Database Programming with PL/SQL

6-1: Introduction to Explicit Cursors Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

EXPLICIT CURSOR	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
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.

Un cursor implícito se crea por defecto, en cambio un cursor explícito lo tienes que declarar tu mismo.

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

Una sentencia SELECT.

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

Cuando una consulta devuelve más de una fila.

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 currencies ORDER BY currency_name asc;
v_currency_code currencies.currency_code%type;
v_currency_name 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_OUTPUT.PUT_LINE(v_currency_code|| ' '|| v_currency_name);
END LOOP;
```

```
CLOSE currencies_cur;
END;
```

B. Add a statement to open the currencies_cur cursor.

```
DECLARE
cursor currencies_cur IS SELECT currency_code, currency_name
FROM currencies ORDER BY currency_name asc;
BEGIN
 open currencies_cur;

END;
```

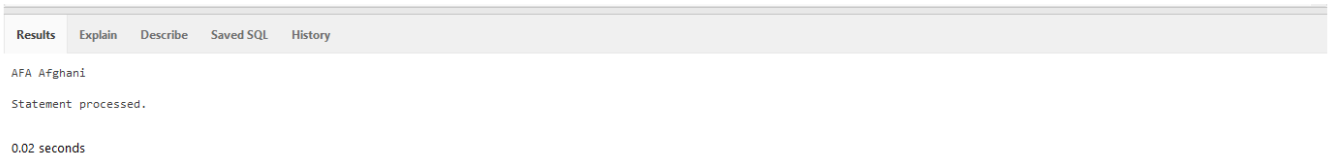
C. Add variable declarations and an executable statement to read ONE row through the currencies_cur cursor into local variables.

```
DECLARE
cursor currencies_cur IS SELECT currency_code, currency_name
FROM currencies ORDER BY currency_name asc;
v_currency_code currencies.currency_code%type;
v_currency_name currencies.currency_name%type;
BEGIN
open currencies_cur;
FETCH currencies_cur INTO v_currency_code, v_currency_name;
CLOSE currencies_cur;
END;
```

D. Add a statement to display the fetched row, and a statement to close the currencies_cur cursor.

```
DECLARE
cursor currencies_cur IS SELECT currency_code, currency_name
FROM currencies ORDER BY currency_name asc;
v_currency_code currencies.currency_code%type;
v_currency_name currencies.currency_name%type;
BEGIN
open currencies_cur;
FETCH currencies_cur INTO v_currency_code, v_currency_name;
DBMS_OUTPUT.PUT_LINE(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 currencies ORDER BY currency_name asc;
v_currency_code currencies.currency_code%type;
v_currency_name 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_OUTPUT.PUT_LINE(v_currency_code|| ' '|| v_currency_name);

END LOOP;
CLOSE currencies_cur;

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 cur_countries IS SELECT country_name, NATIONAL_HOLIDAY_DATE,
NATIONAL_HOLIDAY_NAME

FROM countries WHERE NATIONAL_HOLIDAY_DATE IS NOT NULL AND REGION_ID =5;

v_country_name countries.country_name%type;

v_NATIONAL_HOLIDAY_DATE countries.NATIONAL_HOLIDAY_DATE%type;

v_NATIONAL_HOLIDAY_NAME countries.NATIONAL_HOLIDAY_NAME%type;


BEGIN

open cur_countries;

LOOP

FETCH cur_countries INTO v_country_name , v_NATIONAL_HOLIDAY_DATE,
v_NATIONAL_HOLIDAY_NAME ;

EXIT WHEN cur_countries%NOTFOUND;

DBMS_OUTPUT.PUT_LINE(v_country_name || ' '|| v_NATIONAL_HOLIDAY_DATE || ' '||
v_NATIONAL_HOLIDAY_NAME );

END LOOP;

CLOSE cur_countries;

END;
```

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

Nombre del cursor

La consulta que va almacenar el cursor

Abrir el cursor

Cerrar el cursor

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
CURSOR cur_regions IS SELECT REGION_NAME, count(*) AS countries
FROM regions r, countries c
 WHERE r.region_id = c.region_id
GROUP By r.region_name
HAVING COUNT(*)>10
ORDER BY r.region_name asc;
v_region_name regions.region_name%type;
v_count NUMBER(3);
BEGIN
OPEN cur_regions;
loop
FETCH CUR_REGIONS INTO v_region_name, v_count;
DBMS_OUTPUT.PUT_LINE('Region '||v_region_name ||' Numero de paises '||v_count);
EXIT WHEN cur_regions%NOTFOUND;
end loop;
close cur_regions;
END;
```