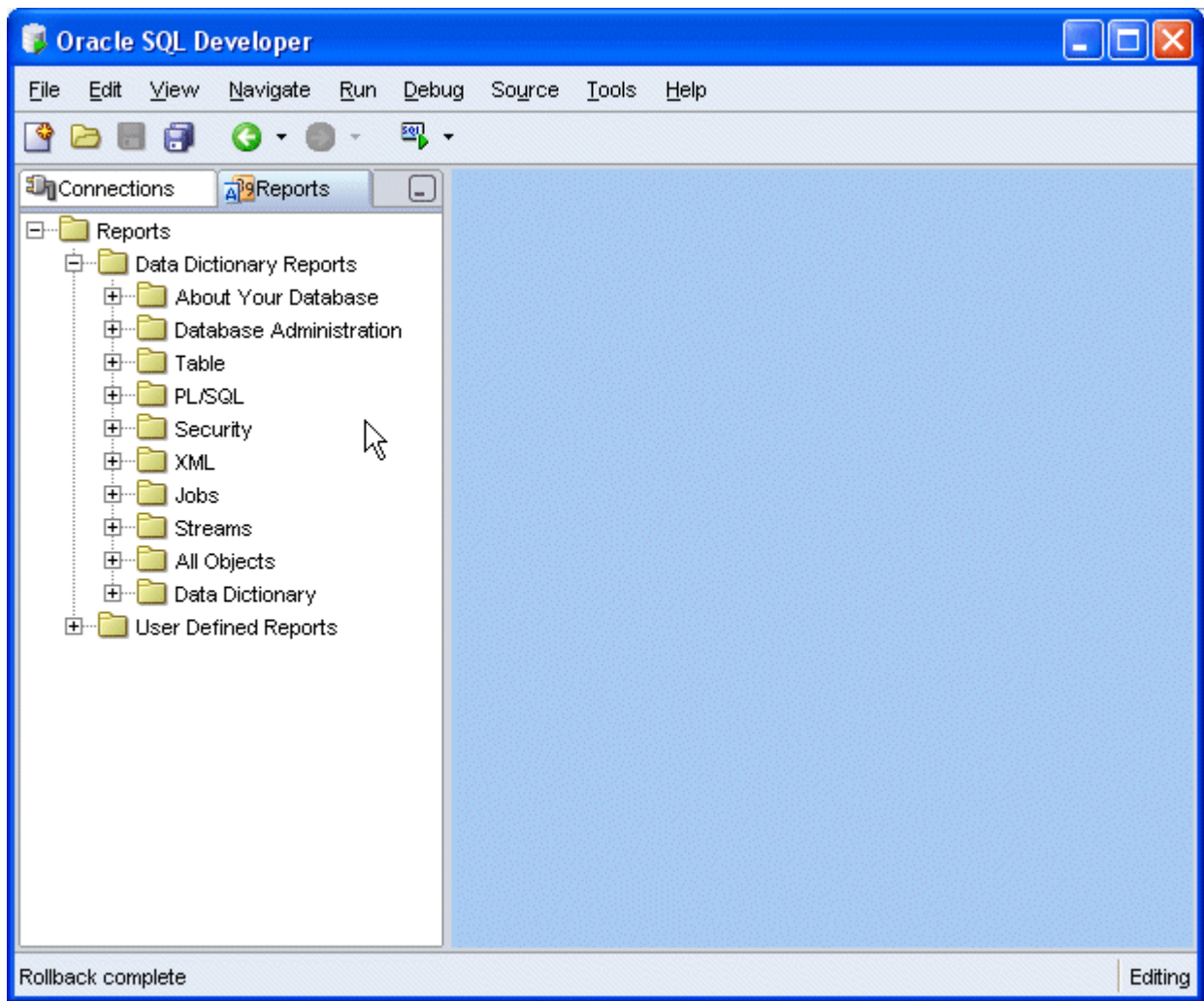
6. 随即显示结果。



使用可用的报表

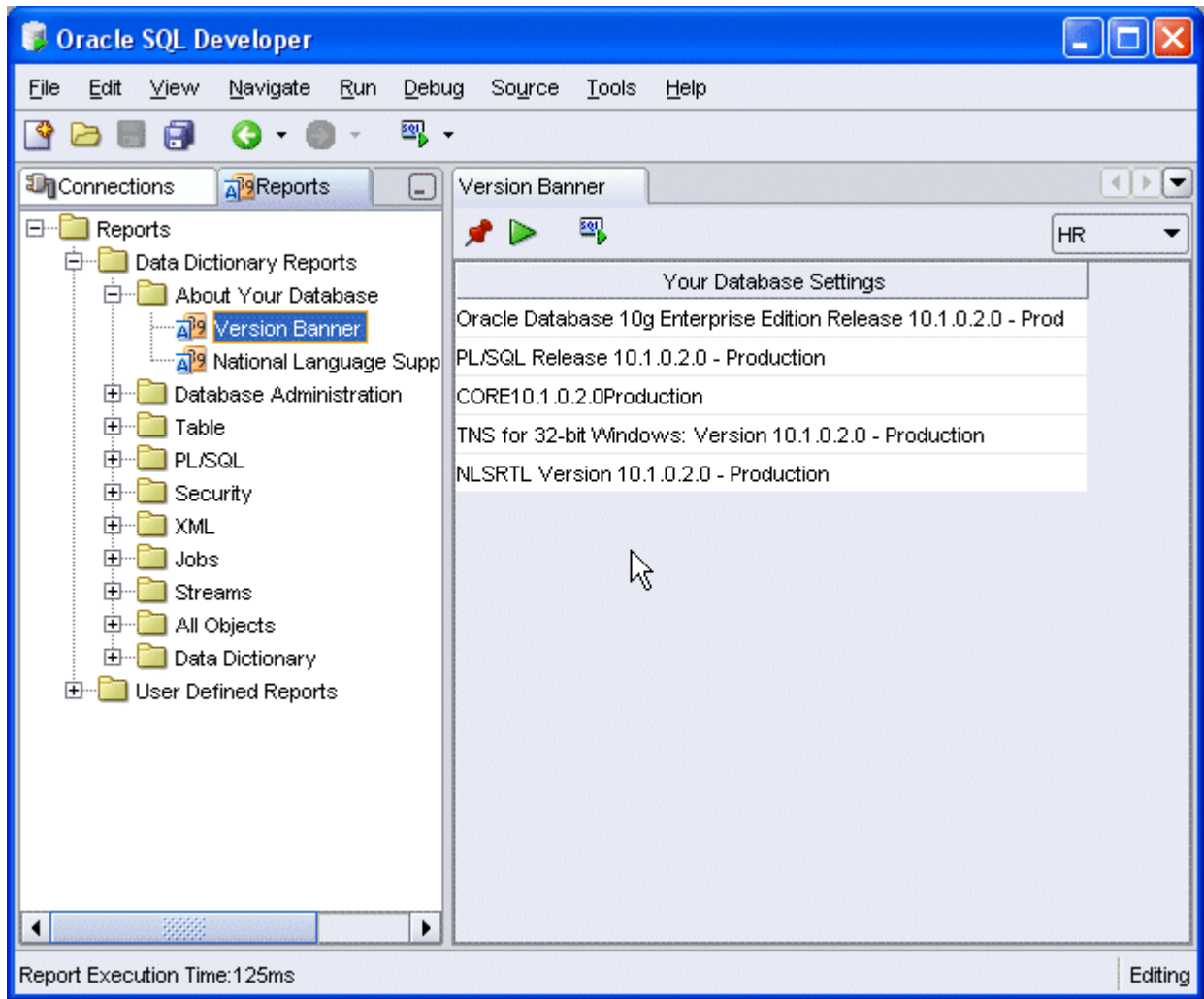
SQL Developer 为本产品提供了许多报表。在这一节中，我们将了解如何运行这些报表。

1. 在 SQL Developer 中，切换至 Reports 选项卡，展开 Data Dictionary Reports 节点。



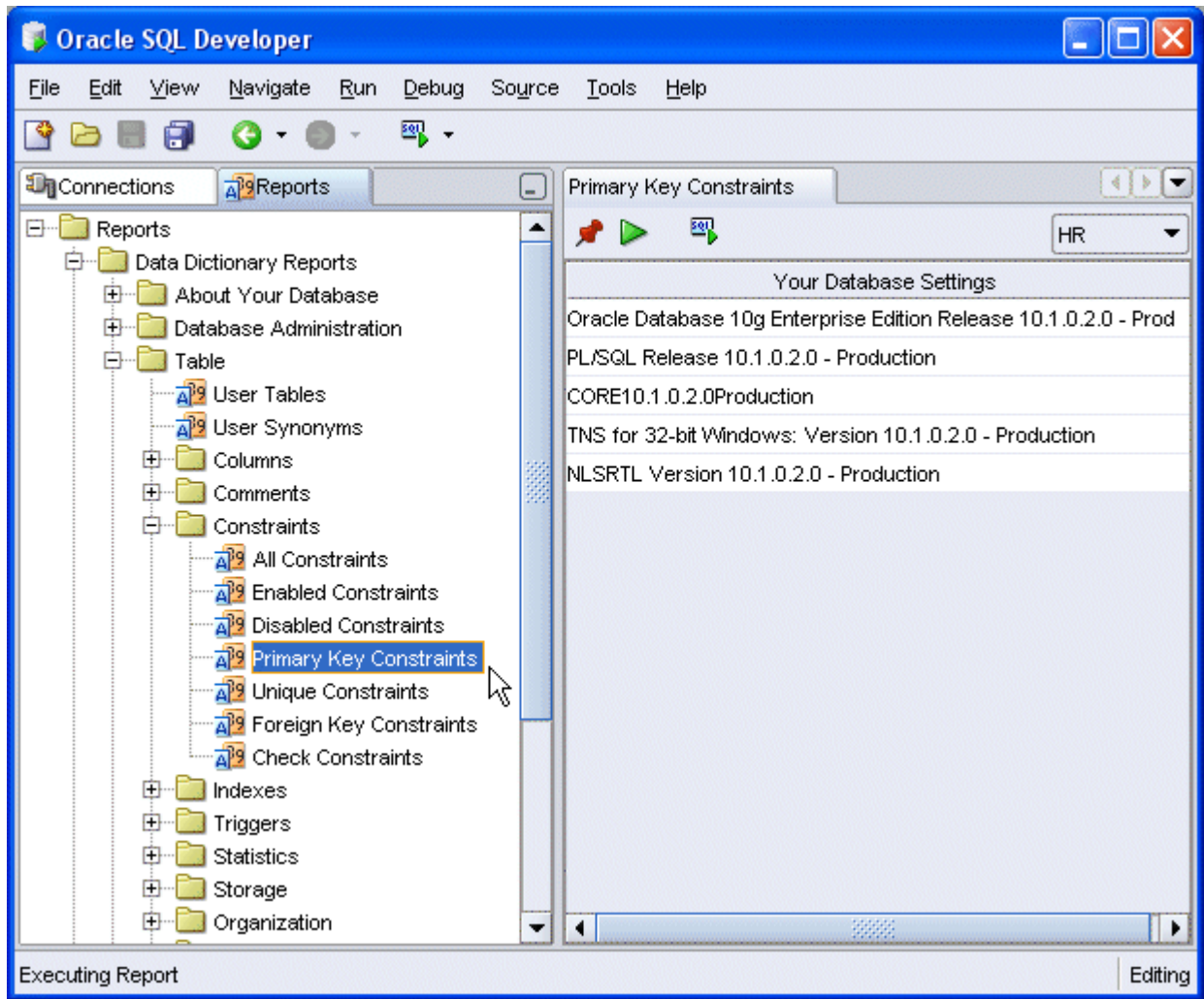
2. 选择 About your Database 节点并展开它，可以了解有关数据库的基本详细信息。

单击 Version Banner。



3. 为完成本练习，可以选择任何报表并运行它。例如，有一个报表列出了表的所有主键。

展开 Table -> Constraints 节点，选择 Primary Key Constraints



4. 选择该节点后，会弹出一个窗口。该窗口允许限制对某个条件的查询。立即单击 Apply，即可选择所有记录。

清除 Null 复选框，在域中键入 e。（该操作限制了对所有名称中包含字母“e”的表的查询）

单击 Apply。

Enter Bind Values

Table Name

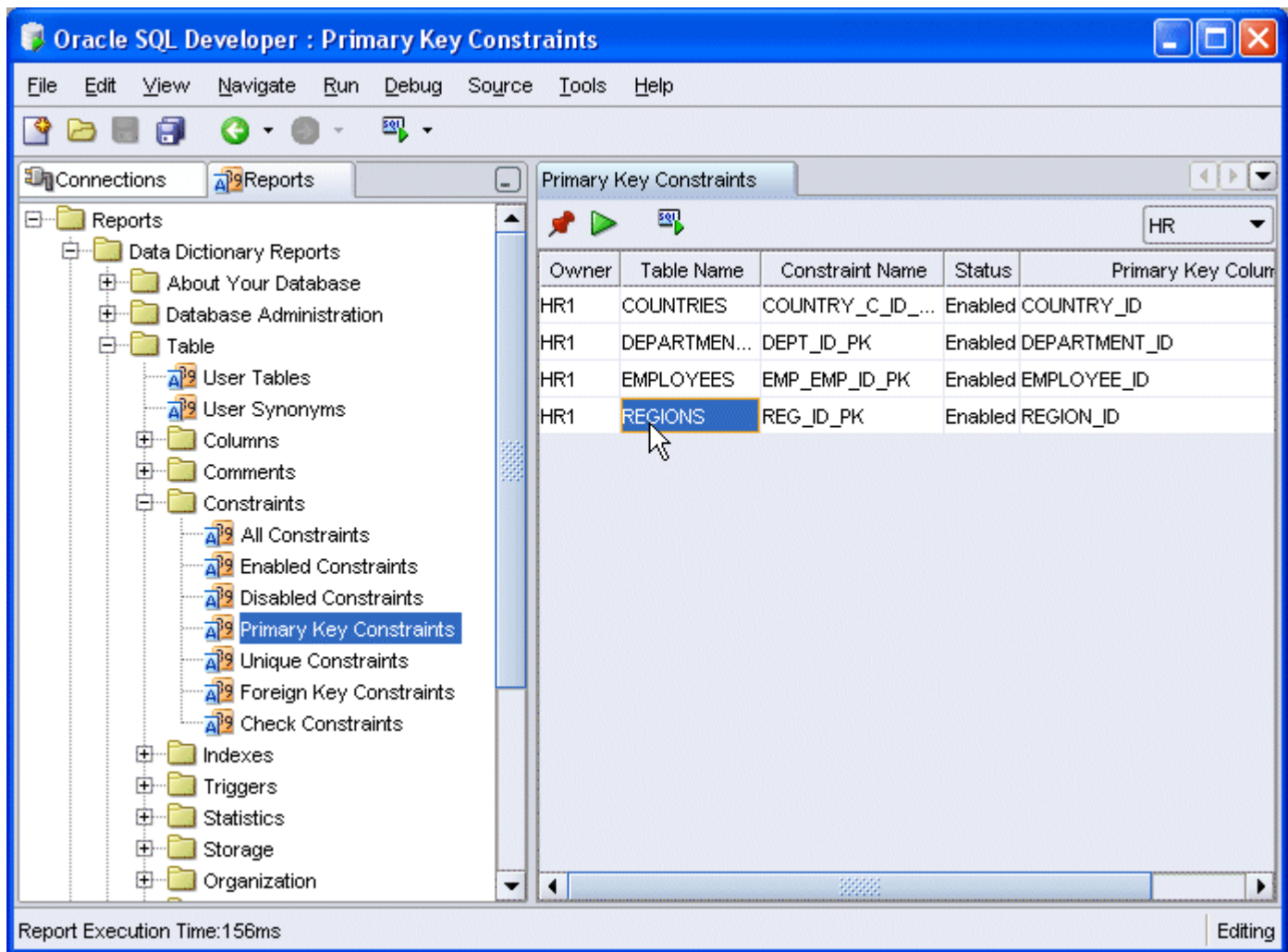
Name: Table Name

Value: e ☐ NULL

Description: Name or part of name to search for (case insensitive).

Apply Cancel

5. 产生的报表显示了所有名称中包含字母“e”的、具有主键的报表。

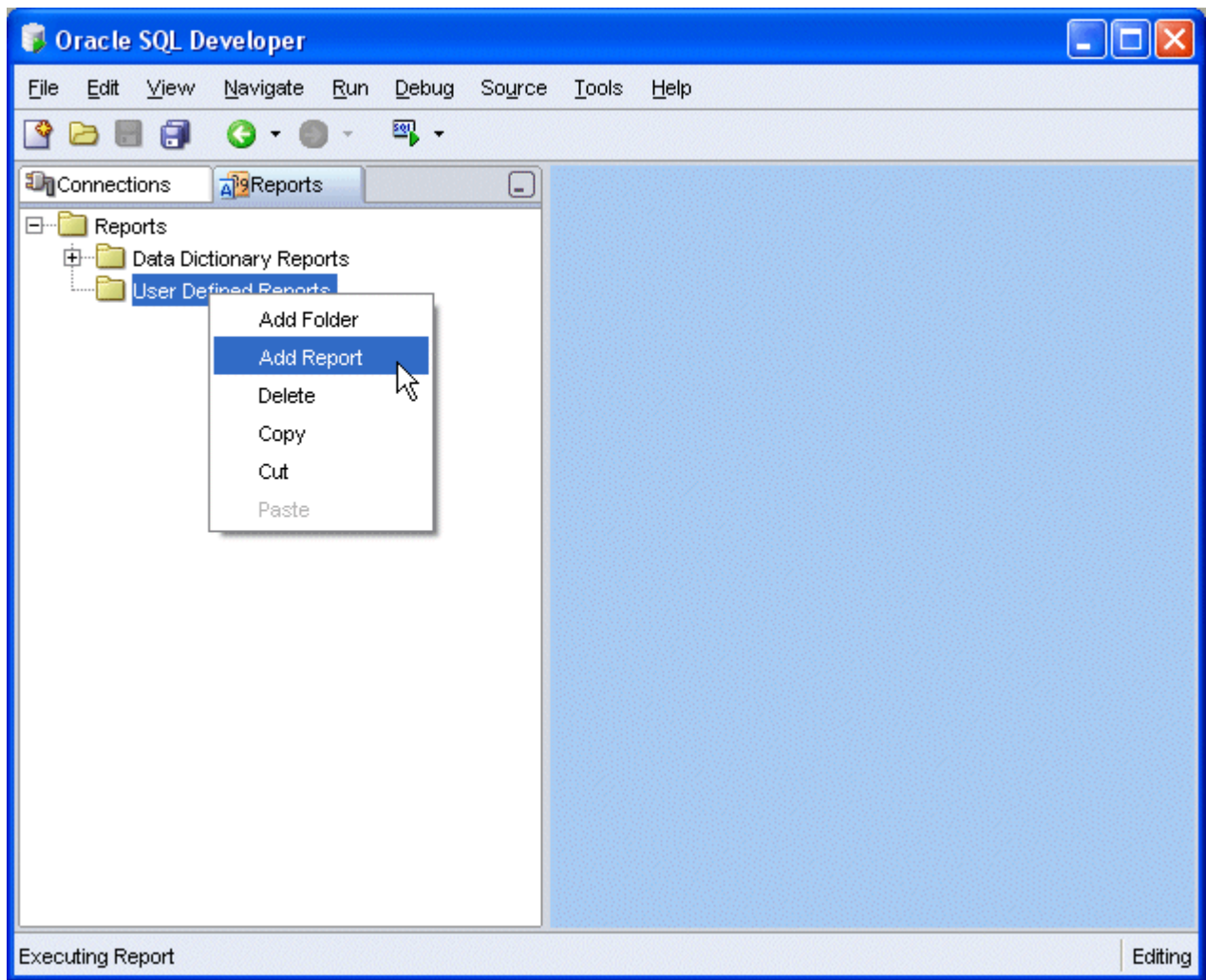


添加用户定义的报表

SQL Developer 提供创建、保存以及运行自己的报表的能力。只需要 SQL 即可。报表可以简单，也可以复杂。

一般把自己需要频繁执行的 sql 语句作为报表保存起来。

1. 使用上下文菜单添加一个新报表。



2. 可以在此处添加任何查询。例如，使用以下查询：

```
select object_name, object_type  
from user_objects;
```

可以对报表和名称进行描述。例如：

Edit Report Dialog

*Name: User_Objects

Description: Finding all user objects in a schema

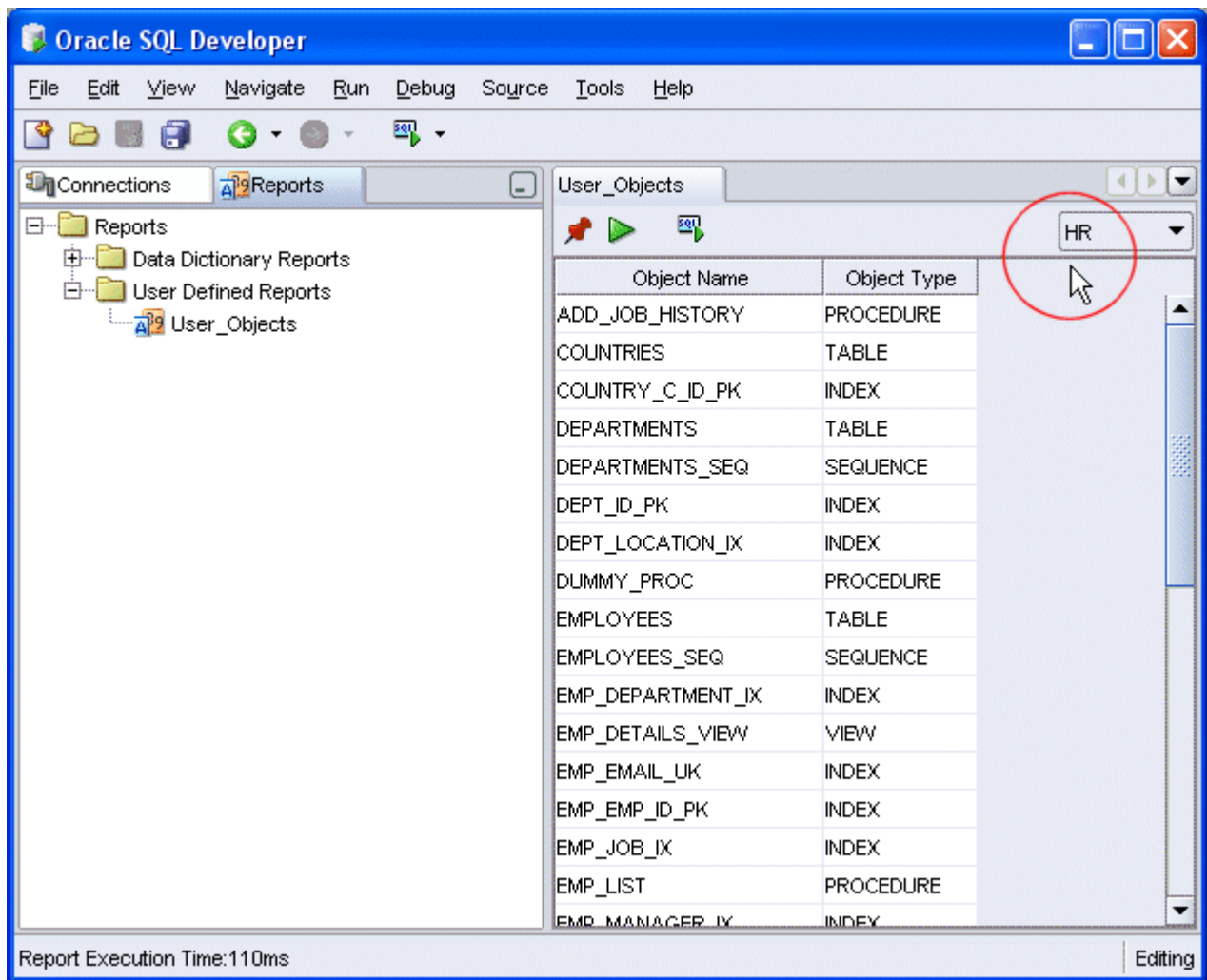
ToolTip:

SQL:

```
select object_name, object_type  
from user_objects;
```

Buttons: Help, Apply, Cancel

- 单击 Apply 后，报表提示连接到数据库。可以使用 HR，或者使用其他连接。



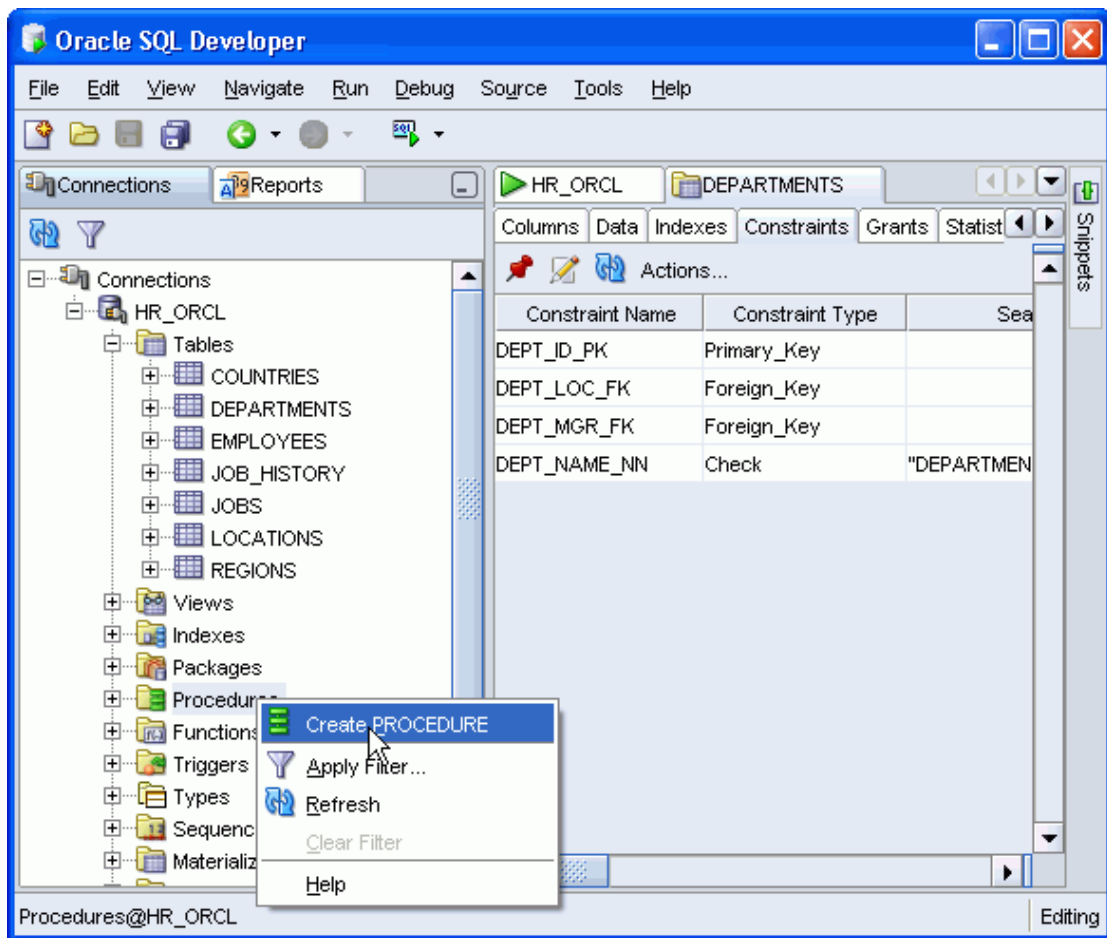
注意，从右上角可以识别使用了哪个连接、何时运行该报表。

4. 最后，可以采用该查询，在 SQL Worksheet 使用它，方法是单击 Run Report in SQL Worksheet 按钮。

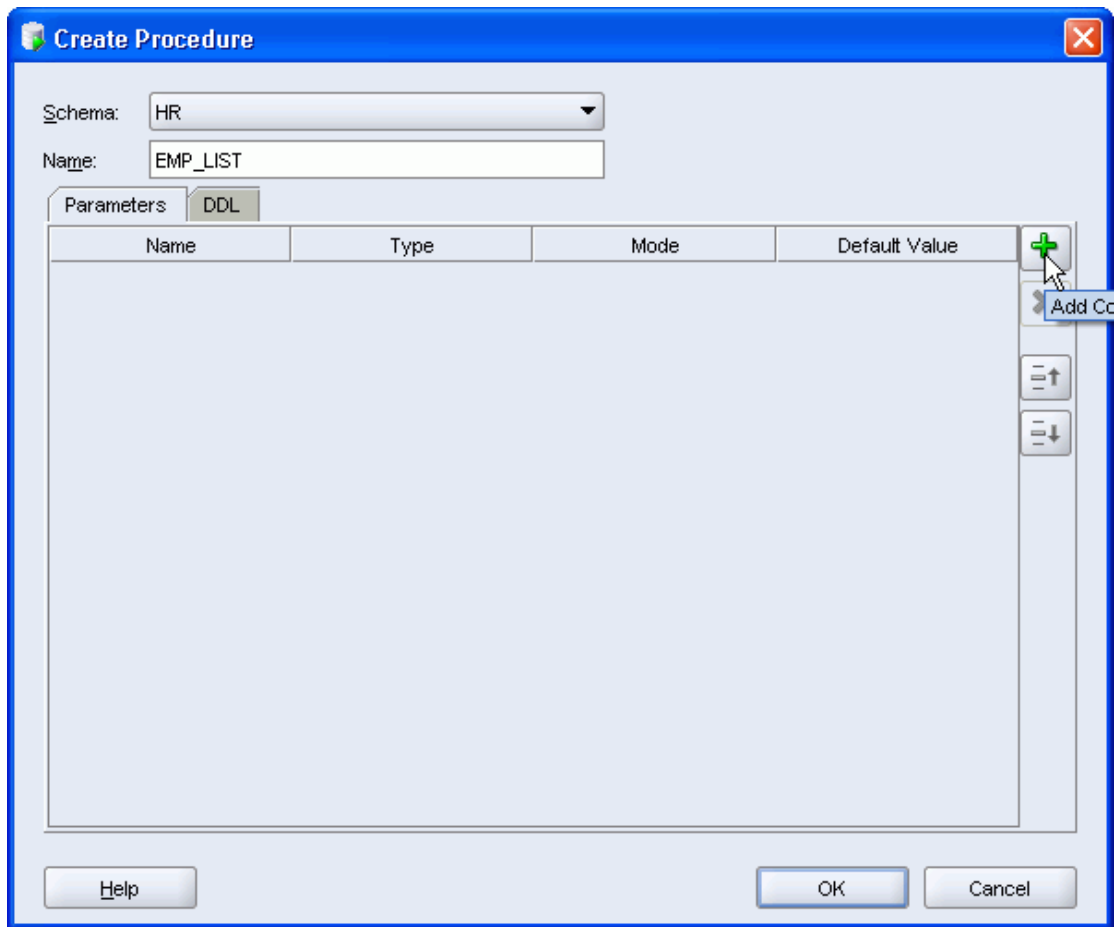
创建并编译 PL/SQL 过程

创建、编辑并编译一个 PL/SQL 过程。执行以下步骤：

1. 在 Connections 导航器中，右键单击 Procedures 节点以调用上下文菜单，然后选择 Create PROCEDURE。



2. 输入 EMP_LIST 作为过程名。然后单击 +，添加一个参数。



3. 双击参数，将值更改为 pMaxRows，然后将 VARCHAR2 更改为 NUMBER。确保在单击 OK 之前按了 Enter 键。

Create Procedure

Schema:

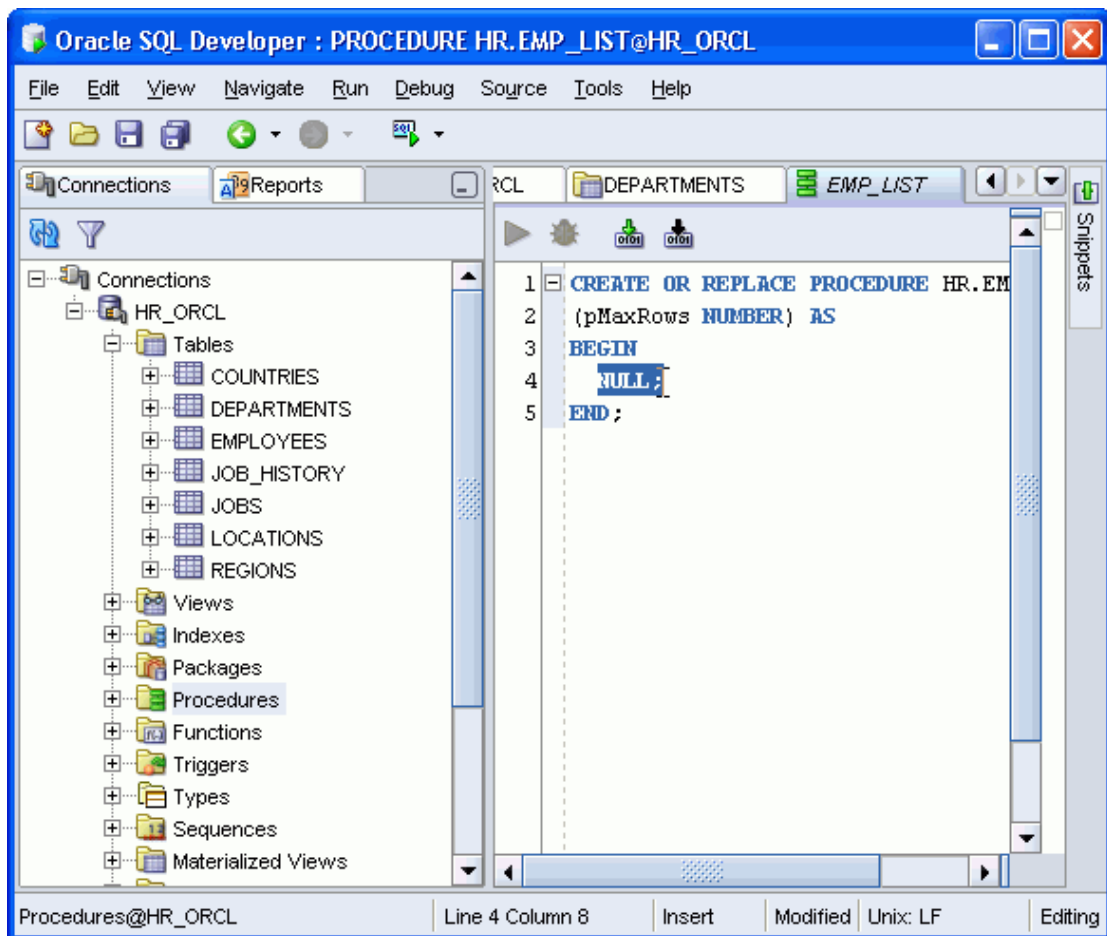
Name:

Parameters **DDL**

Name	Type	Mode	Default Value
pMaxRows	NUMBER		

Buttons: +, -, ↑, ↓, OK, Cancel, Help

4. 随即显示指定了参数的过程的框架。



5. 替换以下 PL/SQL:

BEGIN

NULL;

END;

使用以下代码:

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

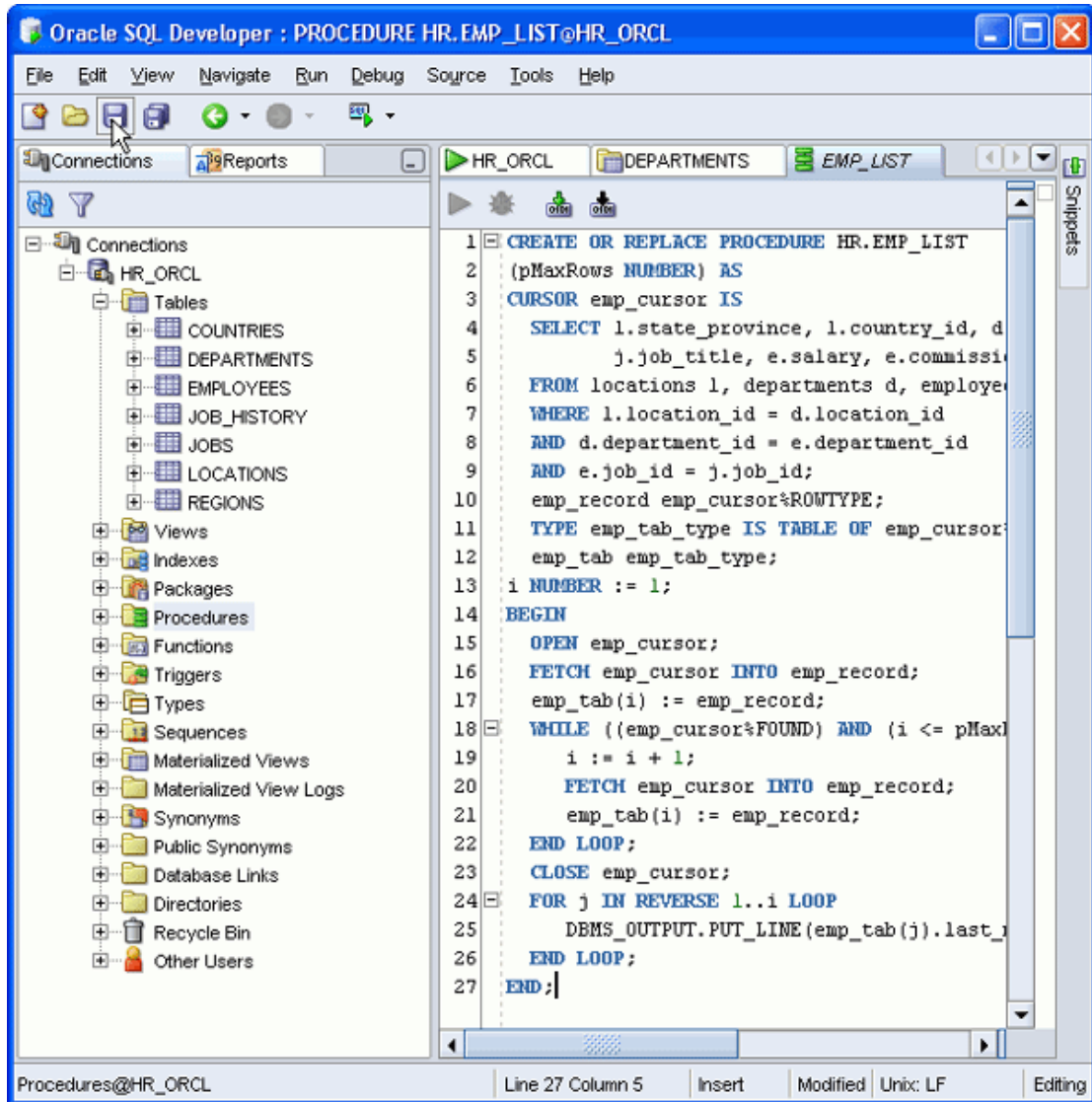
        DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name);

    END LOOP;

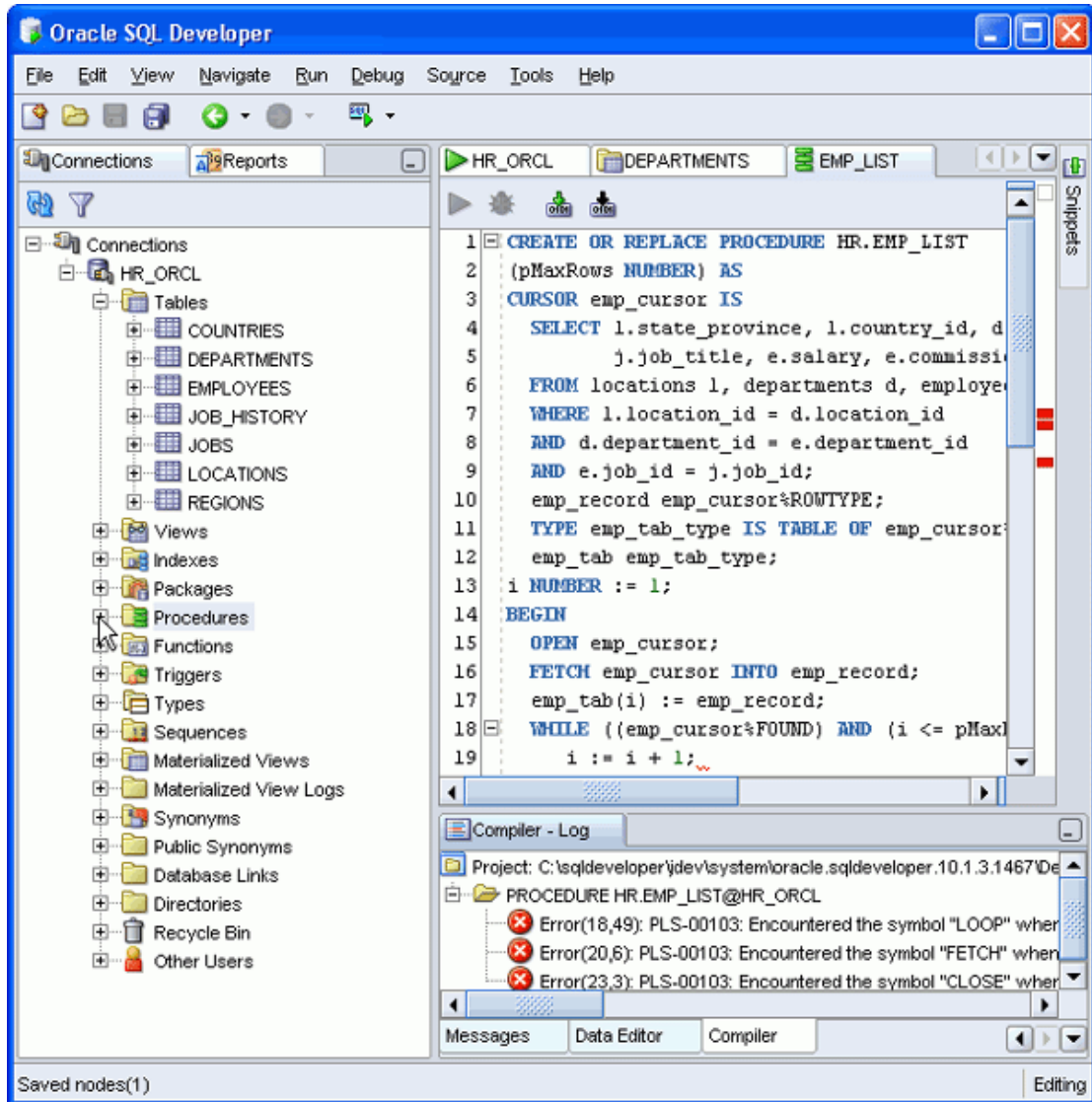
END;

```

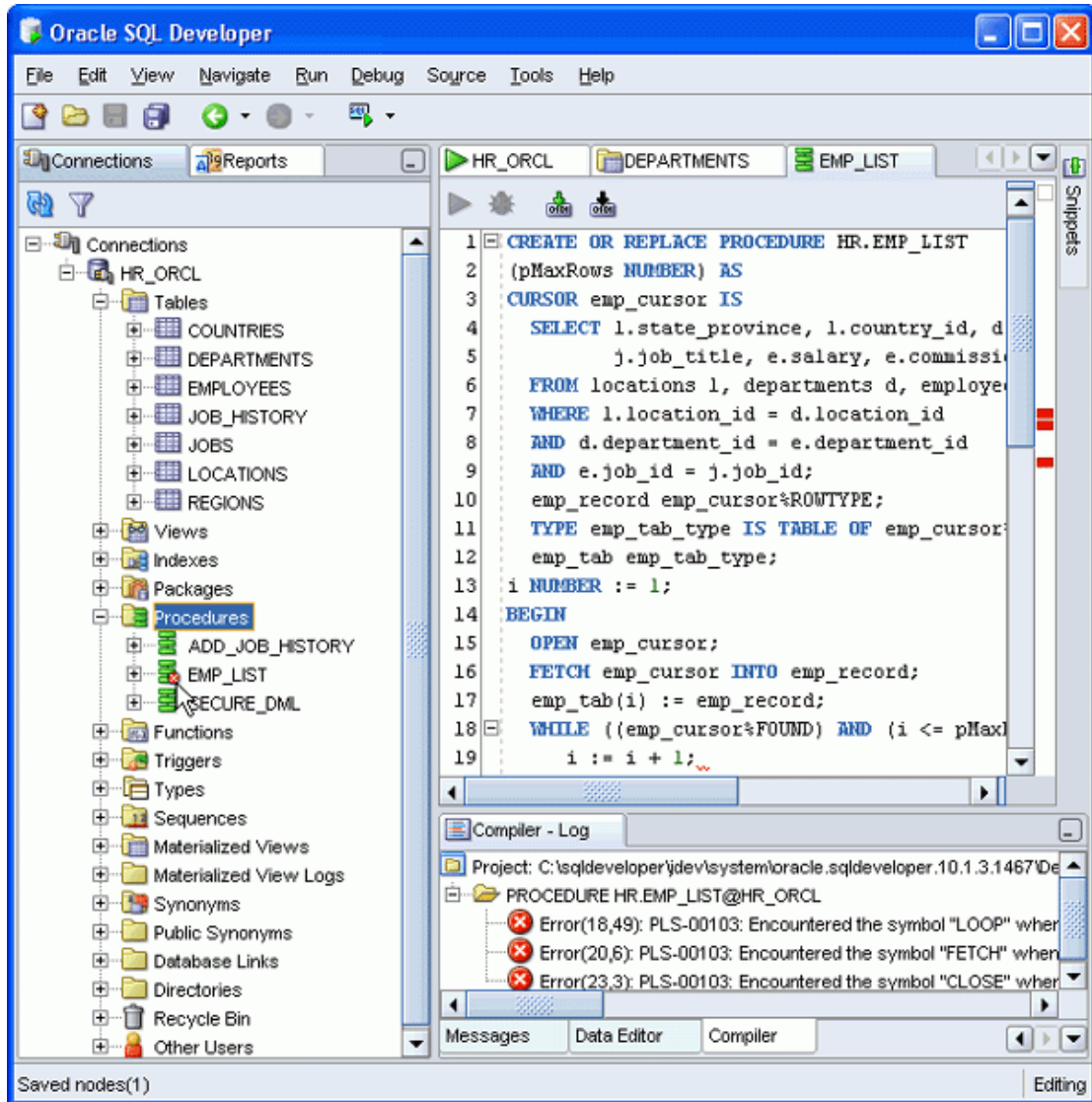
单击工具栏中的 Save 按钮，编译 PL/SQL 子程序。



6. 在导航器中展开 Procedures。

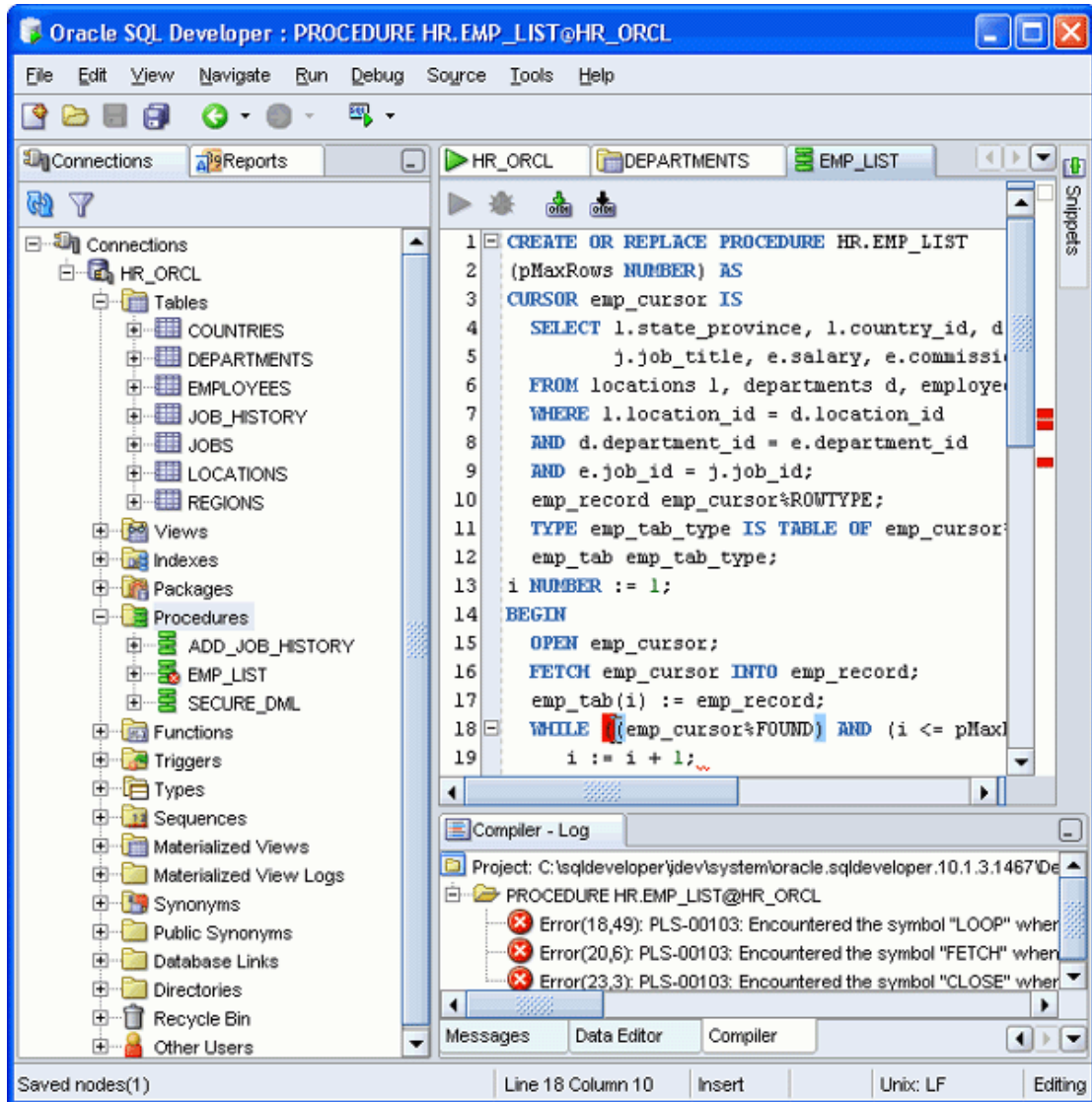


7. 注意，当 SQL Developer 检测到无效 PL/SQL 子程序时，系统导航器中该子程序的图标上用红色的 X 来指示状态。

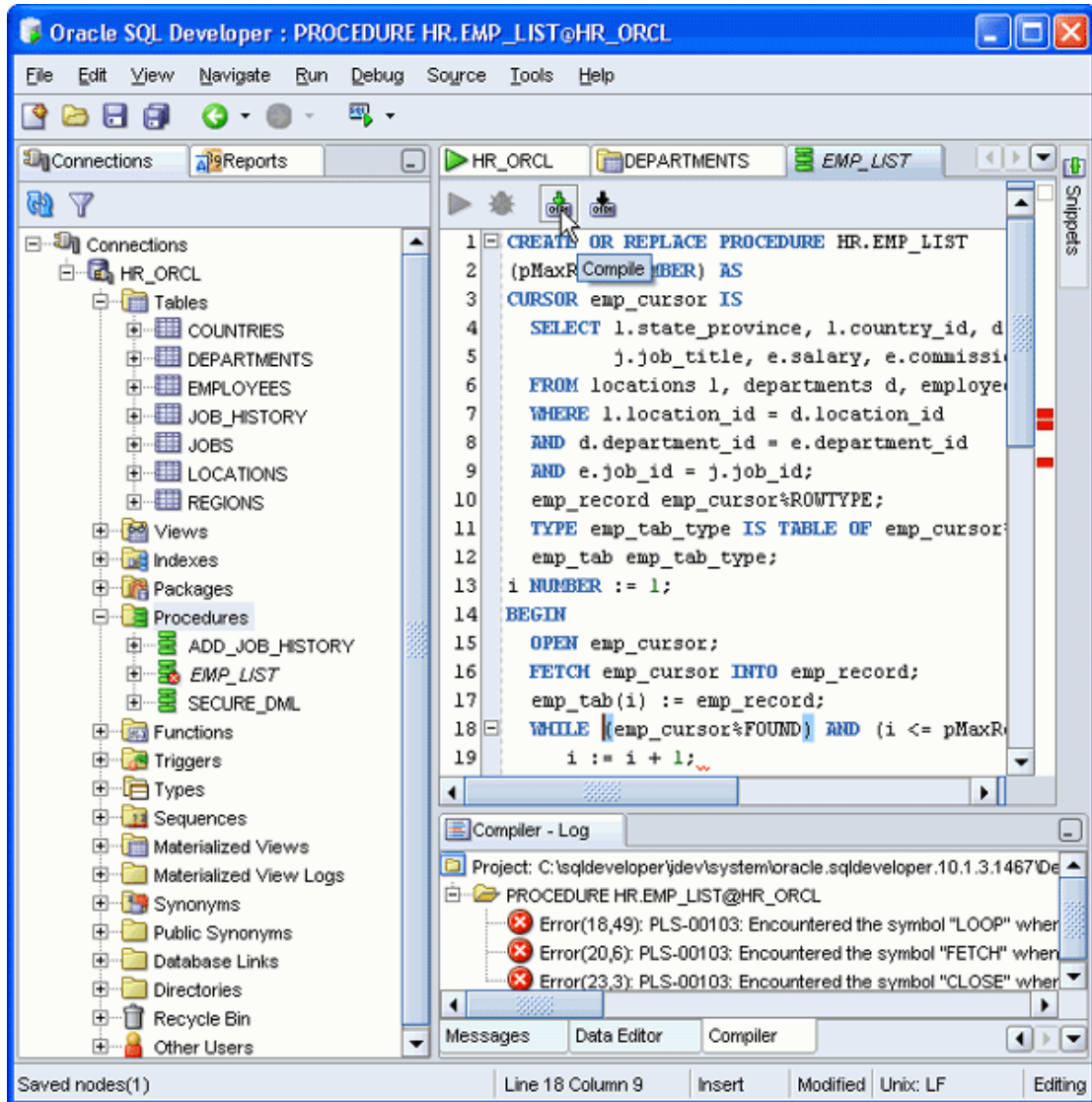


Log 窗口中显示编译错误。只需双击错误，即可导航到错误中报告的对应该行。SQL Developer 还在右侧边列 (gutter) 中显示错误和提示。如果将鼠标放在边列中每个红色方块上，将显示错误消息。

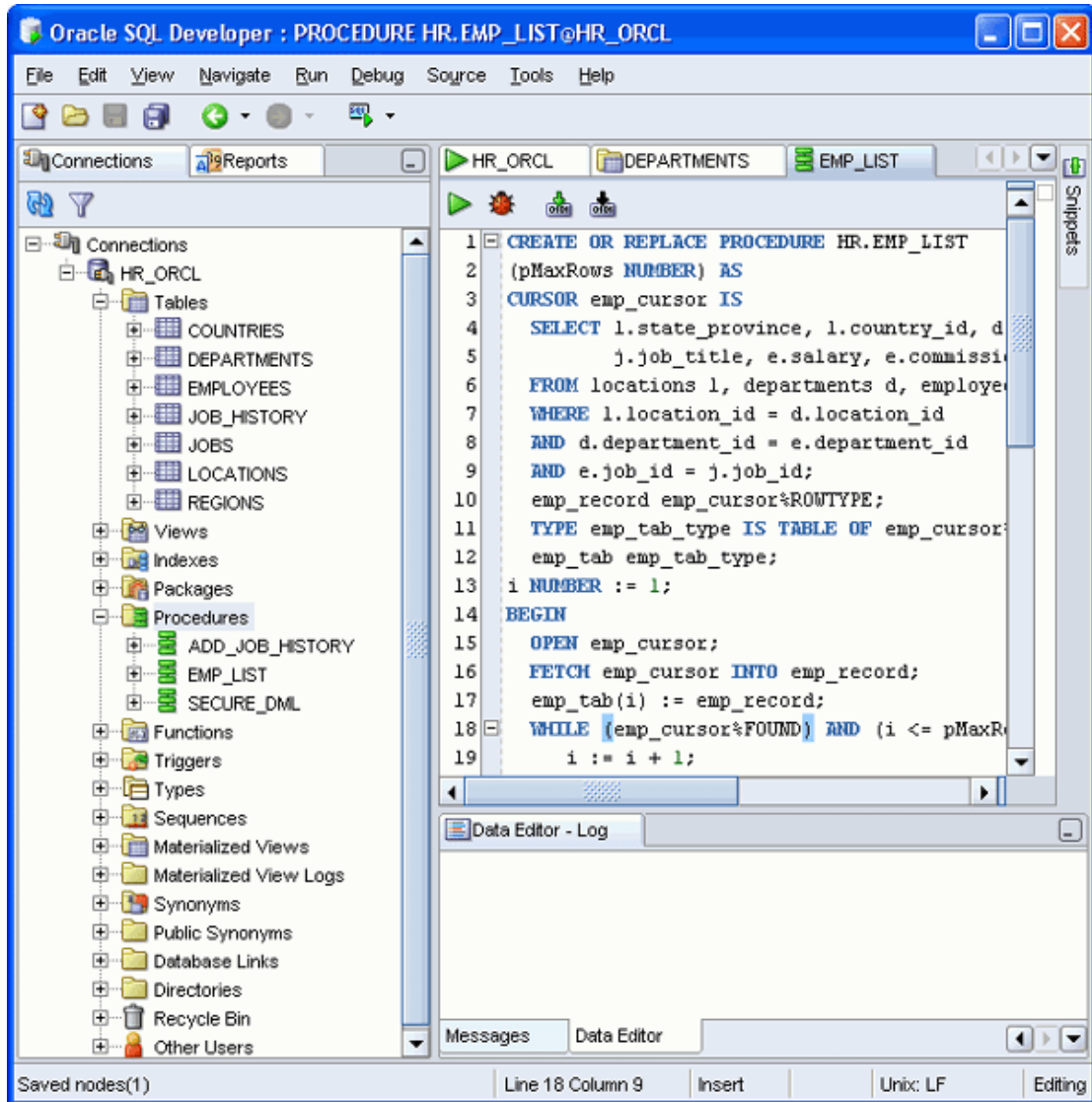
在本例中，错误消息指示 LOOP 语句中存在格式错误。仔细检查代码后，会发现 WHILE 语句中多了一个圆括号。删除多余的圆括号。



8. 单击 Compile 图标。



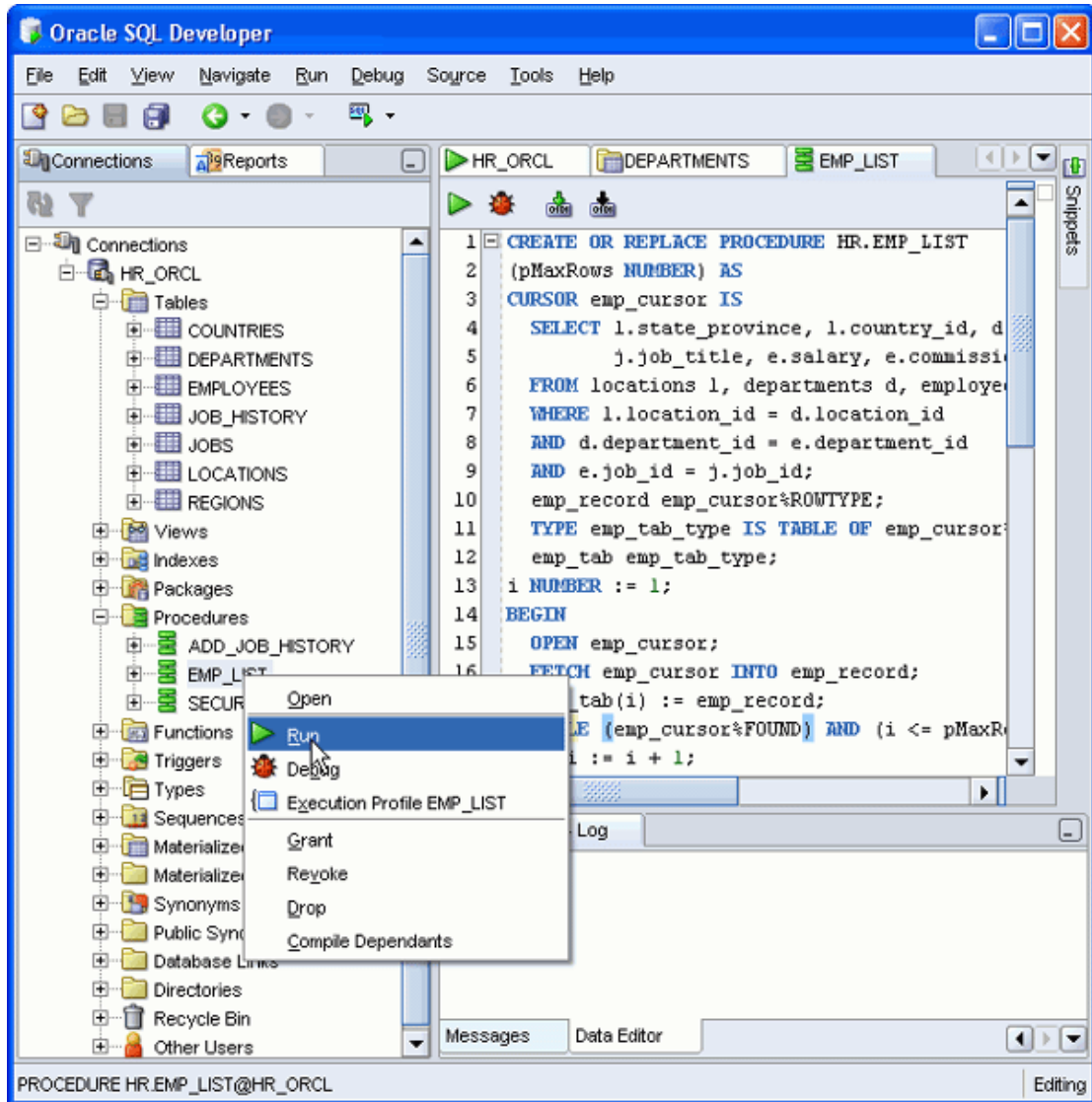
9. 该过程成功编译。现在可以运行该过程。



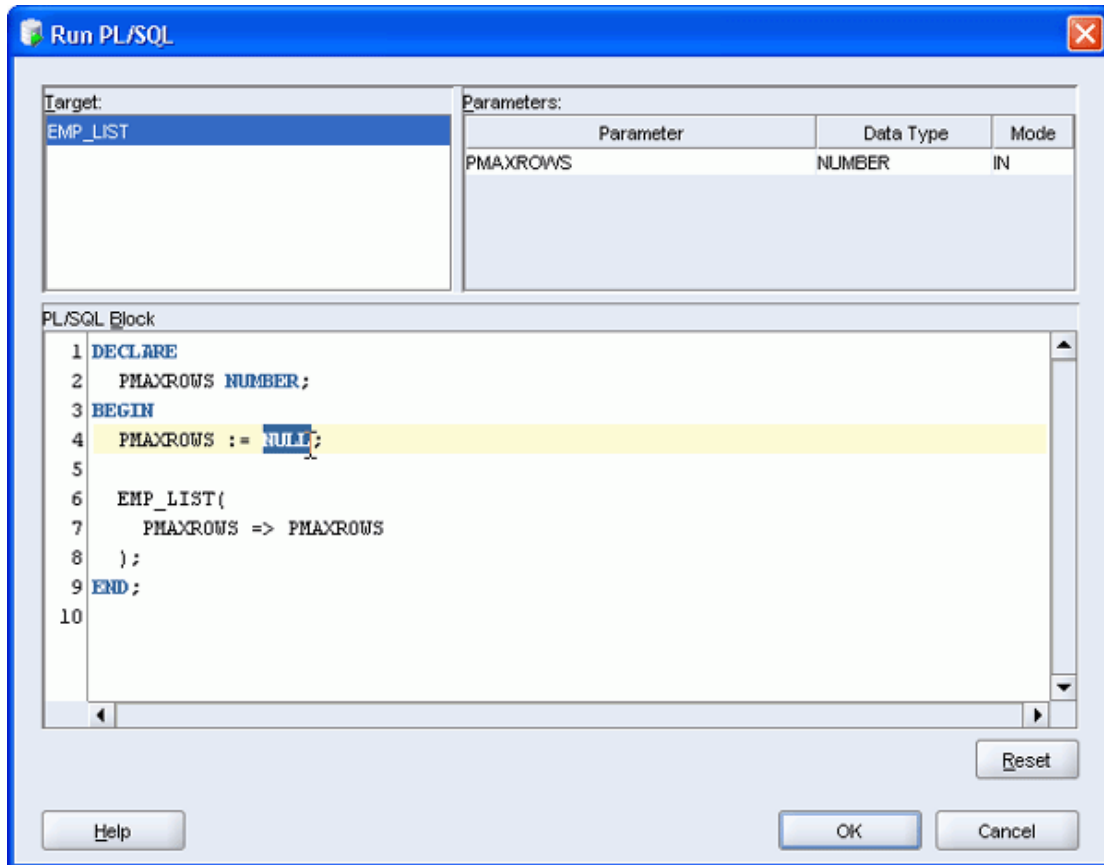
运行 PL/SQL 过程

创建并编译了一个 PL/SQL 过程之后，可以使用 SQL Developer 运行它。执行以下步骤：

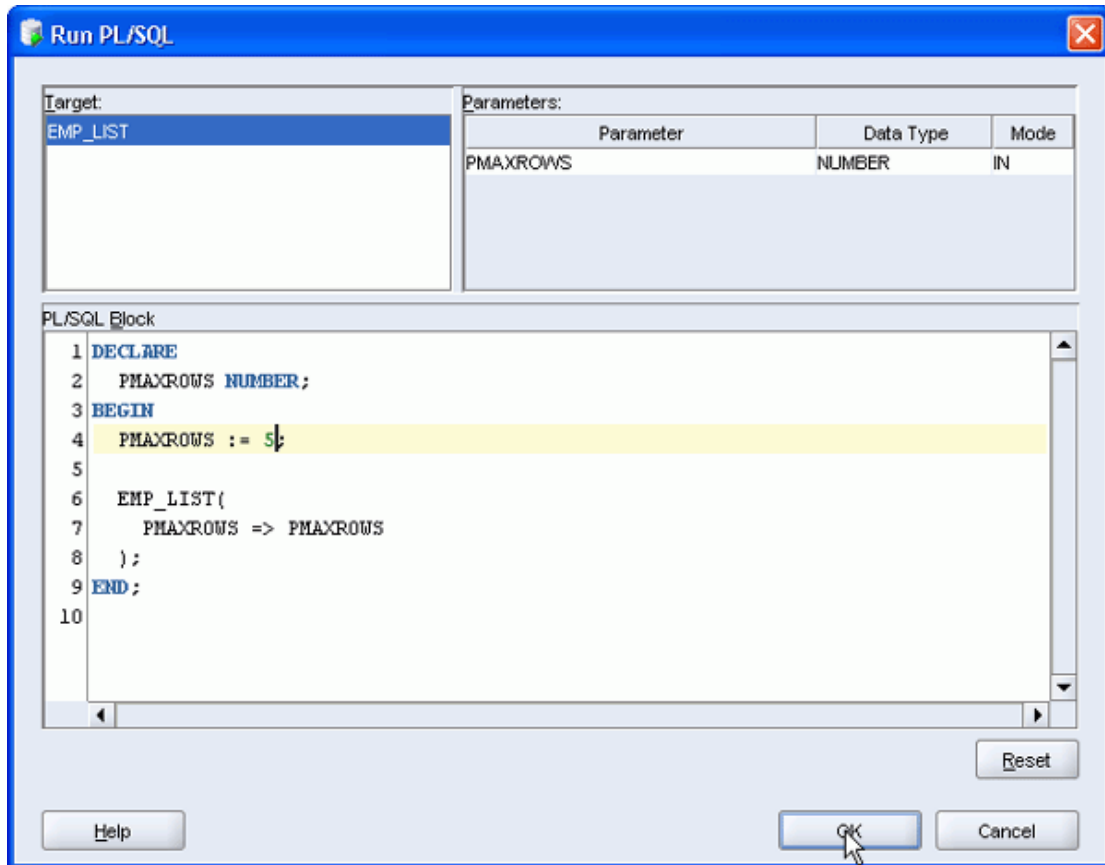
1. 在左侧导航器中，右键单击 EMP_LIST 并选择 Run。



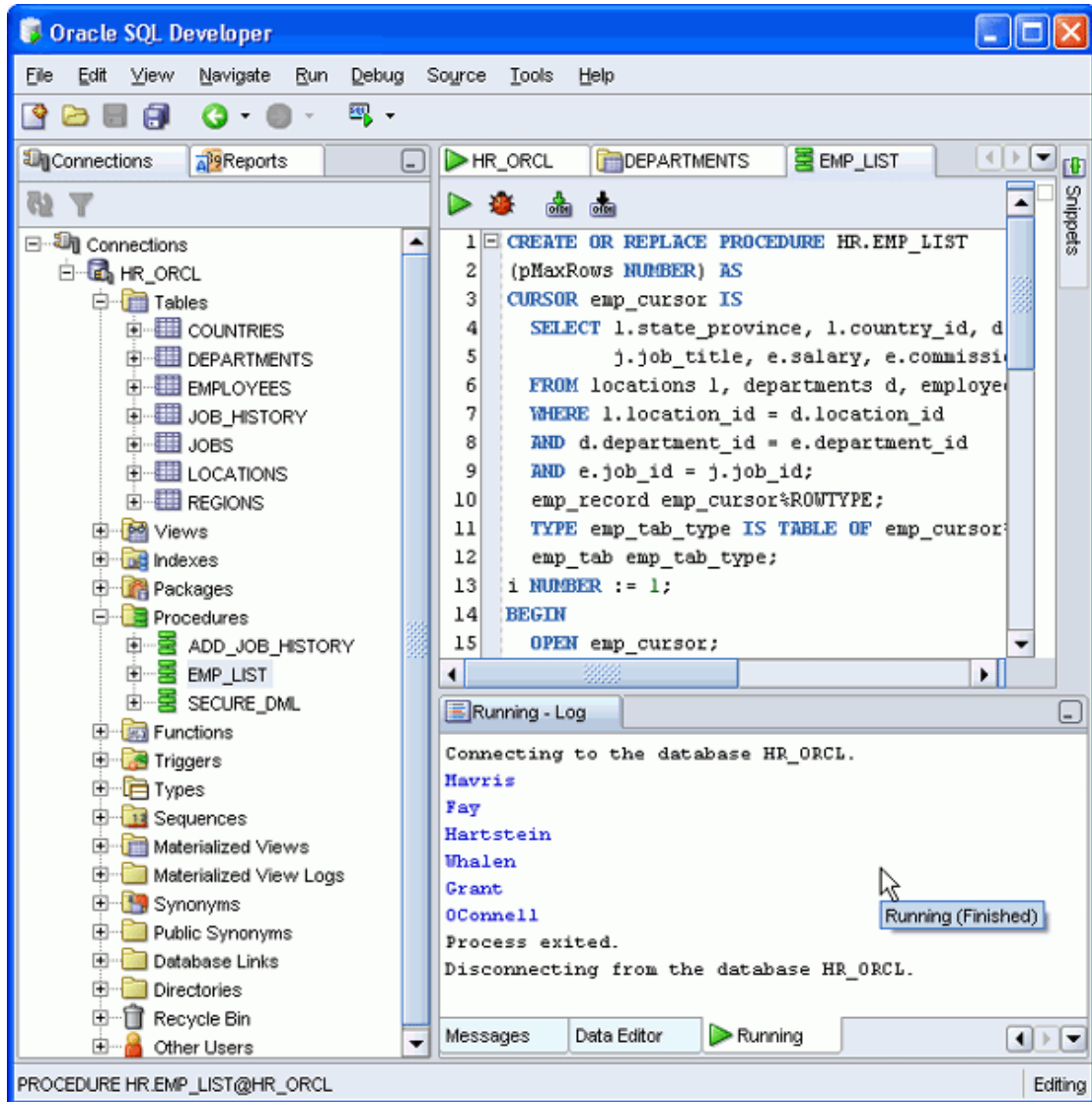
2. 该操作将调用 Run PL/SQL 对话框。Run PL/SQL 对话框允许选择要运行的目标过程或函数（对程序包有用），并显示所选目标的参数列表。PL/SQL 块文本区域中包含的是 SQL Developer 用来调用所选程序的生成代码。使用该区域填充要传送到程序单元的参数以及处理复杂的返回类型。



将 PMAXROWS := NULL; 更改为 PMAXROWS := 5; 然后，单击 OK。



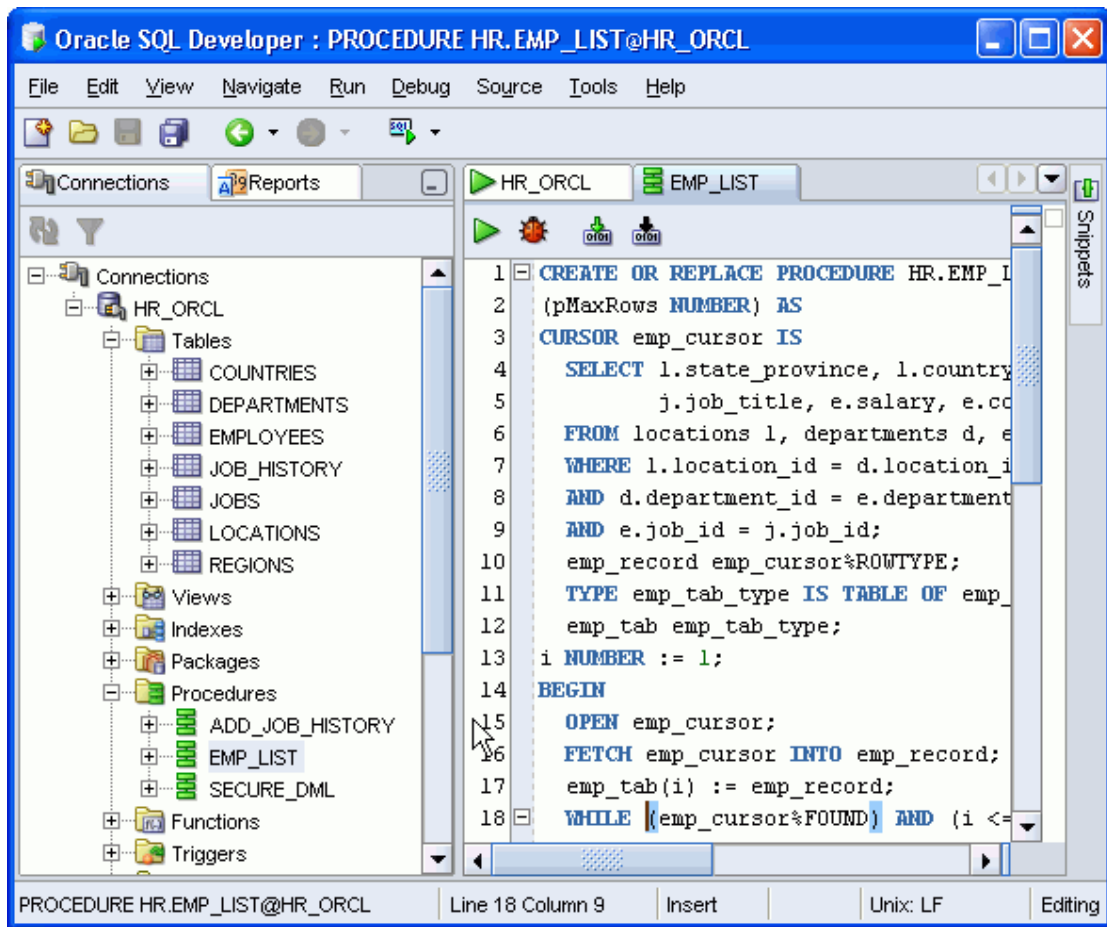
3. Log 窗口中显示了返回的 5 行结果。



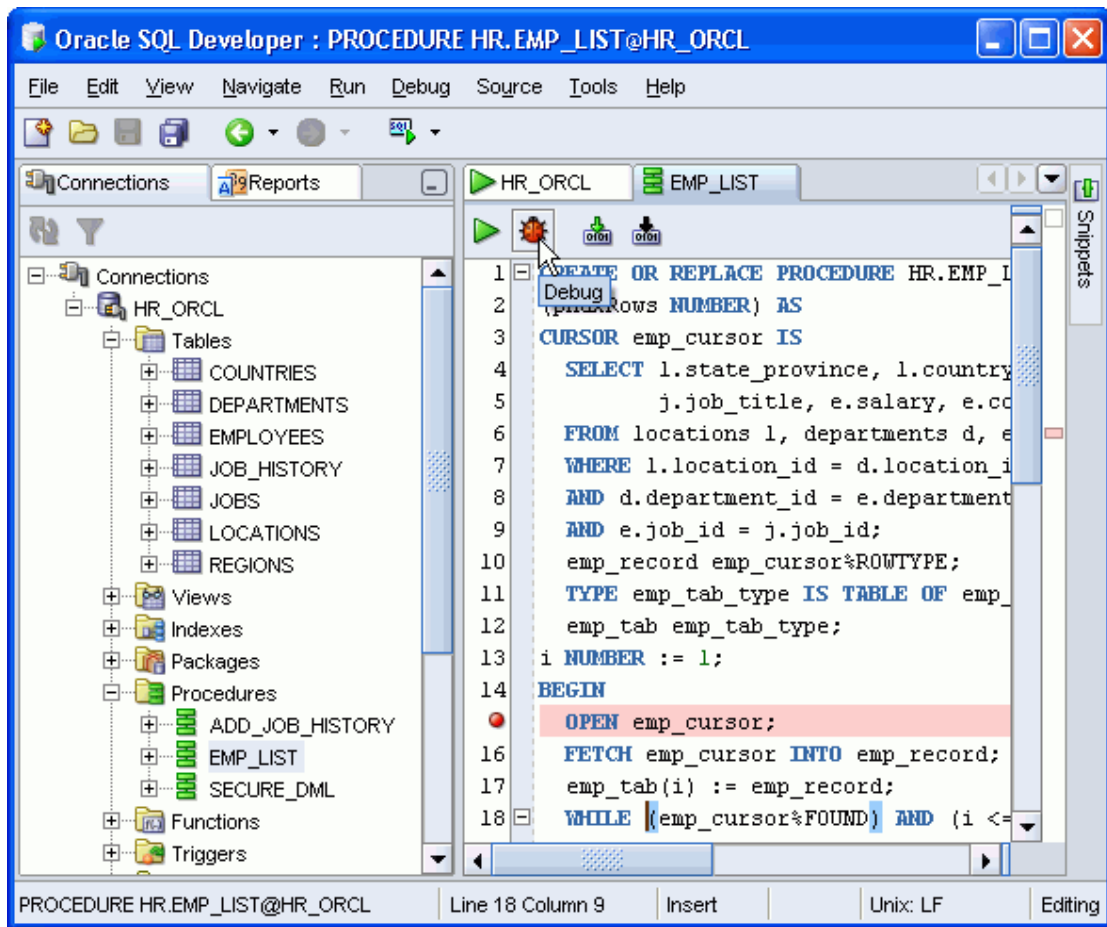
调试 PL/SQL 过程

SQL Developer 还支持使用 Oracle 数据库进行 PL/SQL 调试。调试一个 PL/SQL 过程、单步调试代码并在运行时修改值。执行以下步骤：

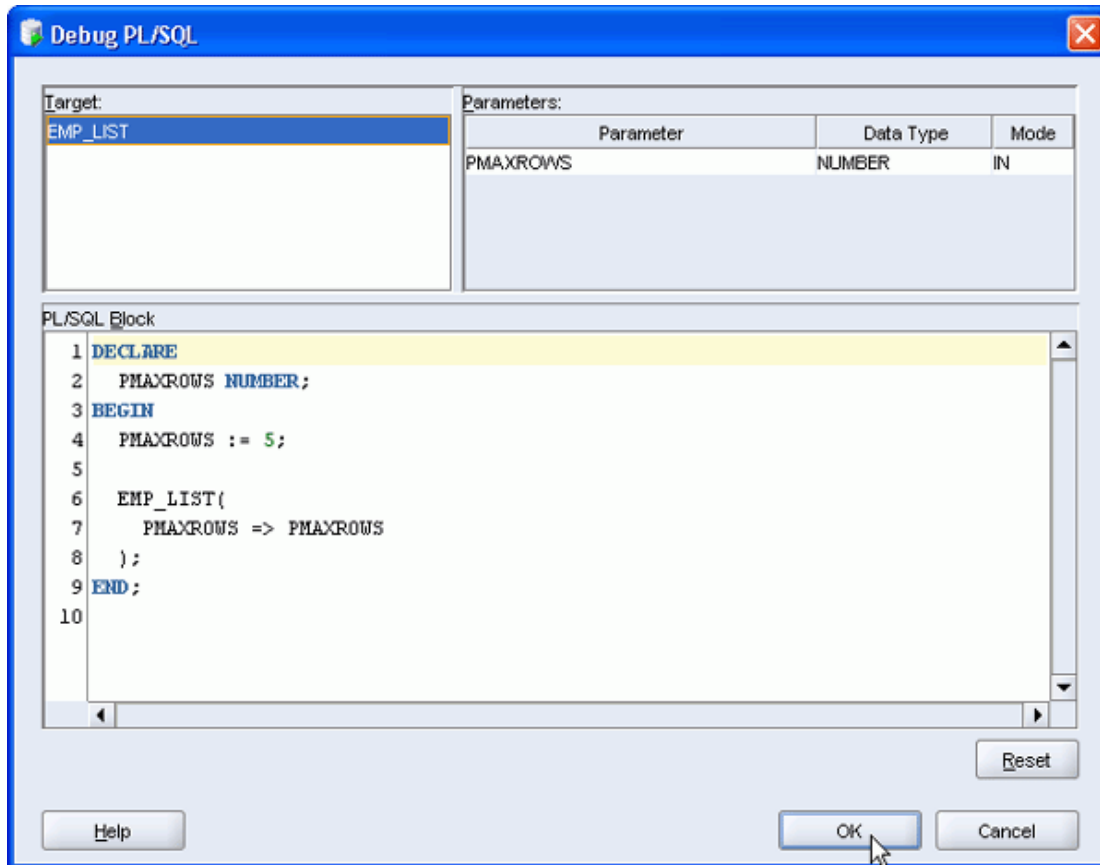
1. 通过在 `OPEN emp_cursor;` 语句所在行对应的代码左侧区中单击，可在 `EMP_LIST` 过程中设置一个断点。



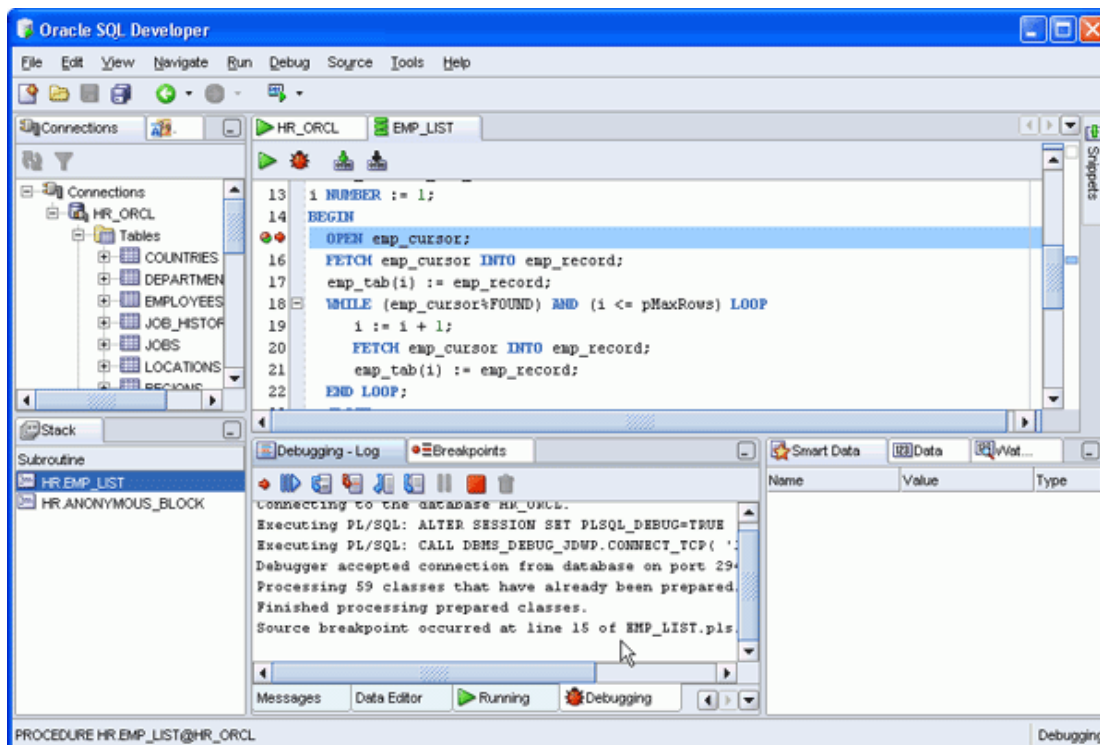
2. 单击 Debug 图标（瓢虫）。



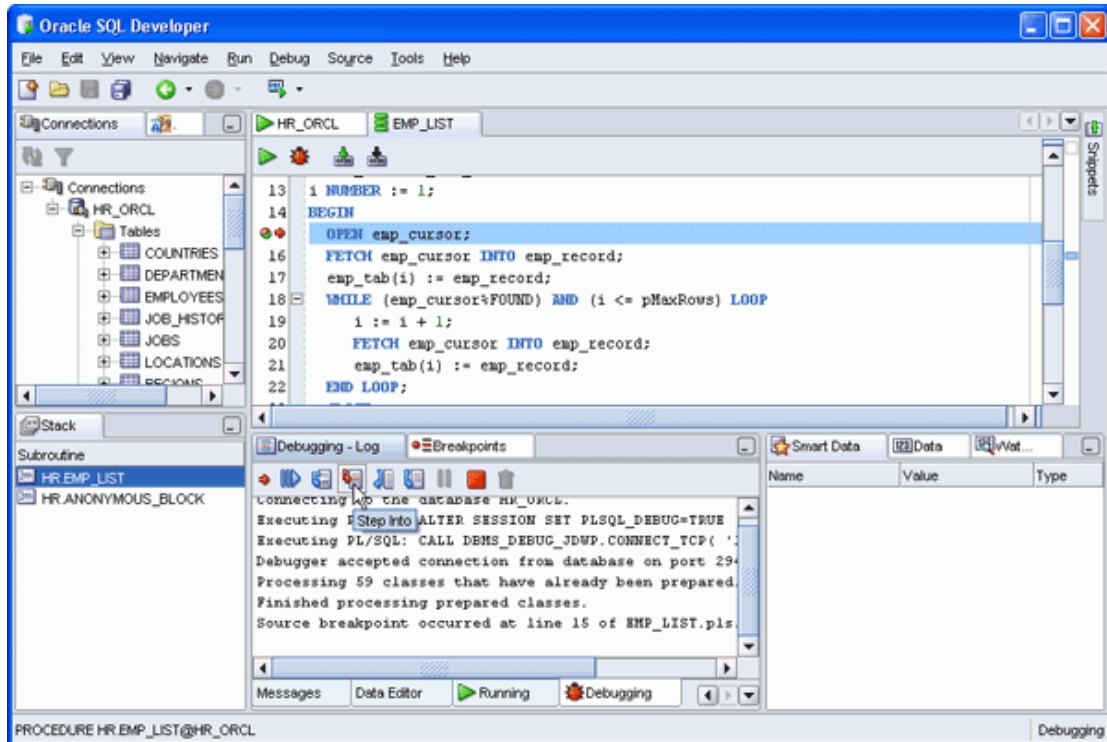
3. Debug PL/SQL 对话框应仍显示值 PMAXROWS = 5; 单击 OK。



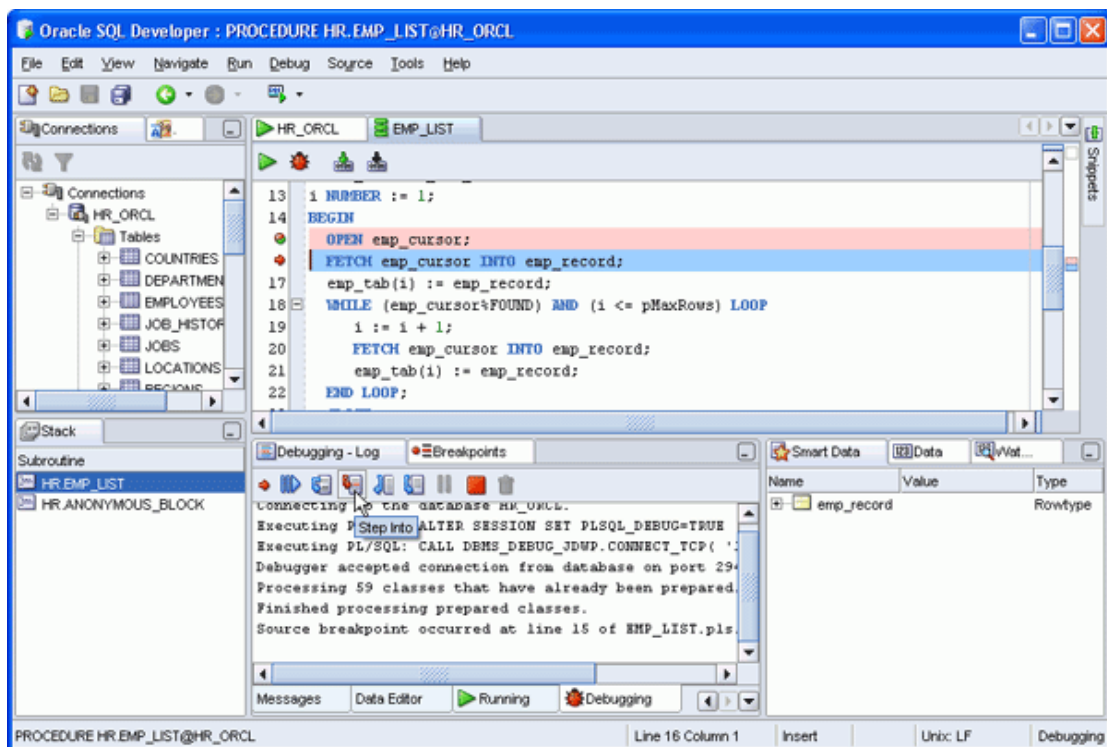
4. 调试器应在设置断点处暂停。可以控制执行流程、修改变量值以及执行其他调试函数。



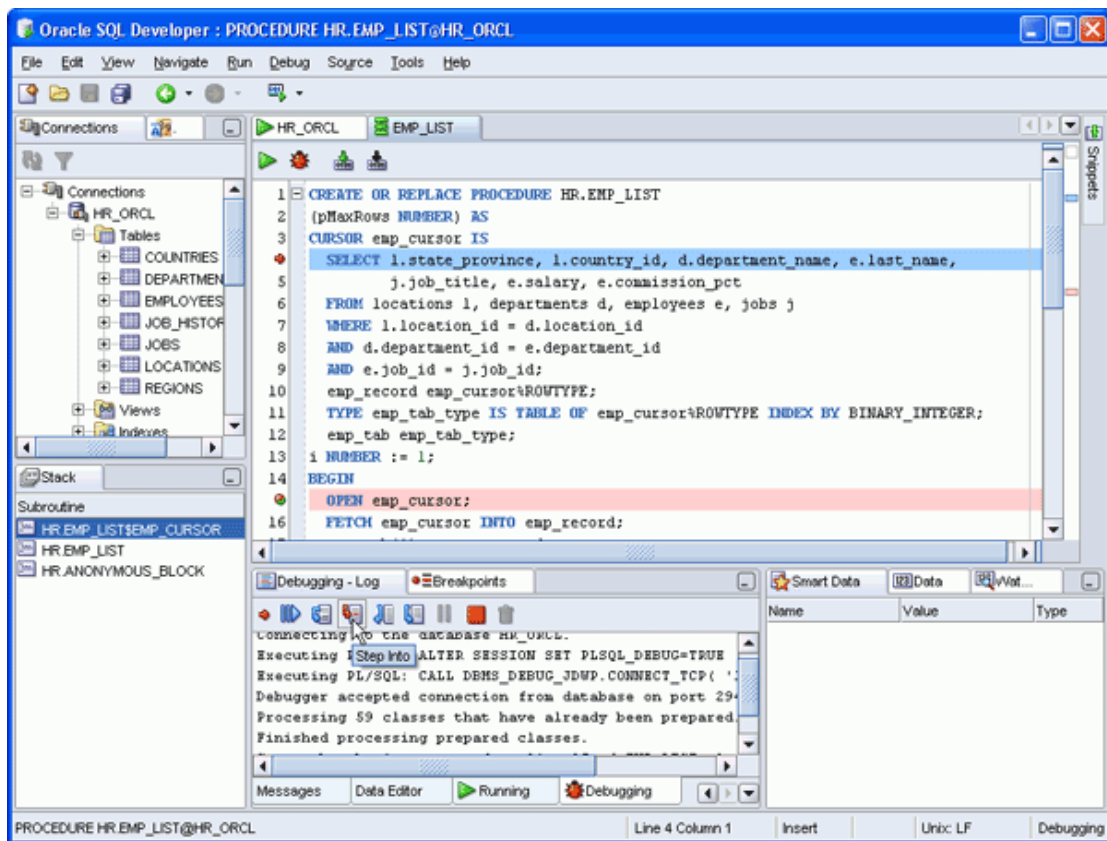
5. 单击 Step Into 。



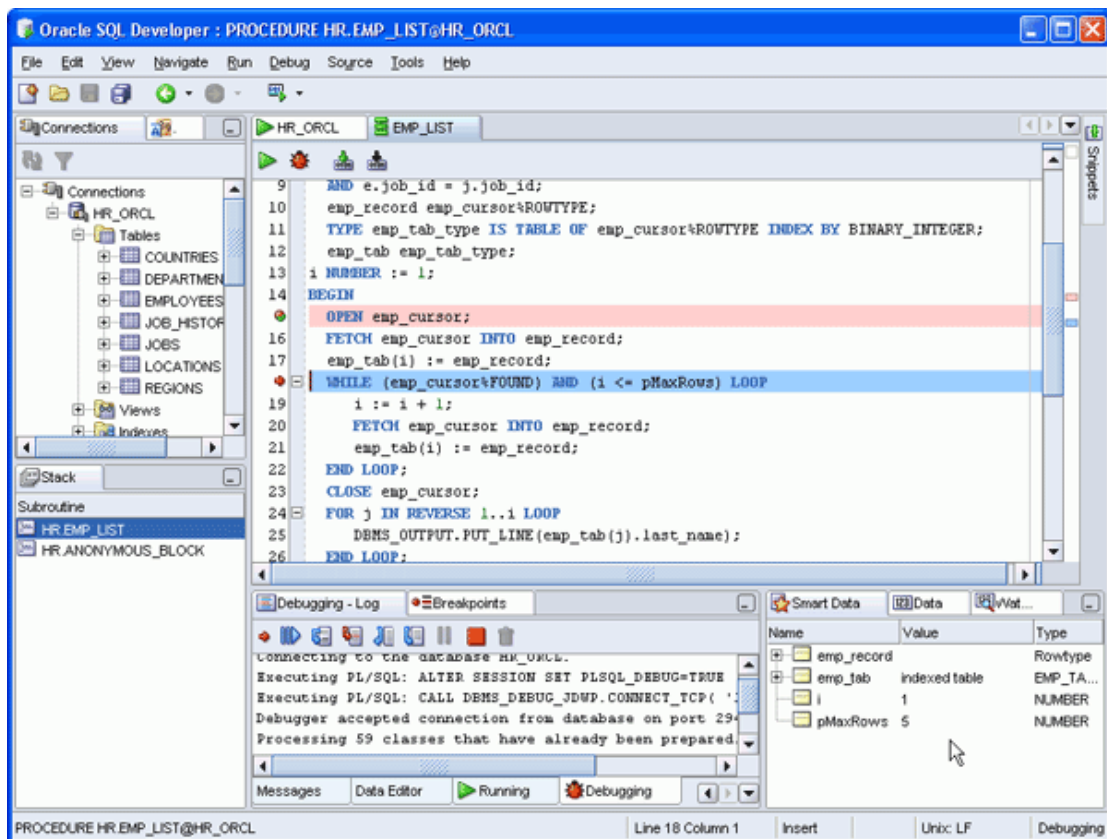
6. 这将转至光标所在的第一行。再次单击 Step Into 。



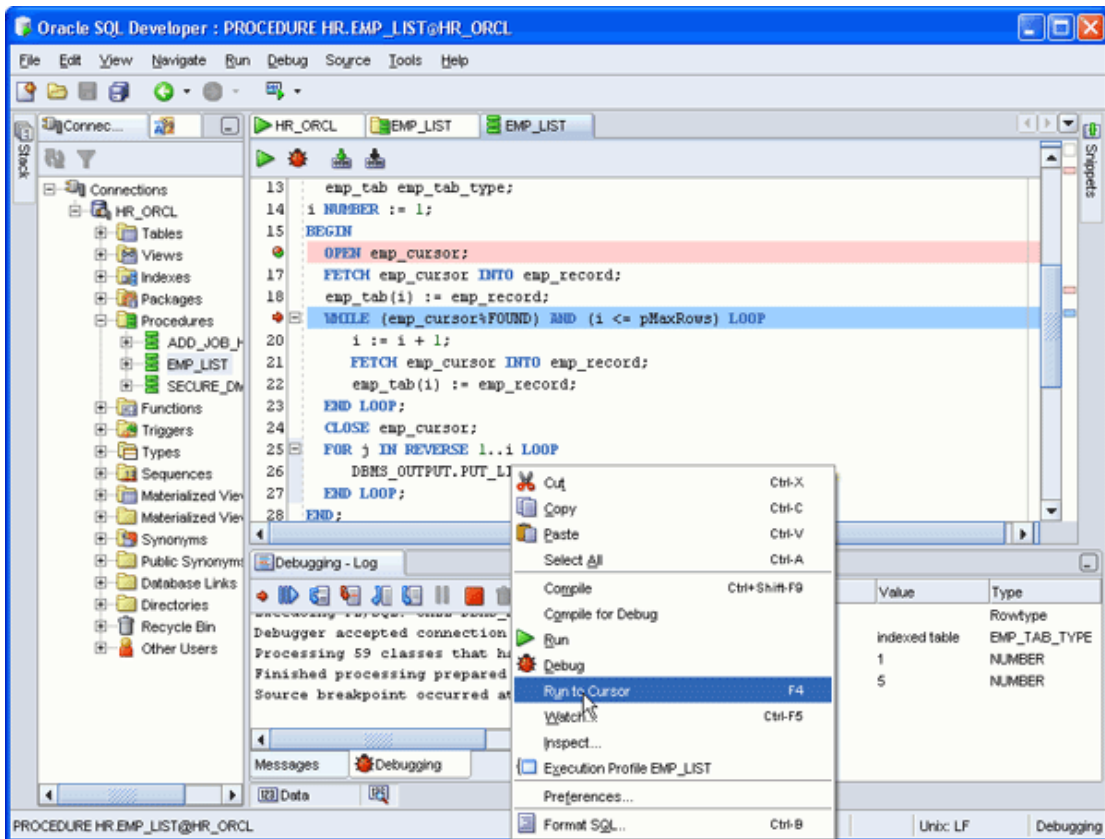
7. 现在应该选择光标所在的第一行。第 3 次单击 Step Into 。



8. Smart Data 窗口开始显示有限的变量列表，这些变量用在要执行的代码行中以及之前执行的代码中。

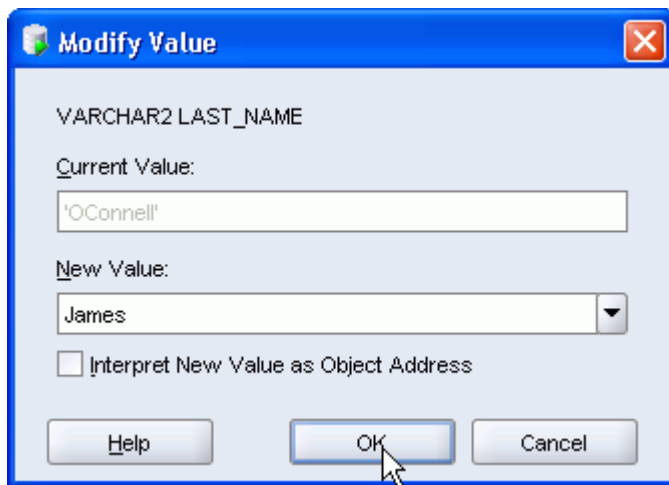


9. 右键单击 DBMS_OUTPUT.PUT_LINE(emp_tab(j).last_name); 这行，然后选择 Run to Cursor。

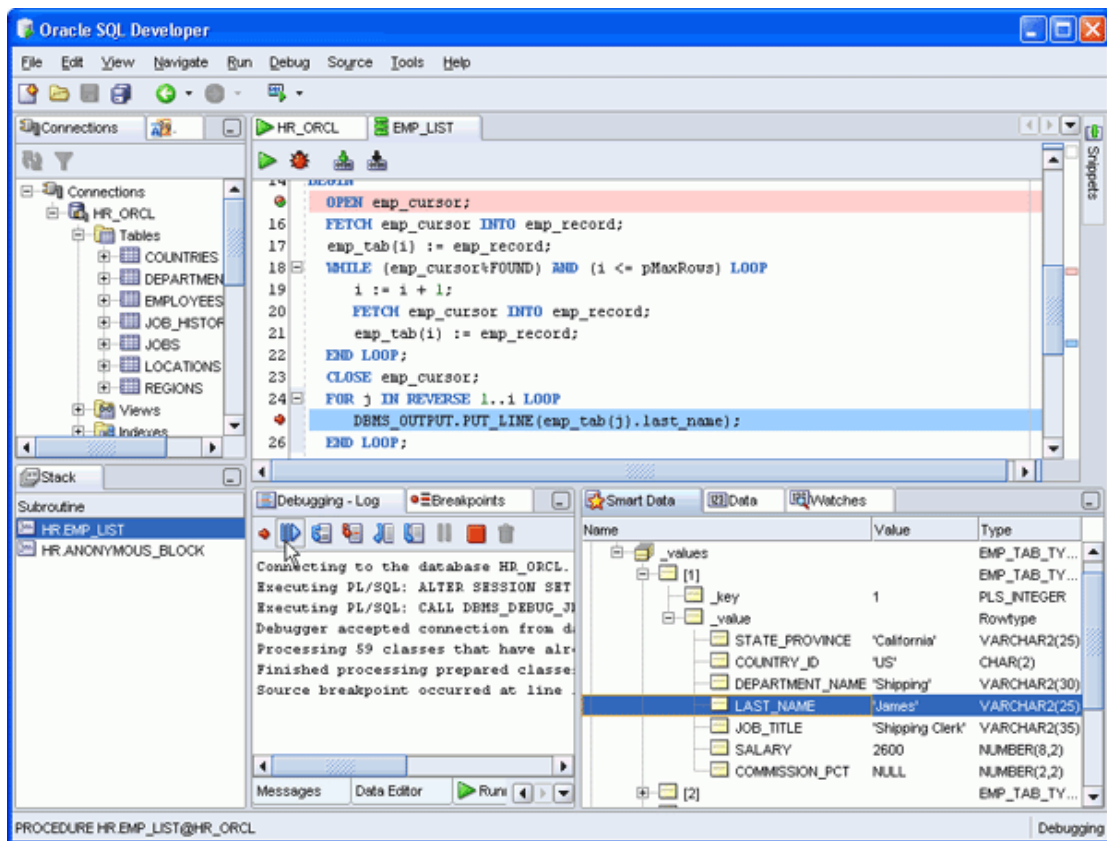


10. 展开 emp_tab > values > [1] > _key。会看到表中给定记录中域的值。选择 LAST_NAME 域。

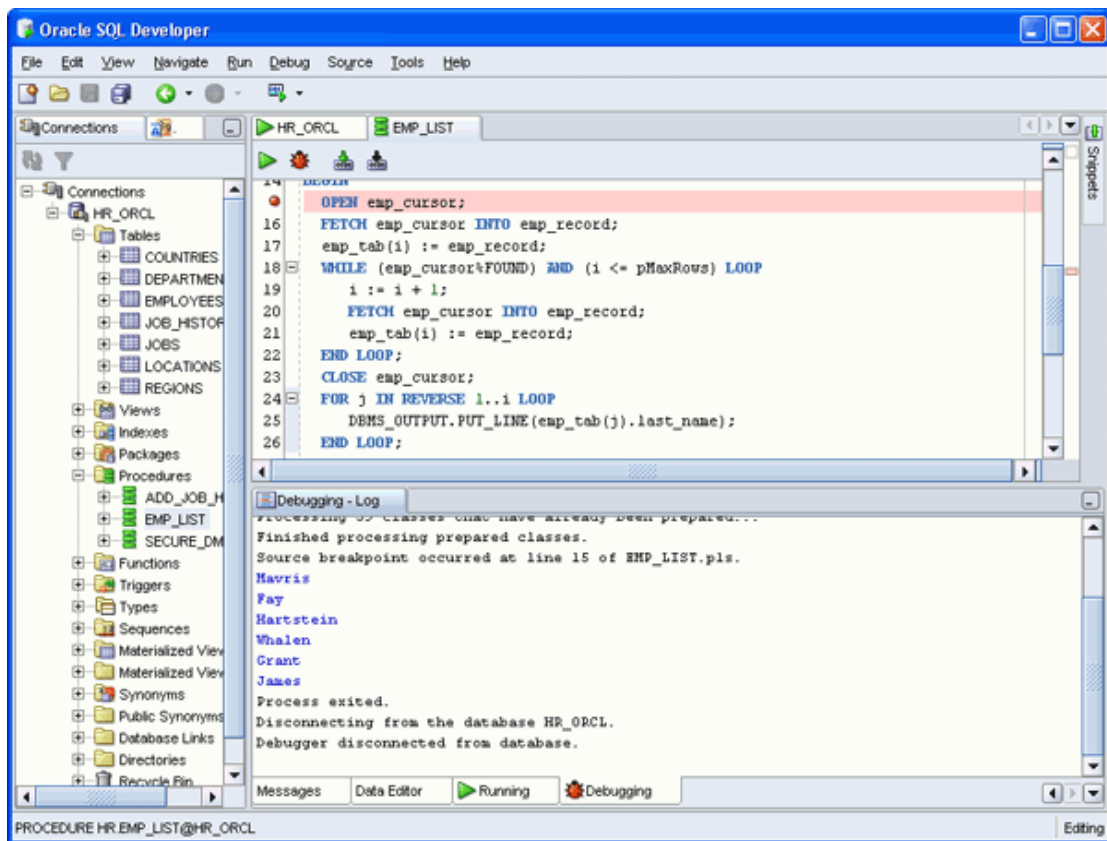
12. 将名称更改为别的内容，单击 OK。



13. 单击 Resume 图标允许 PL/SQL 运行直至完成。



14. 查看修改后的值是否显示在 Log 窗口中。

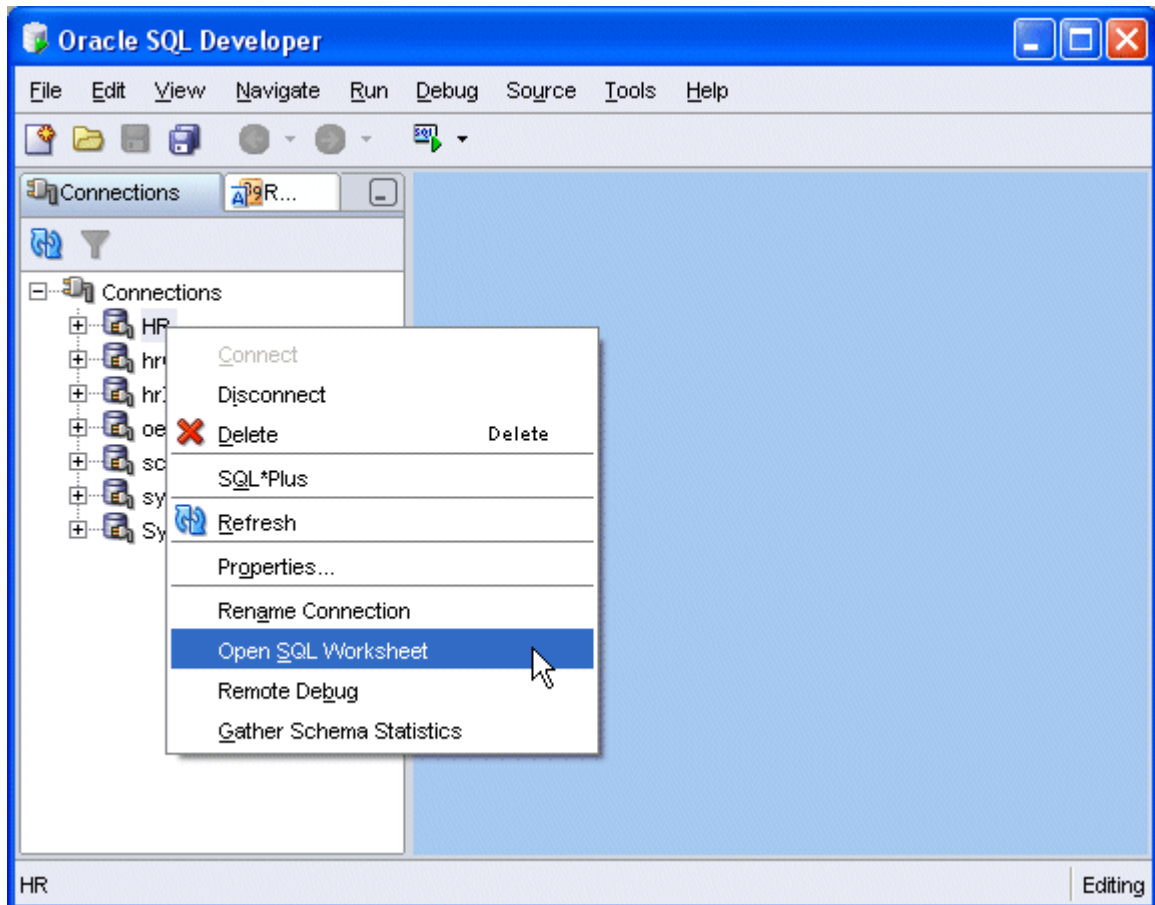


使用 SQL Worksheet

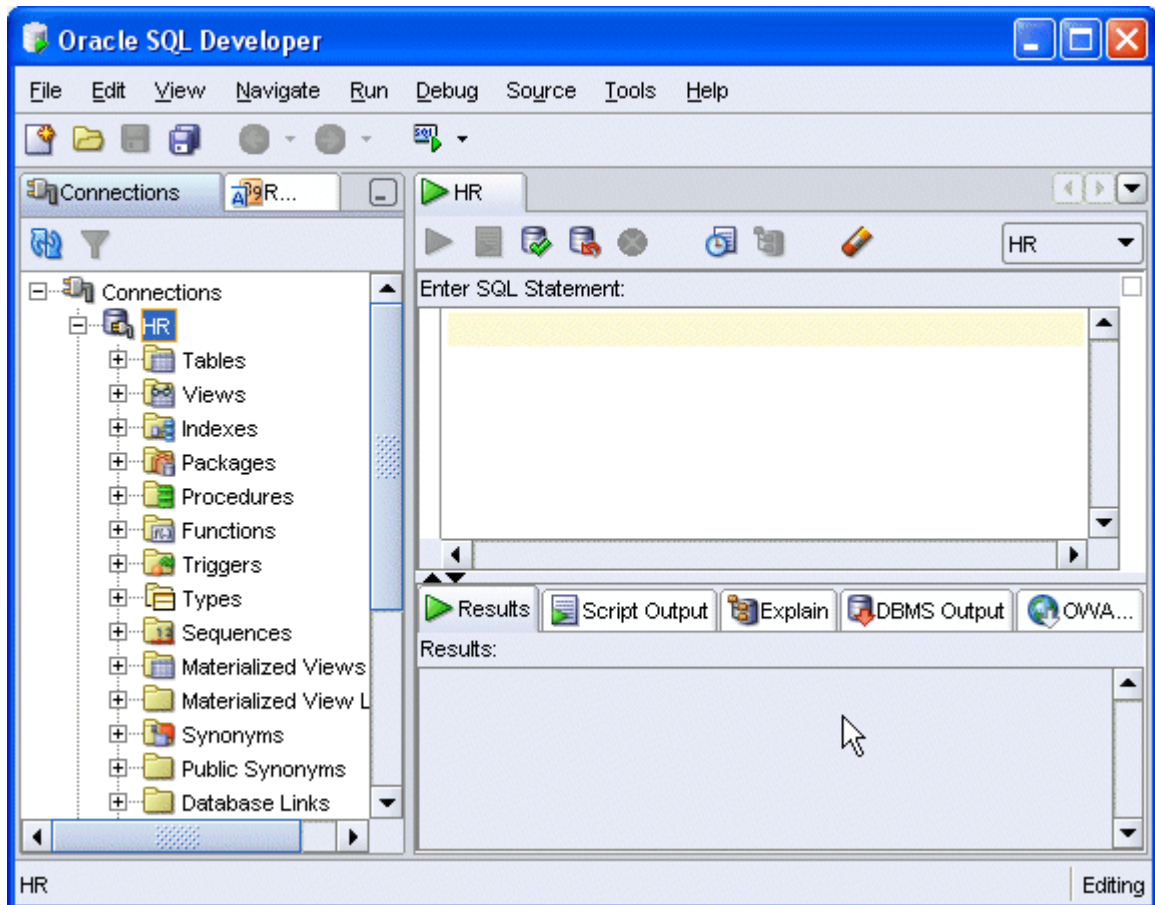
编写 Select 语句

连接到数据库之后，可以以浏览模式，查询和修改数据了。

1. 创建了数据库连接之后，需要确保 SQL Worksheet 是打开的。如果工作表尚未打开，则使用上下文菜单打开它。



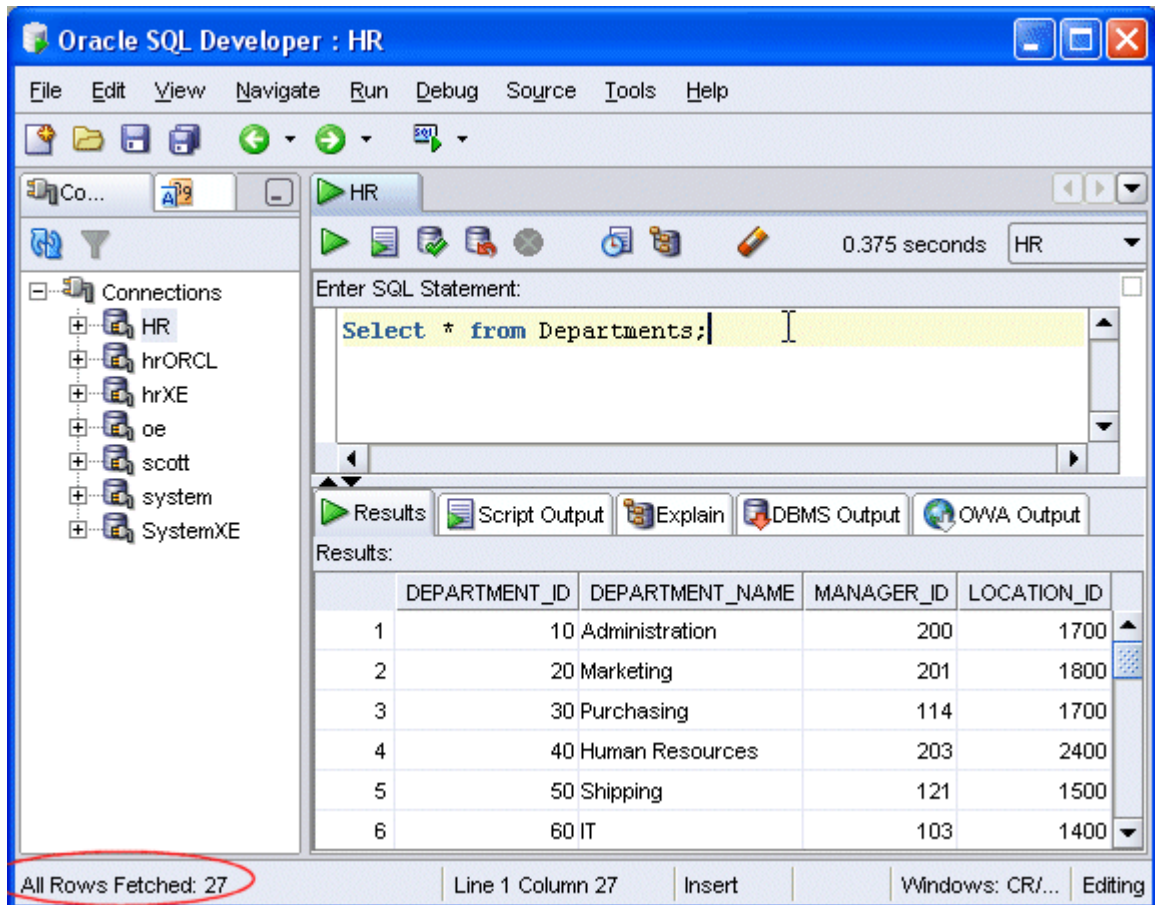
2. 连接之后，应当看到 SQL Worksheet Enter SQL Statement 窗口。



3. 查询 DEPARTMENTS 表中的所有数据。输入

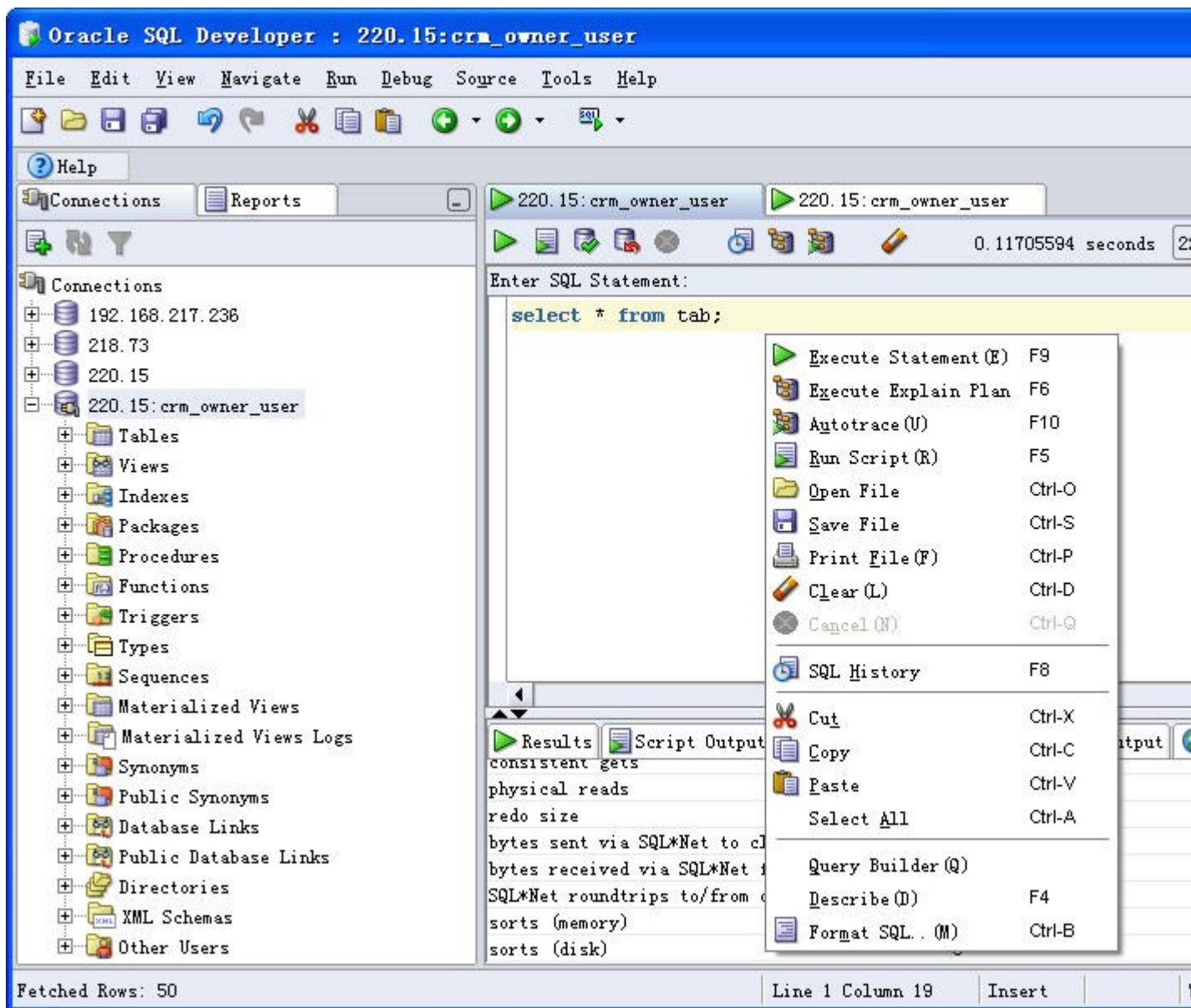
```
Select * from Departments;
```

按 F9



注意，检索到的行数显示在 Results 选项卡下方。

4、在 oracle sql Developer 中可以开启多个 Sql WorkSheet：在 Sql WorkSheet 中可以实现 Explain Plan 、 Run Script，和 SQL Histroy 等功能。

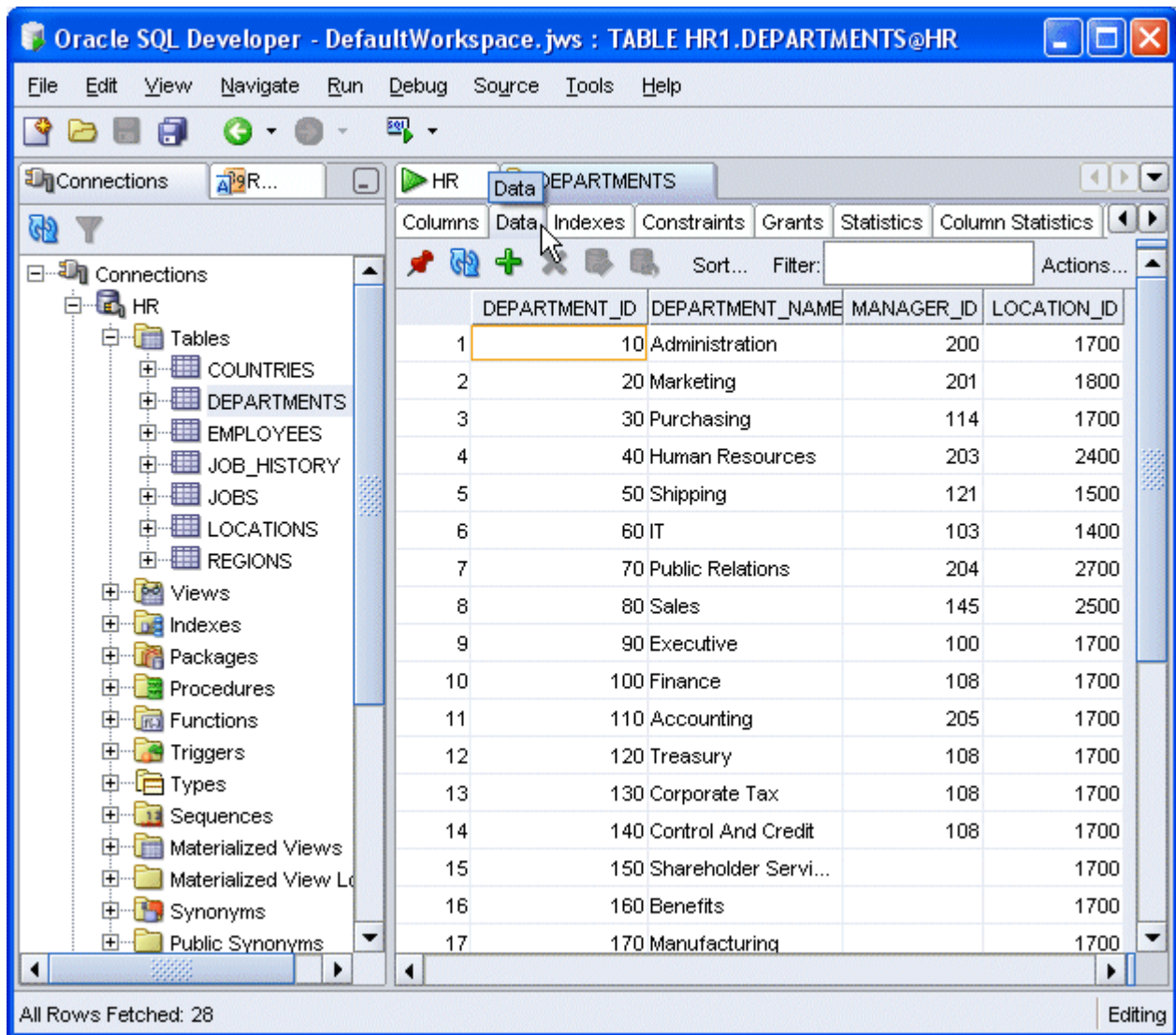


导出数据

SQL Developer 能够将用户数据导出为各种格式：CSV、XML、LOADER、TEXT 和 INSERT

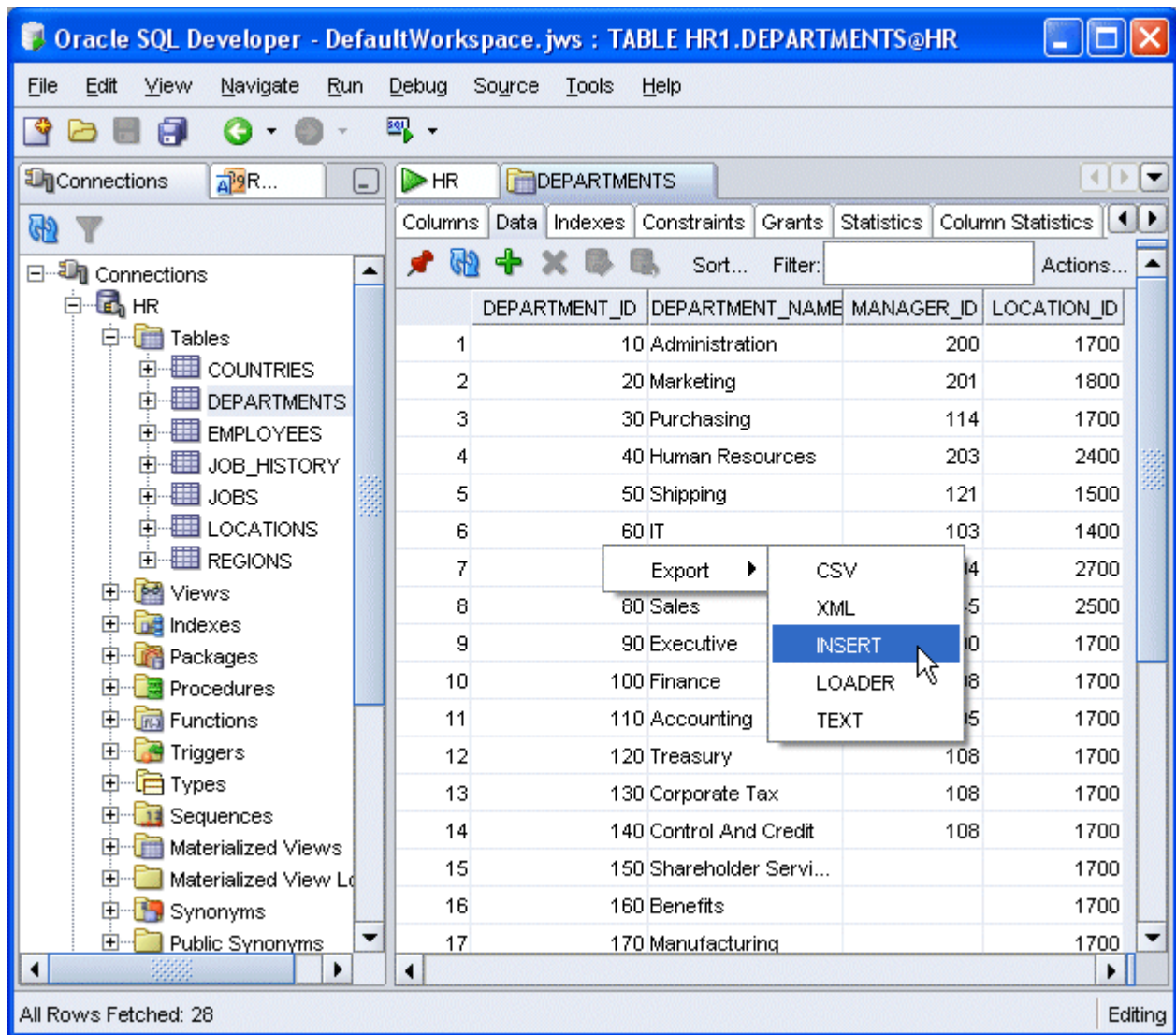
1. 为了导出表中的数据，展开 **Connections Navigator** 和 **Tables** 节点。

单击 **DEPARTMENTS**。选择 **Data** 选项卡。



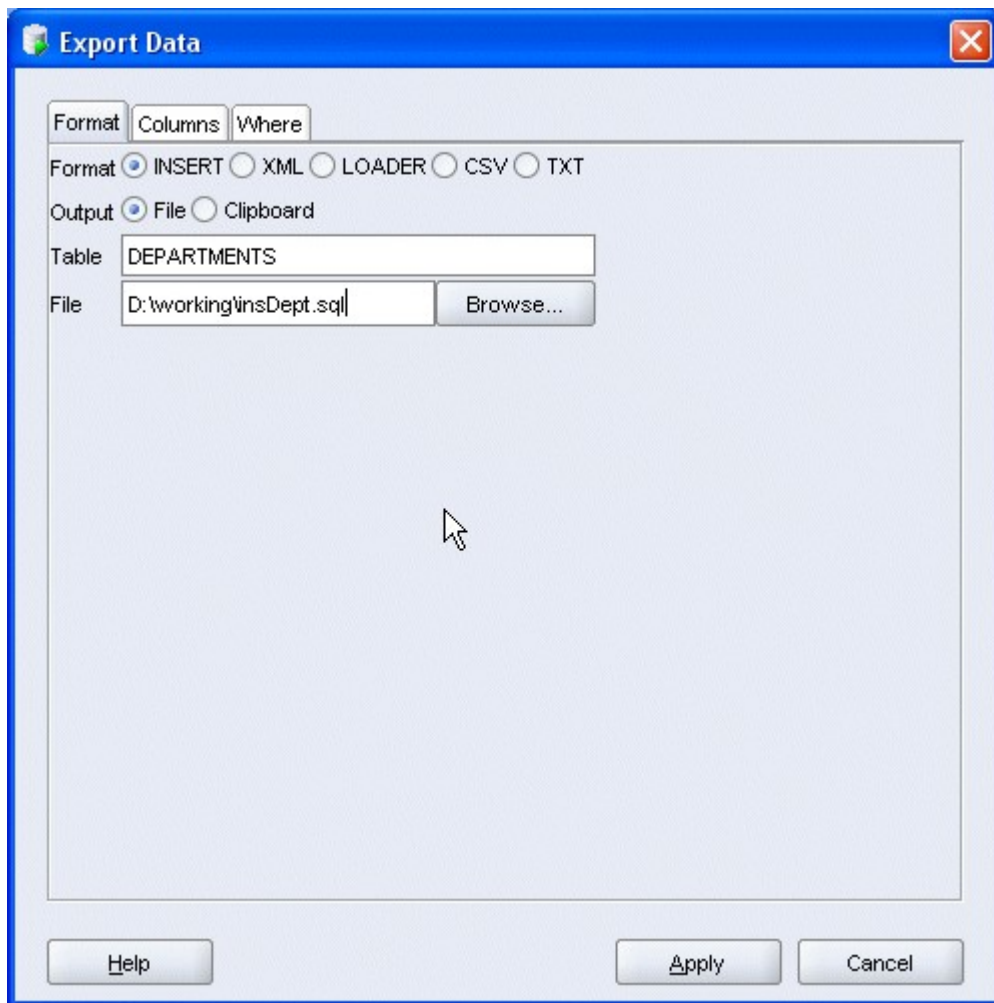
2. 右键单击数据的任何位置以调用上下文菜单。

从 EXPORT 上下文菜单中选择 **INSERT**。这将生成一个包含 Insert 语句的文件。



3. 选择适当的目录，并为该文件命名。

还可以通过选择列并添加 where 子句来减少返回的记录数，对导出的内容进行微调。



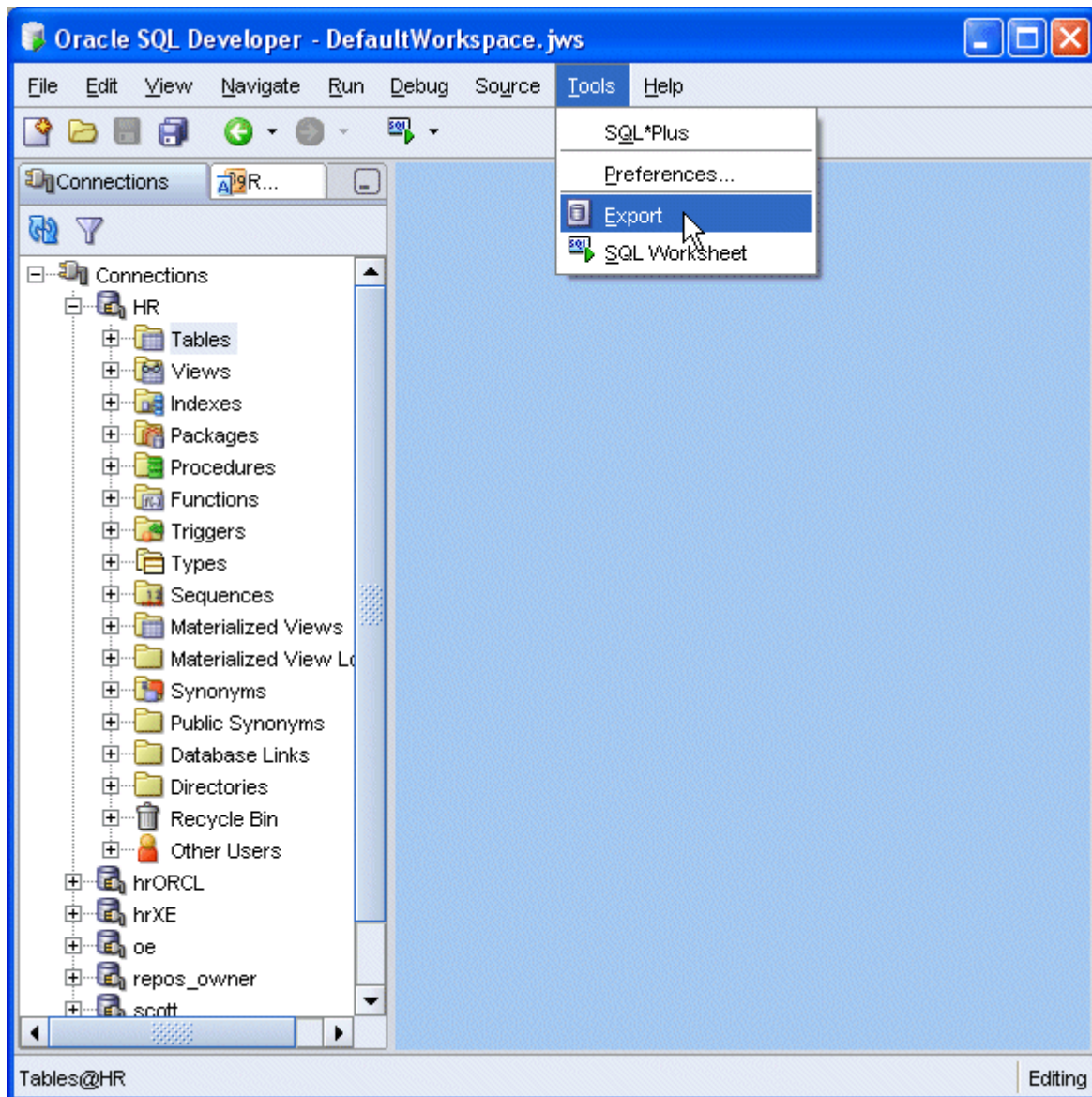
4. 单击 Apply 创建文件。

5. 查看创建的文件。可以为每个文件格式重复这些步骤，以观察所提供的不同格式。

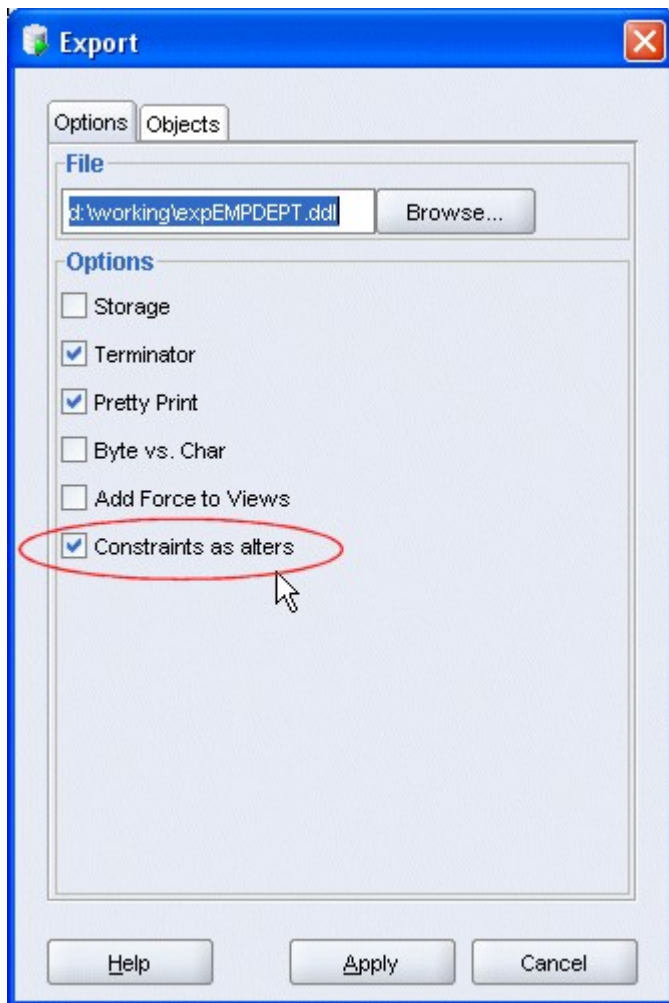
导出对象定义

SQL Developer 能够将部分或全部对象定义导出到脚本文件。可以将这些脚本文件用作对象定义的备份，或者在其他模式中运行它们。

1. 在主菜单中，选择 **Tools -> Export**。

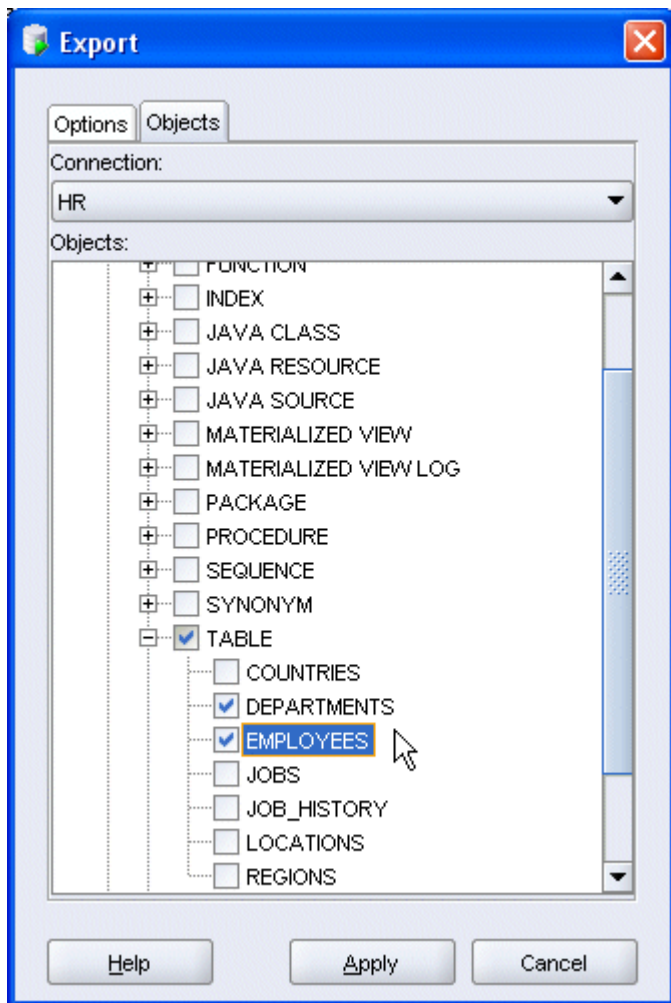


2. 在第一个选项卡中，确保提供了目录和文件名。



注意，可以选择内联添加约束，或者将约束作为“alter”表命令。

3. 切换选项卡，选择所需的对象定义。选择了 **EMPLOYEES** 和 **DEPARTMENTS**。



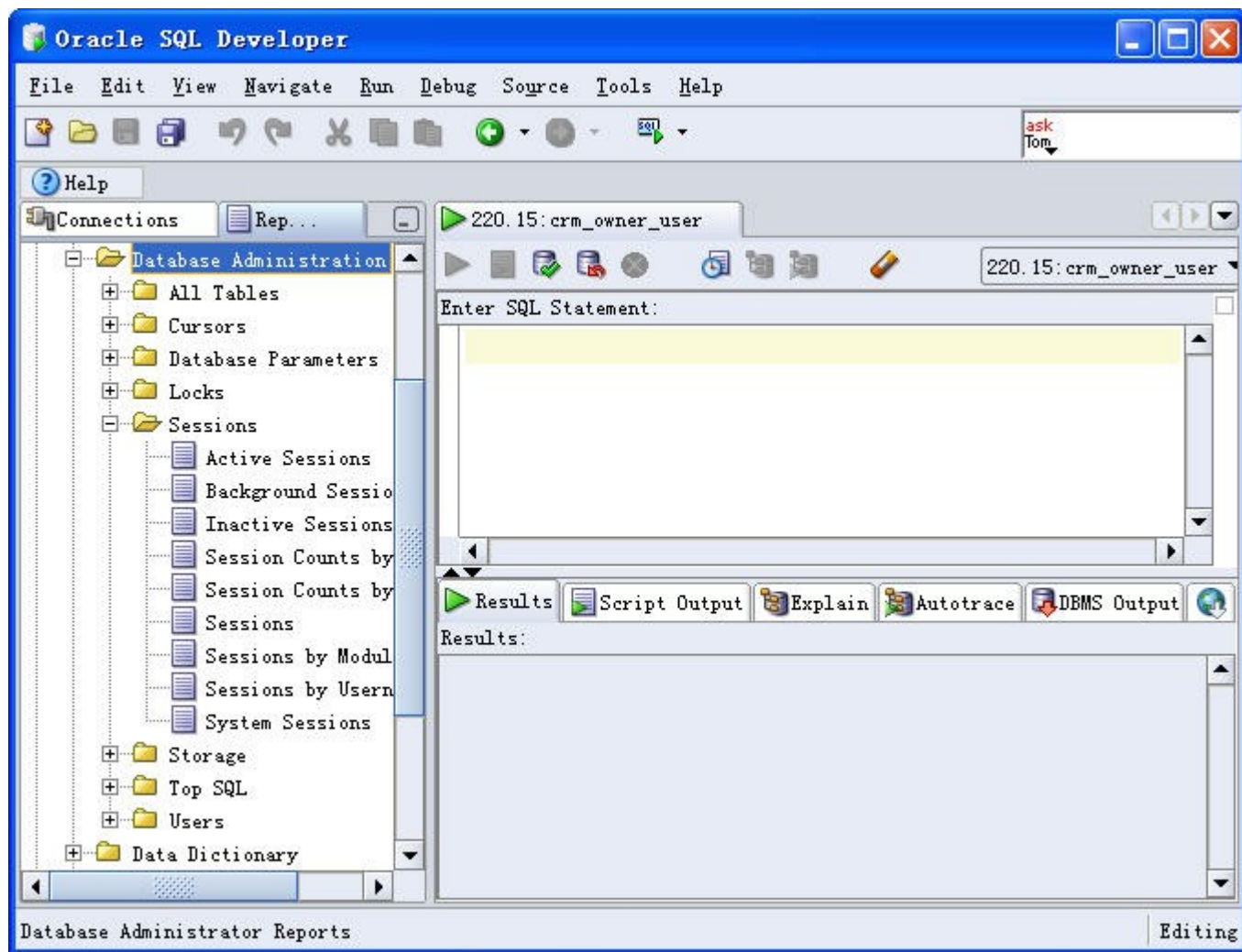
单击 **Apply**。

4. 查看创建的文件。

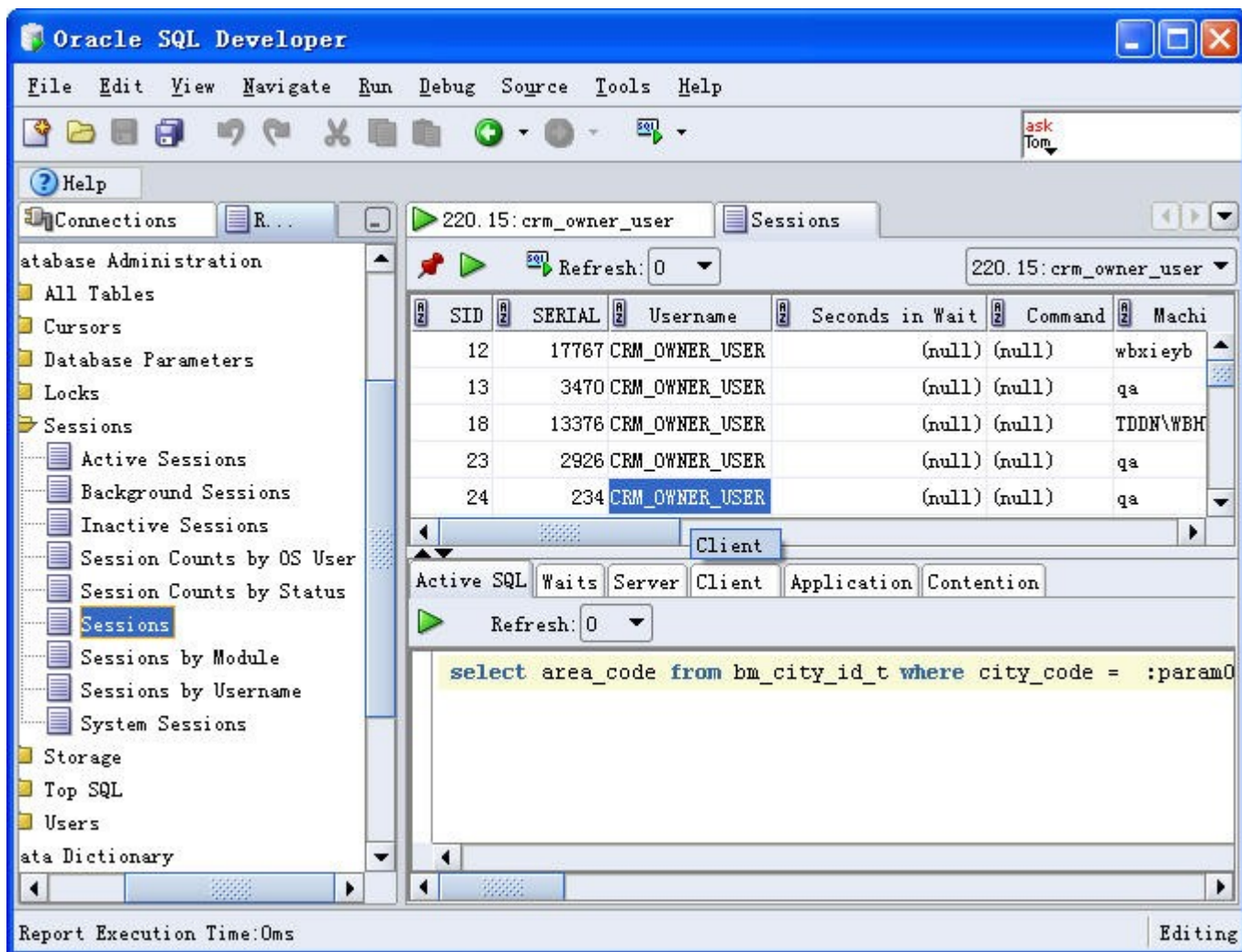
session 功能

Data Dictionary Report 中定义了一些预定义的查询内容。以下就我们常用的功能逐一讲解。

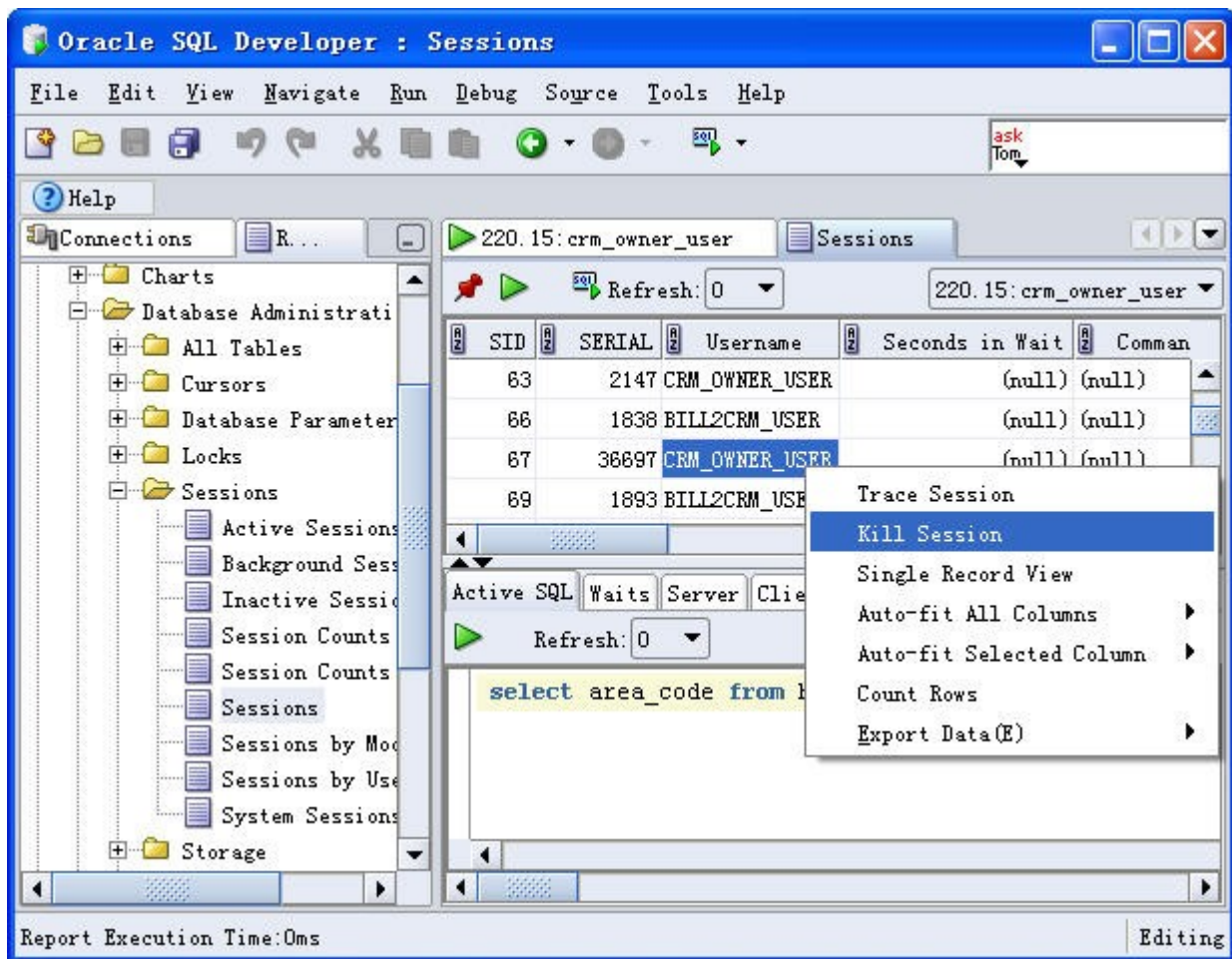
1. session browser 功能。打开 Data Dictionary Reports 节点。选择 Database Administration 节点打开。在 session 节点中存在可以观看 session 状态的报告。包括活动的 session 报告，后台 session 报告，非活动 session 报告，按照操作系统用户、session 状态、使用程序等统计的 session 数。选择 sasession，如下图：



2. 在 session 报告中可以选中某一个 session 进程，可以观察它的当前 SQL 等信息，等待事件等等。



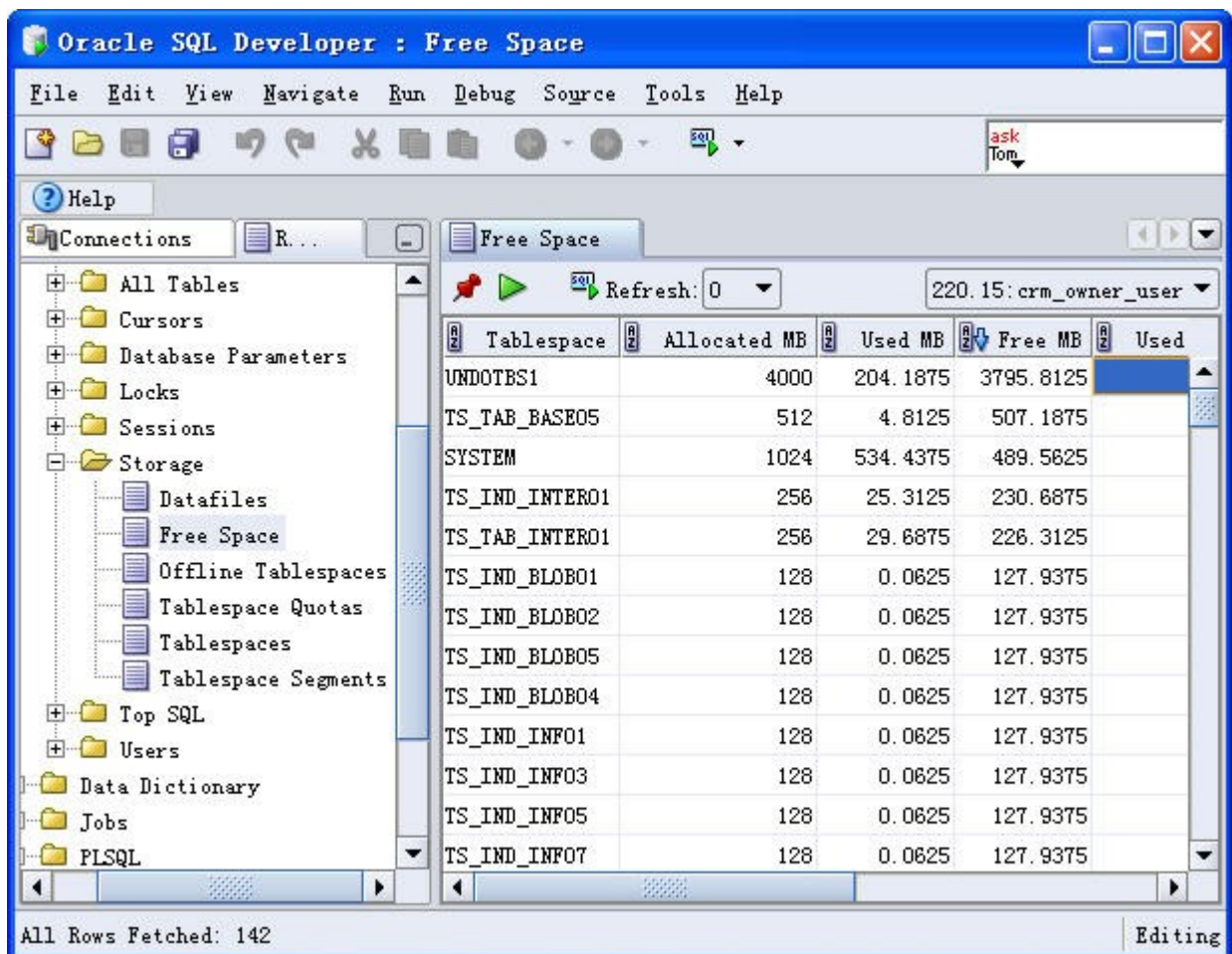
3. 选中一条 session 记录，右键单击，会出现一个小菜单，可以执行 kill session，trace session 等功能。



注意，从右上角可以识别使用了哪个连接、何时运行该报表。

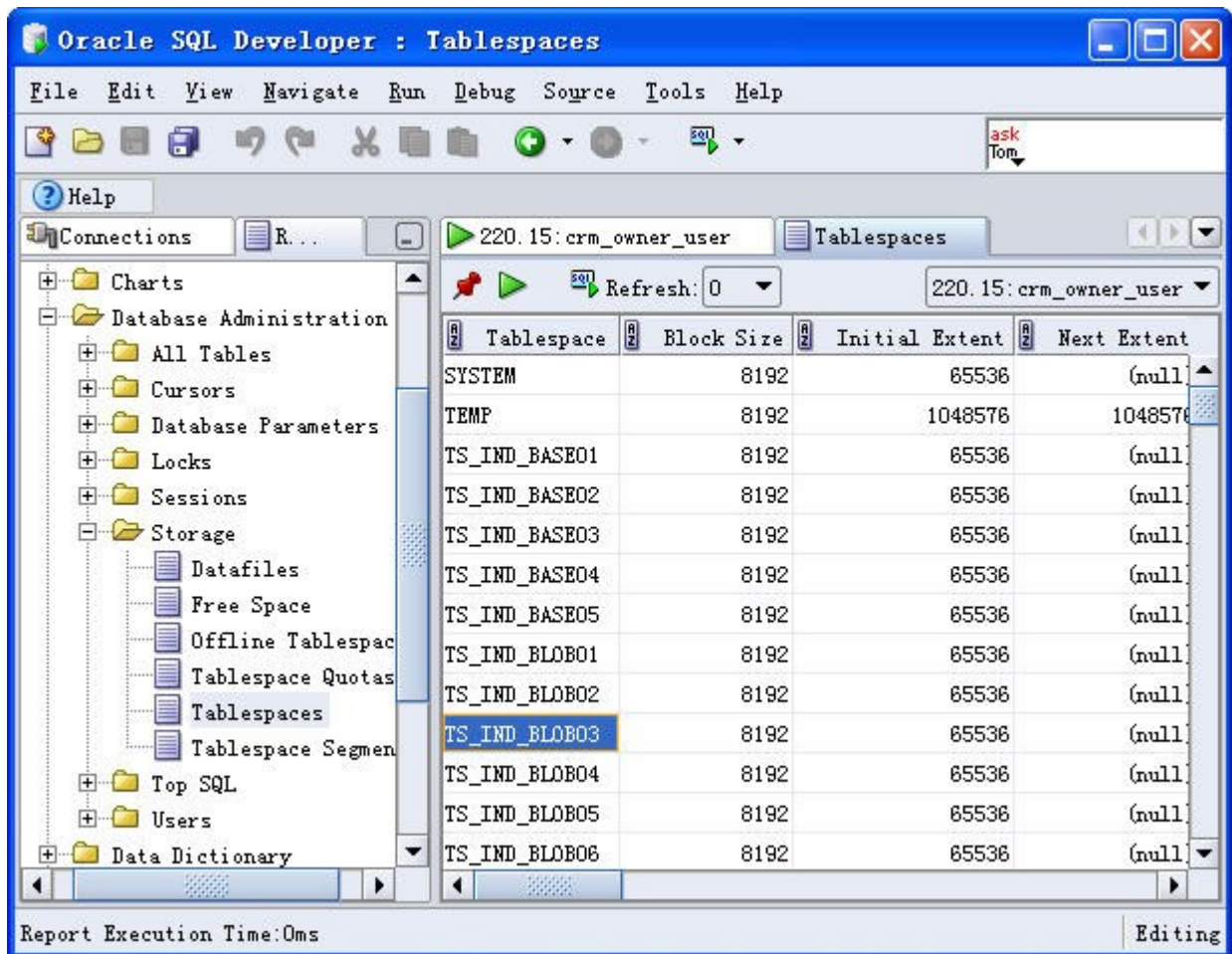
观察 TableSpace 功能

1、在 sql developer 中可以观察 tablespace 的使用情况以及 datafile 等信息。如下图：



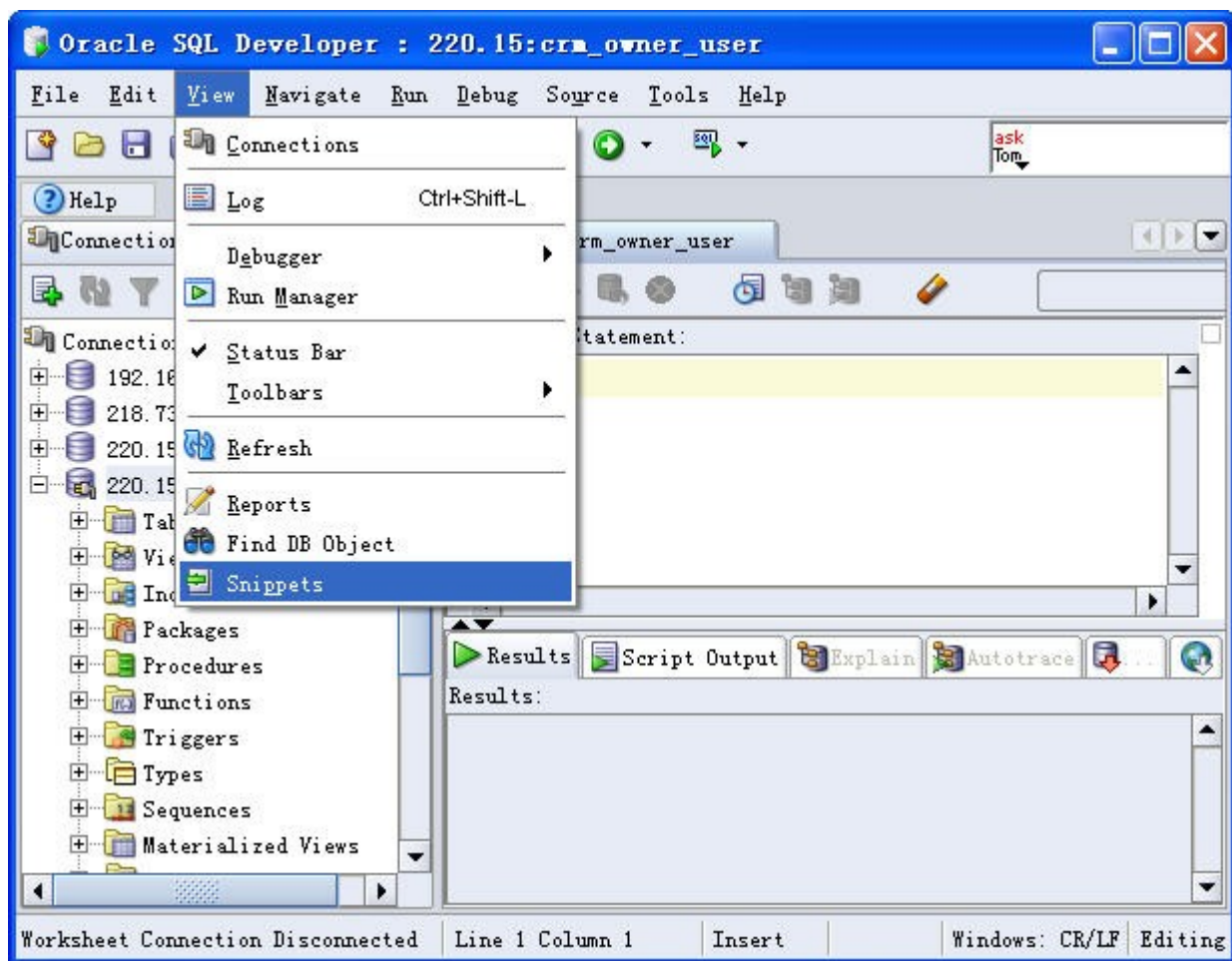
在上图中可以看到，可以对查询出来的结果进行双击排序。

下图中显示了 tablespace 的一些属性。

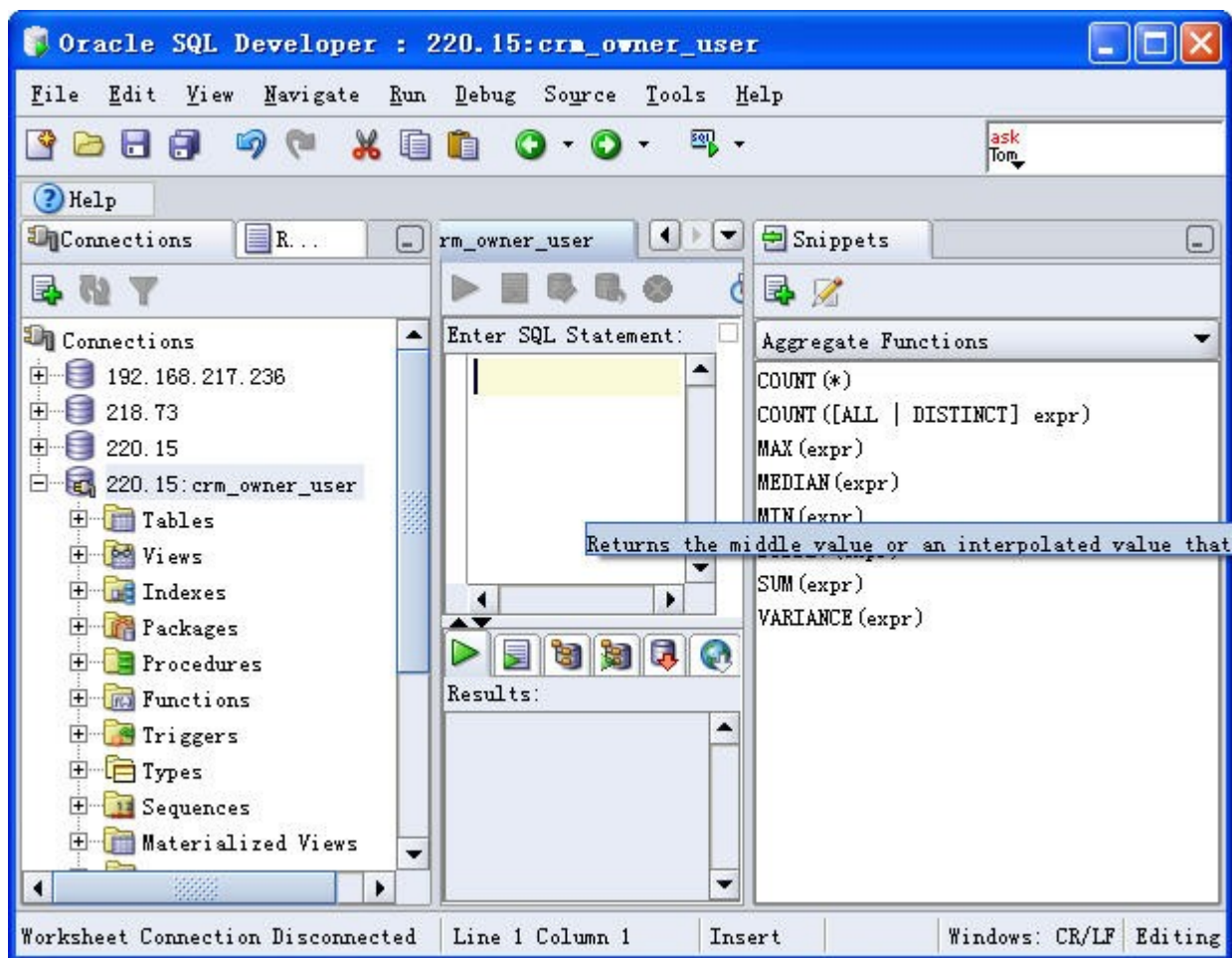


Sql Developer 中 code snippets 工具栏

如下图所示，在 view 菜单下，打开 snippets：



将鼠标停留在函数上就会出现函数的说明及用法。如下图：



在 snippets 中出现的函数可以采用拖拽方式拖拽到 Sql WorkSheet 中。

连接 MySQL

环境:

RHEL 5

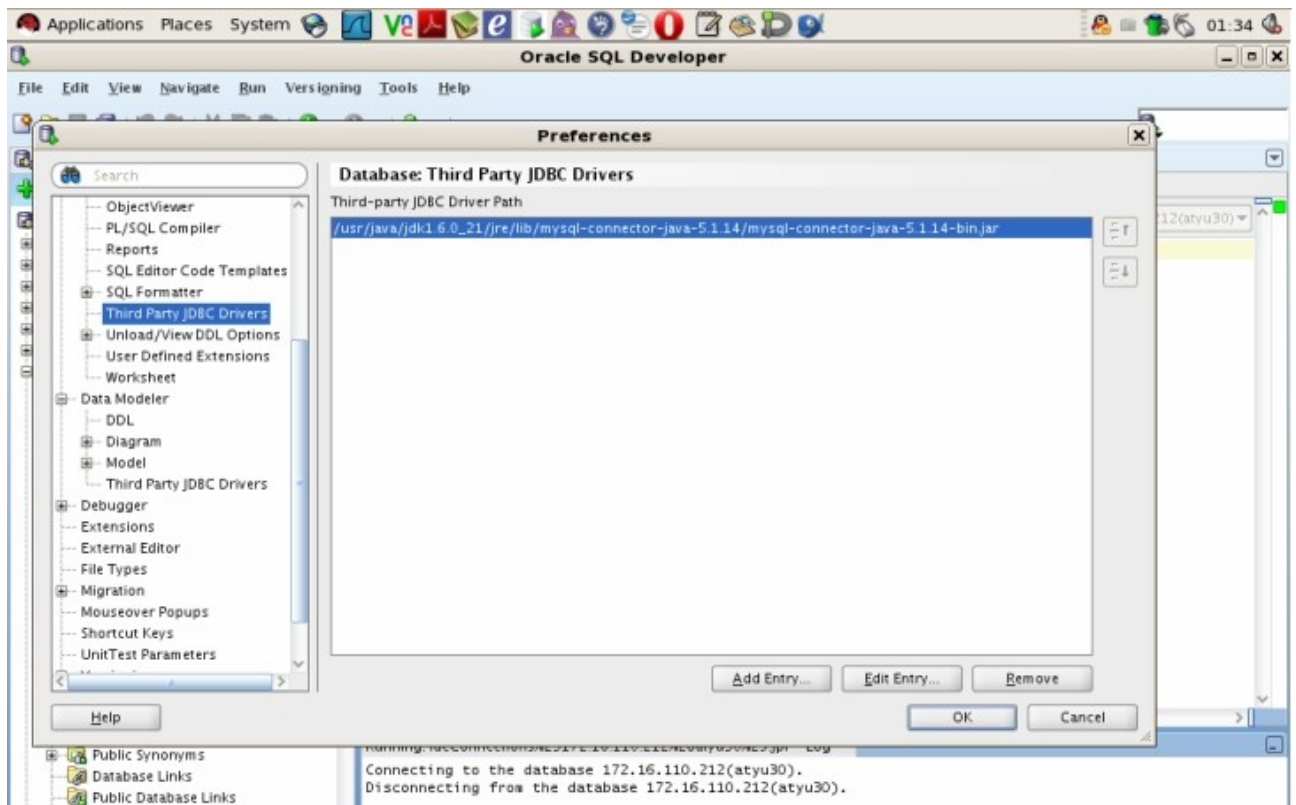
sqldeveloper-3.0.02.83-1.noarch.rpm

mysql-connector-java-5.1.14.tar.gz

*<http://mysql.oss.eznetsols.org/Downloads/Connector-J/mysql-connector-java-5.1.14.tar.gz>

安装 mysql 驱动:

```
# tar -zcf mysql-connector-java-5.1.14.tar.gz -C /usr/java/jdk1.6.0_21/jre/lib/
```

使用:

