 



Database Programming with PL/SQL 5-1: Introduction to Explicit Cursors Practice Activities

# Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| Cursor explicit | Declared by the programmer for queries that return more than one 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 |
| Cursor implicit | Defined automatically by Oracle for all SQL DML statements, and for SELECT statements that return only one row |
| OPEN statement | Statement that executes the query associated with the cursor, identifies the active set, and positions the cursor pointer to the first row |
| FETCH statement | 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.

Cursorul implicit este declarat automat de SQL si este utilizat cu DML Statements(INSERT, UPDATE, DELETE) si cu clauze SELECT care returneaza o singura linie.

Cursorul explicit este declarat de programator si este utilizat cand clauza SELECT returneaza mai mult de o linie.

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

SELECT

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

Daca dorim sa afisam salariile dintr-o companie care are mai multi angajati sau daca vrem sa afisam numele angajatiilor din companie.

1. Exercise using CURRENCIES tables:
   1. 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\_depts IS

SELECT currency\_code, currency\_name FROM currencies

ORDER BY currency\_name;

v\_currency\_code currencies.currency\_code%TYPE;

v\_currency\_name currencies.currency\_name%TYPE;

BEGIN

OPEN currencies\_depts;

LOOP

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

EXIT WHEN currencies\_depts%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_currency\_code||' '||v\_currency\_name);

END LOOP;

CLOSE currencies\_depts;

END;

* 1. Add a statement to open the currencies\_cur cursor.

OPEN currencies\_cur;

* 1. Add variable declarations and an executable statement to read ONE row through the currencies\_cur cursor into local variables.

v\_curr\_code currencies.currency\_code%TYPE;

v\_curr\_name currencies.currency\_name%TYPE:

* 1. Add a statement to display the fetched row, and a statement to close the currencies\_cur cursor.

FETCH currencies\_cur INTO v\_curr\_code, v\_curr\_name; DBMS\_OUTPUT.PUT\_LINE( v\_curr\_code || ' ' || v\_curr\_name ); CLOSE currencies\_cur;

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

DECLARE CURSOR currencies\_cur IS SELECT currency\_code, currency\_name FROM currencies; v\_curr\_code currencies.currency\_code%TYPE;

v\_curr\_name currencies.currency\_name%TYPE;

BEGIN

OPEN currencies\_cur; FETCH currencies\_cur INTO v\_curr\_code, v\_curr\_name;

DBMS\_OUTPUT.PUT\_LINE( v\_curr\_code || ' ' || v\_curr\_name ); CLOSE currencies\_cur;

END;

* 1. 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 currencies;

v\_curr\_code currencies.currency\_code%TYPE;

v\_curr\_name currencies.currency\_name%TYPE;

BEGIN

OPEN currencies\_cur;

LOOP

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

DBMS\_OUTPUT.PUT\_LINE( v\_curr\_code || ' ' || v\_curr\_name );

EXIT WHEN currencies\_cur%NOTFOUND;

END LOOP;

CLOSE currencies\_cur;

END;

* 1. 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\_name,

national\_holiday\_date FROM countries

WHERE region\_id = 5 AND national\_holiday\_name IS NOT NULL;

v\_country\_name countries.country\_name%TYPE; 4

v\_hol\_name countries.national\_holiday\_name%TYPE;

v\_hol\_date countries.national\_holiday\_date%TYPE;

BEGIN

OPEN countries\_cur;

LOOP

FETCH countries\_cur INTO v\_country\_name, v\_hol\_name, v\_hol\_date;

DBMS\_OUTPUT.PUT\_LINE( v\_country\_name || ' ' || v\_hol\_name || ' ' || v\_hol\_date );

EXIT WHEN countries\_cur%NOTFOUND;

END LOOP;

CLOSE countries\_cur;

END;

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

-cursorul trebuie declarat

-cursorul trebuie deschis

-cursorul trebuie inchis

1. 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

CURSOR region\_cur IS SELECT distinct location, count(location)

FROM countries

HAVING count(\*) > 10

group by location;

v\_reg\_name countries.location%TYPE;

v\_count PLS\_INTEGER;

BEGIN

OPEN region\_cur;

LOOP

FETCH region\_cur INTO v\_reg\_name, v\_count;

DBMS\_OUTPUT.PUT\_LINE( ' Nombre:' || ' ' || v\_reg\_name || ' ' ||

'Num de paises: ' || ' ' || v\_count );

EXIT WHEN region\_cur%NOTFOUND;

END LOOP;

CLOSE region\_cur;

END;