

# Manipulating Data in PL/SQL

## Terminology

Directions: Identify the vocabulary word for each definition below:

1. \_\_\_\_\_ Defined automatically by Oracle for all SQL data manipulation statements, and for queries that return only one row.
2. \_\_\_\_\_ Defined by the programmer for queries that return more than one row.
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.
4. \_\_\_\_\_ Statement adds new rows to the table.
5. \_\_\_\_\_ Statement removes rows from the table.
6. \_\_\_\_\_ Statement modifies existing rows in the table.

## 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.
2. \_\_\_\_\_ cursors are created by the programmer.
3. \_\_\_\_\_ cursors are created by the Oracle server.
4. SQL%FOUND, SQL%NOTFOUND, and SQL%ROWCOUNT are \_\_\_\_\_, and are available when you use \_\_\_\_\_ 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;
```

5. Examine and run the following PL/SQL code, which obtains and displays the maximum department\_id from new\_depts.

```
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;
```

6. 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;
```

7. Modify the code to add 10 to the current maximum department number and assign the result to v\_dept\_id.
8. 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. Save your code.
9. Execute the block and check that the new row has been inserted.
10. Now modify the code to use SQL%ROWCOUNT to display the number of rows inserted, and execute the block again.
11. 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.