3-3: Manipulating Data in PL/SQL

Agustín Alejandro Mota Hinojosa

October 28, 2023

Contents

1 Vocabulary

1. Defined automatically by Oracle for all SQL data manipulation statements, and for queries that return only one row.

   Cursor implicit

2. Defined by the programmer for queries that return more than one row.

   Cursor explicit

3. 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.

   Merge

4. Statement adds new rows to the table.

   Insert

5. Statement removes rows from the table.

   Delete

6. Statement modifies existing rows in the table.

   Update

2 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 **Cursor Attributes** and are available when you use **implicit** cursors.

3. Examine and run the following PL/SQL code, which obtains and displays the maximum $\text{department}_{\text{id}}$ from $\text{new}_{\text{depts}}$. What is the maximum department id?

```
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_max_deptno new_depts.department_id%TYPE;
    v_dept_name new_depts.department_name%TYPE := 'A New Department';
    v_dept_id 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_{deptid}.

```
DECLARE
    v_max_deptno new_depts.department_id%TYPE;
    v_dept_name new_depts.department_name%TYPE := 'A New Department';
    v_dept_id 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);
    v_dept_id := v_max_deptno + 10;
    DBMS_OUTPUT.PUT_LINE('v_dept_id: '|| v_dept_id);
END;
```

6. Modify the code to include an INSERT statement to insert a new row into the new_{depts} table, using v_{deptid} and $v_{deptname}$ 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_max_deptno new_depts.department_id%TYPE;
    v_dept_name new_depts.department_name%TYPE := 'A New Department';
    v_dept_id 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);
    v_dept_id := v_max_deptno + 10;
    INSERT INTO new_depts(department_id, department_name, manager_id, location_id)
    VALUES(v_dept_id, v_dept_name, NULL, NULL);
    DBMS_OUTPUT.PUT_LINE('v_dept_id: '|| v_dept_id);
END;
```

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

```
DECLARE
    v_max_deptno new_depts.department_id%TYPE;
    v_dept_name new_depts.department_name%TYPE := 'A New Department';
    v_dept_id 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);
    v_dept_id := v_max_deptno + 10;
    INSERT INTO new_depts(department_id, department_name, manager_id, location_id)
    VALUES(v_dept_id, v_dept_name, NULL, NULL);
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT);
END;
```

8. Now modify the block, removing the INSERT statement and adding a statement that will UPDATE all rows with $\text{location}_{\text{id}} = 1700$ to $\text{location}_{\text{id}} = 1400$. Execute the block again to see how many rows were updated.

```
DECLARE
    v_max_deptno new_depts.department_id%TYPE;
    v_dept_name new_depts.department_name%TYPE := 'A New Department';
    v_dept_id 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);
    v_dept_id := v_max_deptno + 10;
    UPDATE new_depts SET location_id = 1400 WHERE location_id = 1700;
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT);
END;
```