Database Programming with PL/SQL

3-3: Manipulating Data in PL/SQL

Practice Activities

Cursorul implicit se numeste mereu SQL si este o zona de memorie alocata catre instructiunea sql si datele folosite de aceasta

SQL%(NOT)FOUND – boolean, SQL%ROWNUM – cate linii afectate de cea mai recenta interogare

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. |

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.

False! Ca foloseste de ala implicit

2. SQL%FOUND, SQL%NOTFOUND, and SQL%ROWCOUNT are \_cursor 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?

**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;**

190

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;

gata

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

v\_dept\_id.

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\_max\_deptno:=v\_max\_deptno + 10;

v\_dept\_id := v\_max\_deptno;

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.

NU AM MODIFICAT

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\_max\_deptno:=v\_max\_deptno + 10;

v\_dept\_id := v\_max\_deptno;

END;

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

execute the block again.

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.

Efectiv nu mai am chef