Database Programming with PL/SQL

5-1: Introduction to Explicit Cursors

Practice Activities

Vocabulary

|  |  |
| --- | --- |
| Explicit cursor | Declared by the programmer for queries that return more than  A label for a context area or a pointer to the context areaone row |
| cursor | A label for a context area or a pointer to the context area |
| CLOSE | Disables a cursor, releases the context area, and undefines the  active set |
| Context Area | An allocated memory area used to store the data processed by  a SQL statement |
| Implicit Cursor | Defined automatically by Oracle for all SQL DML statements,  and for SELECT statements that return only one row |
| OPEN | Statement that executes the query associated with the cursor,  identifies the active set, and positions the cursor pointer to the  first row |
| FETCH | Statement that retrieves the current row and advances the  cursor to the next row either until there are no more rows or until  a specified condition is met |
| Active Set | The set of rows returned by a multiple row query in an explicit  cursor operation |

Try it/Solve it

1. In your own words, explain the difference between implicit and explicit cursors.

Explicit cursor poate returna mai multe linii si este definit manual de programator.

Pentru mai multe linii returnate, ai nevoie de un LOOP care sa se opreasca cu conditia numeCursor%NOTFOUND (devine True)

2. Which SQL statement can use either an explicit or an implicit cursor, as needed?

Instructiunea select care returneaza o singura tupla

Update o singura linie

3. List two circumstances in which you would use an explicit cursor.

Ca mai sus, doar ca cu mai multe

4. Exercise using CURRENCIES tables:

A. Write a PL/SQL block to declare a cursor called currencies\_cur. The cursor will be used to

read and display all rows from the CURRENCIES table. You will need to retrieve

currency\_code and currency\_name, ordered by ascending currency\_name.

DECLARE

cursor currencies\_cur is SELECT currency\_code, currency\_name FROM wf\_currencies

ORDER BY currency\_name;

B. Add a statement to open the currencies\_cur cursor.

BEGIN

OPEN currencies\_cur;

C. Add variable declarations and an executable statement to read ONE row through the

currencies\_cur cursor into local variables.

In declare:

v\_currency\_code wf\_currencies.currency\_code%TYPE;

v\_currency\_name wf\_currencies.currency\_name%TYPE;

in begin, dupa open:

FETCH currencies\_cur INTO v\_currency\_code, v\_currency\_name;

D. Add a statement to display the fetched row, and a statement to close the currencies\_cur

cursor.

in begin, dupa open:

FETCH currencies\_cur INTO v\_currency\_code, v\_currency\_name;

DBMS...(v\_currency\_code||’ ‘||v\_currency\_name);

CLOSE currencies\_cur;

END;

E. Run your block to confirm that it works. It should display: AFA Afghani.

F. Your code so far displays only one row. Modify your code so that it fetches and displays all the

rows, using a LOOP and EXIT statement. Test your modified block. It should fetch and display

each row in the CURRENCIES table. If it doesn't, check that your EXIT statement is in the

correct place in the code.

DECLARE

cursor currencies\_cur is SELECT currency\_code, currency\_name FROM wf\_currencies

ORDER BY currency\_name;

v\_currency\_code wf\_currencies.currency\_code%TYPE;

v\_currency\_name wf\_currencies.currency\_name%TYPE;

BEGIN

OPEN currencies\_cur;

LOOP

FETCH currencies\_cur INTO v\_currency\_code, v\_currency\_name;

EXIT WHEN currencies\_cur%NOTFOUND;

DBMS...(v\_currency\_code||’ ‘||v\_currency\_name);

CLOSE currencies\_cur;

END LOOP;

END;

G. Write and test a PL/SQL block to read and display all the rows in the COUNTRIES table for all

countries in region 5 (South America region). For each selected country, display the

country\_name, national\_holiday\_date, and national\_holiday\_name. Display only those

countries having a national holiday date that is not null. Save your code (you will need it in the

next practice).

Declare

cursor countries\_cur is select country\_name, national\_holiday\_date, national\_holiday\_name from wf\_countries where region\_id=5 and national\_holiday\_date IS NOT NULL;

v\_country\_name wf\_countries.country\_name%TYPE;

v\_nat\_name wf\_countries.national\_country\_name%TYPE;

v\_nat\_date wf\_countries.national\_country\_date%TYPE;

BEGIN

open countries\_curl

LOOP

FETCH countries\_cur into v\_country\_name, v\_nat\_name, v\_nat\_date;

EXIT WHEN countries\_cur%NOTFOUND;

dbms(mesaj adecvat)

end loop;

close countries\_cur;

end;

5. Identify three guidelines for declaring and using explicit cursors.

Mereu deschide cursorul inainte de a-l folosi

Nu baga in clauza select termenul INTO

Fetch fiecare linie una cate una

Mereu inchide cursorul dupa utilizare

6. Write a PL/SQL block to read and display the names of world regions, with a count of the number of countries in each region. Include only those regions having at least 10 countries. Order your output by ascending region name.

DECLARE

 v\_region\_id wf\_world\_regions.region\_id%TYPE;

 v\_count NUMBER(2);

 v\_region\_name wf\_world\_regions.region\_name%TYPE;

 CURSOR regions\_cur IS

 SELECT region\_id

 FROM wf\_world\_regions;

BEGIN

 OPEN regions\_cur;

 LOOP

FETCH regions\_cur INTO v\_region\_id;

EXIT WHEN regions\_cur%NOTFOUND;

SELECT count(country\_id) INTO v\_count

FROM wf\_countries

WHERE region\_id = v\_region\_id;

IF v\_count >= 10 THEN

SELECT region\_name INTO v\_region\_name

FROM wf\_world\_regions

WHERE region\_id = v\_region\_id;

DBMS\_OUTPUT.PUT\_LINE(v\_region\_name || ' ' || v\_count);

END IF;

END LOOP;

 CLOSE regions\_cur;

END;