Database Programming with PL/SQL

3-3: Manipulating Data in PL/SQL Practice

Activities

Vocabulary

IMPLICIT CURSOR	Defined automatically by Oracle for all SQL data manipulation statements, and for queries that return only one row.
EXPLICIT CURSOR	Defined by the programmer for queries that return more than one row.
MERGE	Statement selects rows from one table to update and/or insert into another table. The decision whether to update or insert into the target table is based on a condition in the ON clause.
INSERT	Statement adds new rows to the table.
DELETE	Statement removes rows from the table.
UPDATE	Statement modifies existing rows in the table.

Identify the vocabulary word for each definition below:

Try It / Solve It

1. True or False: When you use DML in a PL/SQL block, Oracle uses explicit cursors to track the data changes. **TRUE**

2. SQL%FOUND, SQL%NOTFOUND, and SQL%ROWCOUNT are __________**ATTRIBUTES** ________ and are available when you use ________**IMPLICIT**_______ cursors.

The following questions use a copy of the departments table. Execute the following SQL statement to create the copy table.

```
CREATE TABLE new_depts AS SELECT * FROM departments;
```

3. Examine and run the following PL/SQL code, which obtains and displays the maximum department_id from new_depts. What is the maximum department id? **190**

```
DECLARE
  v_max_deptno new_depts.department_id%TYPE;
BEGIN
  SELECT MAX(department_id) INTO v_max_deptno
    FROM new_depts;
  DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);
END;
```

4. Modify the code to declare two additional variables (assigning a new department name to one of them), by adding the following two lines to your Declaration section:

```
v_dept_name    new_depts.department_name%TYPE := 'A New Department'; v_dept_id
    new_depts.department_id%TYPE;
 DECLARE

   v_dept_name    new_depts.department_name%TYPE := 'A New Department';

   v_dept_id        new_depts.department_id%TYPE;

   v_max_deptno new_depts.department_id%TYPE;

BEGIN

   SELECT MAX(department_id) INTO v_max_deptno

     FROM new_depts;

   DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);

END;
```

5. Modify the code to add 10 to the current maximum department number and assign the result to v_dept_id.

```
DECLARE

   v_dept_name    new_depts.department_name%TYPE := 'A New Department';

   v_dept_id        new_depts.department_id%TYPE;

   v_max_deptno new_depts.department_id%TYPE;

BEGIN

   SELECT max(department_id) INTO v_max_deptno

     FROM new_depts;

   SELECT department_name INTO v_dept_name

   FROM new_depts WHERE department_id=v_max_deptno;


   DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);

   DBMS_OUTPUT.PUT_LINE('The department name is: ' || v_dept_name);

END;
```

6. Modify the code to include an INSERT statement to insert a new row into the new_depts table, using v_dept_id and v_dept_name to populate the department_id and department_name columns. Insert NULL into the location_id and manager_id columns. Execute your code and confirm that the new row has been inserted.

```
 DECLARE
  v_dept_name    new_depts.department_name%TYPE := 'A New Department';
  v_dept_id        new_depts.department_id%TYPE;
  v_max_deptno new_depts.department_id%TYPE;
BEGIN
  SELECT max(department_id) INTO v_max_deptno
    FROM new_depts;
  SELECT department_id, department_name INTO v_dept_id, v_dept_name
  FROM new_depts WHERE department_id=v_max_deptno;

  DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);
  DBMS_OUTPUT.PUT_LINE('The department name is: ' || v_dept_name);
  INSERT INTO new_depts VALUES(v_dept_id,v_dept_name,NULL,NULL);
END;
```

7. Now modify the code to use SQL%ROWCOUNT to display the number of rows inserted, and execute the block again.

```
DECLARE
  v_dept_name    new_depts.department_name%TYPE := 'A New Department';
  v_dept_id         new_depts.department_id%TYPE;
  v_max_deptno new_depts.department_id%TYPE;
BEGIN
  SELECT max(department_id) INTO v_max_deptno
    FROM new_depts;
  SELECT department_id, department_name INTO v_dept_id, v_dept_name
  FROM new_depts WHERE department_id=v_max_deptno;

  DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);
  DBMS_OUTPUT.PUT_LINE('The department name is: ' || v_dept_name);
  INSERT INTO new_depts VALUES(v_dept_id,v_dept_name,NULL,NULL);
  DBMS_OUTPUT.PUT_LINE('Rows Inserted ' || SQL%ROWCOUNT);
END;
```

8. Now modify the block, removing the INSERT statement and adding a statement that will UPDATE all rows with location_id = 1700 to location_id = 1400. Execute the block again to see how many rows were updated.

```
 DECLARE
  v_dept_name    new_depts.department_name%TYPE := 'A New Department';
  v_dept_id         new_depts.department_id%TYPE;
  v_max_deptno new_depts.department_id%TYPE;
BEGIN
  SELECT max(department_id) INTO v_max_deptno
    FROM new_depts;
  SELECT department_id, department_name INTO v_dept_id, v_dept_name
  FROM new_depts WHERE department_id=v_max_deptno;

  DBMS_OUTPUT.PUT_LINE('The maximum department id is: ' || v_max_deptno);
  DBMS_OUTPUT.PUT_LINE('The department name is: ' || v_dept_name);
UPDATE new_depts SET location_id=1400 WHERE location_id= 1700;
  DBMS_OUTPUT.PUT_LINE('Rows Inserted ' || SQL%ROWCOUNT);
END;
```