

CHAPTER-13

INTERACTING WITH DATABASES

Cursor is a method that provides a way to select multiple rows of data from the database and then process each row individually inside the PL/SQL program.

OR

PL/SQL requires special capability to retrieve and process more than one rows of data from the table and this task is performed by Cursor.

Types of Cursors :

- **Implicit Cursor** : Declared for all DML & PL/SQL SELECT statement including queries that return only one row.
- **Explicit Cursor**: For queries that return more than one row. Declared and named by programmer.

Explicit Cursor :

Following steps are followed when using the explicit cursor:

1. Declare the cursor.
2. Open the cursor.
3. Fetch the data from the cursor record by record.
4. Close the cursor.

Syntax for using cursor:

- Syntax of cursor declaration:
cursor <cursorname> is SQL select statement;
- Syntax of open statement:
open <cursorname>;
- Syntax of fetch statement:
fetch <cursorname> into var1,var2,var3.....
or
fetch <cursorname> into record_type_variable;
- Syntax of close statement:
close <cursorname>;

Explicit Cursor attributes:

There are four cursor attributes :

- **%ISOPEN** : It returns true if cursor is open , otherwise it returns false.
Syntax : if cursorname%isopen then

 else

 endif
- **%FOUND** : It returns true if record was fetched properly from the cursor, otherwise it returns false.
Syntax : cursorname%found
- **%NOTFOUND** : It returns true if record is not successfully found, otherwise it returns false.
Syntax : cursorname%notfound
- **%ROWCOUNT** : It returns the no. of records processed by the cursor.
Syntax : cursorname%rowcount

Example code to illustrate cursor:

```
DECLARE
    cursor empdis is SELECT * FROM emp;
    e emp%ROWTYPE;
BEGIN
    open empdis;
    if empdis%ISOPEN then
        dbms_output.put_line('empno  name  salary');
    LOOP
        FETCH empdis into e;
        EXIT WHEN empdis%notfound;
        dbms_output.put_line(e.empno|| e.name|| e.sal);
    ENDLOOP;
    CLOSE empdis;
    else
        dbms_output.put_line('Cannot open the cursor');
    END IF;
END;
```

Implicit Cursors:

These are also called as SQL cursors.

PL/SQL employs implicit cursors for following statements:

- i. INSERT
- ii. UPDTAE
- iii. DELETE
- iv. SELECT(only those SELECT queries that return exactly one row.)

Subqueries,IN WHERE clause,IN FROM clause, Aliases, Expressions,bind variables can be used with Explicit cursors.

Cursor Based Records:**Cursor FOR Loops:**

In a Cursor FOR Loop,a declared cursor is OPENed,FETCHed and CLOSeD automatically.

Syntax:

```
FOR <record_index> IN <cursor_name>  
LOOP  
  <body of loop>  
END LOOP;
```

Cursor FOR Loop with Parameters:**Syntax:**

```
FOR <record_name> IN <cursor_name>(<parameter_list_here>)  
LOOP  
  .  
  .  
  .  
END LOOP;
```

Cursor FOR Loops Using Subqueries:**Syntax:**

```
FOR <record_name> IN (query_expression)  
LOOP  
  .  
  .  
  .  
END LOOP;
```

* * *