5-1: Introduction to Explicit Cursors

Vocabulary

1.Explicit cursor

2.Cursor

3.Close

4.Context Area

5.Implicit cursor

6.OPEN

7.Fetch

8.Active set

Try It / Solve It

1.

implicit:Defined automatically by Oracle for all SQL DML statements, and for SELECT statements that return only one row

explicit:Declared by the programmer for queries that return more than one row

2.

Select

3.

when using a statement that returns more than 1 row

gives more control over the processing of the rows

4.

A.

```
DECLARE

 CURSOR currencies_cur IS SELECT currency_code,currency_name FROM
wf_Currencies ORDER BY currency_name;

 curr_code wf_currencies.currency_code%TYPE;

 curr_name wf_currencies.currency_name%TYPE;

BEGIN

OPEN  currencies_cur;

 FETCH  currencies_cur  INTO curr_code,curr_name;

  DBMS_OUTPUT.PUT_LINE(curr_code||' '|| curr_name);

 CLOSE  currencies_cur;

END;
```

F.

```
DECLARE

 CURSOR currencies_cur IS SELECT currency_code,currency_name FROM
wf_Currencies ORDER BY currency_name;

 curr_code wf_currencies.currency_code%TYPE;

 curr_name wf_currencies.currency_name%TYPE;

BEGIN

OPEN  currencies_cur;

 LOOP

  FETCH  currencies_cur  INTO curr_code,curr_name;

  EXIT WHEN  currencies_cur%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE(curr_code||' '|| curr_name);

 END LOOP;

 CLOSE  currencies_cur;

END;
```

G.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;

 co_name wf_countries.country_name%TYPE;

 co_n_holiday wf_countries.national_holiday_date%TYPE;

 co_h_name wf_countries.national_holiday_name%TYPE;

BEGIN

 OPEN  countries_cur;

 LOOP

  FETCH  countries_cur  INTO  co_name,co_n_holiday, co_h_name ;

  EXIT WHEN  countries_cur%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE(co_name||' '|| co_n_holiday||' '||co_h_name);

 END LOOP;

 CLOSE  countries_cur;

END;
```

5.

- do not include the INTO clause in the cursor declaration
- use ORDER BY for processing rows in an order
- the cursor can be any valid SELECT statement

6.

```
DECLARE
 CURSOR region_cursor IS SELECT region_id, COUNT(*) AS how_many
FROM wf_countries
 GROUP BY region_id
 HAVING COUNT(*) > 10;
v_reg wf_countries.region_id%TYPE;
 nr PLS_INTEGER;
BEGIN OPEN region_cursor;
 LOOP
 FETCH region_cursor INTO v_reg, nr;
 DBMS_OUTPUT.PUT_LINE(v_reg||' -> '||nr);

 EXIT WHEN region_cursor%NOTFOUND; END LOOP;
 CLOSE region_cursor;
end;
```