Practices for Lesson 5: Using SQL Statements within a PL/SQL Block

Chapter 5

Practice 5: Using SQL Statements Within a PL/SQL

Note: If you have executed the code examples for this lesson, make sure that you execute the following code before starting this practice:

```
DROP table employees2;
DROP table copy_emp;
```

In this practice, you use PL/SQL code to interact with the Oracle Server.

1. Create a PL/SQL block that selects the maximum department ID in the `departments` table and stores it in the `v_max_deptno` variable. Display the maximum department ID.
   a. Declare a variable `v_max_deptno` of type `NUMBER` in the declarative section.
   b. Start the executable section with the `BEGIN` keyword and include a `SELECT` statement to retrieve the maximum `department_id` from the `departments` table.
   c. Display `v_max_deptno` and end the executable block.
   d. Execute and save your script as `lab_05_01_soln.sql`. The sample output is as follows:

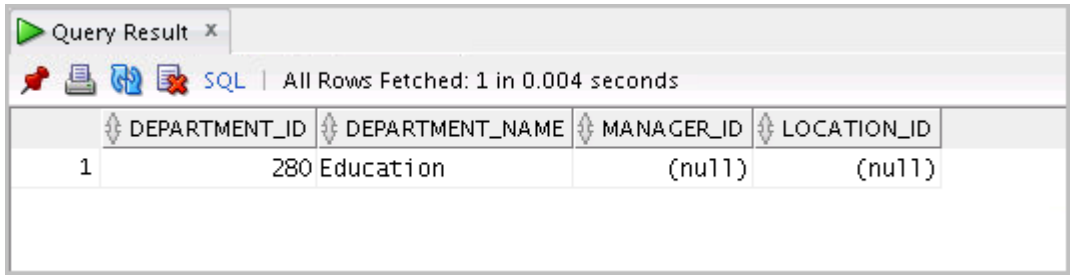
```
PL/SQL procedure successfully completed.

The maximum department_id is : 270
```

2. Modify the PL/SQL block that you created in step 1 to insert a new department into the `departments` table.
   a. Load the `lab_05_01_soln.sql` script. Declare two variables: `v_dept_name` of type `departments.department_name` and `v_dept_id` of type `NUMBER`.
   Assign 'Education' to `v_dept_name` in the declarative section.
   b. You have already retrieved the current maximum department number from the `departments` table. Add 10 to it and assign the result to `v_dept_id`.
   c. Include an `INSERT` statement to insert data into the `department_name`, `department_id`, and `location_id` columns of the `departments` table.
   Use the values in `v_dept_name` and `v_dept_id` for `department_name` and `department_id`, respectively, and use `NULL` for `location_id`.
   d. Use the SQL attribute `SQL%ROWCOUNT` to display the number of rows that are affected.
   e. Execute a `SELECT` statement to check whether the new department is inserted. You can terminate the PL/SQL block with "`/`" and include the `SELECT` statement in your script.
   f. Execute and save your script as `lab_05_02_soln.sql`. The sample output is as follows:

```
PL/SQL procedure successfully completed.

The maximum department_id is : 270
 SQL%ROWCOUNT gives 1
```



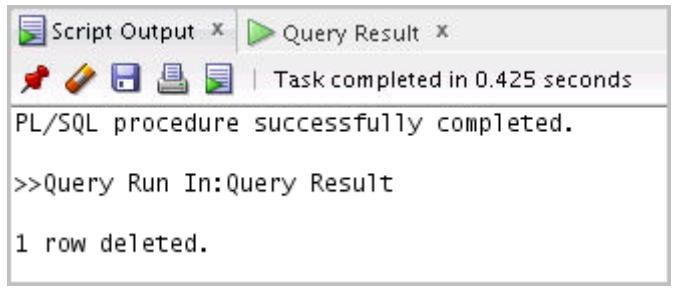
3. In step 2, you set `location_id` to `NULL`. Create a PL/SQL block that updates `location_id` to `3000` for the new department.
   Note: If you successfully completed step 2, continue with step 3a. If not, first execute the solution script `/soln/sol_05.sql`. (Task 2 in `sol_05.sql`)

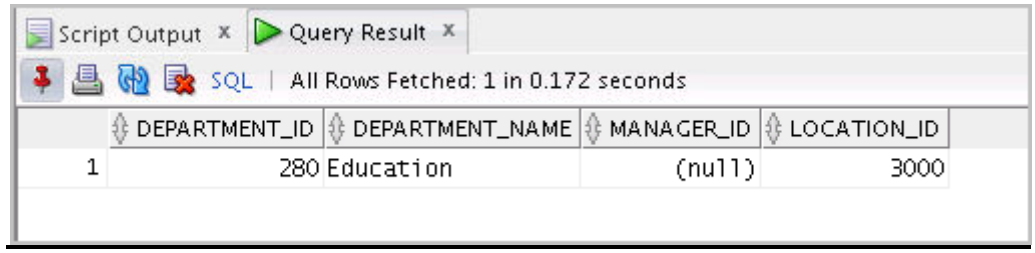
   a. Start the executable block with the `BEGIN` keyword. Include the `UPDATE` statement to set `location_id` to `3000` for the new department (`v_dept_id` =280).

   b. End the executable block with the `END` keyword. Terminate the PL/SQL block with "`/`" and include a `SELECT` statement to display the department that you updated.

   c. Include a `DELETE` statement to delete the department that you added.

   d. Execute and save your script as `lab_05_03_soln.sql`. The sample output is as follows:





Solution 5: Using SQL Statements Within a PL/SQL

In this practice, you use PL/SQL code to interact with the Oracle Server.

1. Create a PL/SQL block that selects the maximum department ID in the `departments` table and stores it in the `v_max_deptno` variable. Display the maximum department ID.
   a. Declare a variable `v_max_deptno` of type `NUMBER` in the declarative section.

```
DECLARE
   v_max_deptno  NUMBER;
```

   b. Start the executable section with the `BEGIN` keyword and include a `SELECT` statement to retrieve the maximum `department_id` from the `departments` table.

```
BEGIN
   SELECT MAX(department_id)  INTO v_max_deptno  FROM
     departments;
```

   c. Display `v_max_deptno` and end the executable block.

```
DBMS_OUTPUT.PUT_LINE('The maximum department_id is : ' ||
v_max_deptno);
END;
```

   d. Execute and save your script as `lab_05_01_soln.sql`. The sample output is as follows:

```
PL/SQL procedure successfully completed.

The maximum department_id is : 270
```

2. Modify the PL/SQL block that you created in step 1 to insert a new department into the `departments` table.
   a. Load the `lab_05_01_soln.sql` script. Declare two variables: `v_dept_name` of type `departments.department_name` and `v_dept_id` of type `NUMBER`.
   Assign 'Education' to `v_dept_name` in the declarative section.

```
  v_dept_name departments.department_name%TYPE:= 'Education';
  v_dept_id NUMBER;
```

   b. You have already retrieved the current maximum department number from the `departments` table. Add 10 to it and assign the result to `v_dept_id`.

```
v_dept_id := 10 + v_max_deptno;
```

c. Include an INSERT statement to insert data into the department_name, department_id, and location_id columns of the departments table. Use the values in v_dept_name and v_dept_id for department_name and department_id, respectively, and use NULL for location_id.

```
…
INSERT INTO departments (department_id, department_name,
location_id)
VALUES (v_dept_id, v_dept_name, NULL);
```

d. Use the SQL attribute SQL%ROWCOUNT to display the number of rows that are affected.

```
DBMS_OUTPUT.PUT_LINE (' SQL%ROWCOUNT gives ' || SQL%ROWCOUNT);
…
```

e. Execute a SELECT statement to check whether the new department is inserted. You can terminate the PL/SQL block with "/" and include the SELECT statement in your script.

```
…
/
SELECT * FROM  departments  WHERE  department_id= 280;
```

f. Execute and save your script as lab_05_02_soln.sql. The sample output is as follows:

```
PL/SQL procedure successfully completed.

The maximum department_id is : 270
 SQL%ROWCOUNT gives 1
```

Query Result

SQL | All Rows Fetched: 1 in 0.004 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	280	Education	(null)	(null)

3. In step 2, you set location_id to NULL. Create a PL/SQL block that updates the location_id to 3000 for the new department.
Note: If you successfully completed step 2, continue with step 3a. If not, first execute the solution script /soln/sol_05.sql. (Task 2 in sol_05.sql)

a. Start the executable block with the BEGIN keyword. Include the UPDATE statement to set location_id to 3000 for the new department (v_dept_id =280).

```
BEGIN
```

```
UPDATE departments SET location_id=3000 WHERE
department_id=280;
```

b. End the executable block with the `END` keyword. Terminate the PL/SQL block with "/" and include a `SELECT` statement to display the department that you updated.

```
END;
/
SELECT * FROM departments WHERE department_id=280;
```

c. Include a `DELETE` statement to delete the department that you added.

```
DELETE FROM departments WHERE department_id=280;
```

d. Execute and save your script as `lab_05_03_soln.sql`. The sample output is as follows: