Bhushan Asodi

Roll No :- 05.

Sub :- DBMS Lab

Date

Page

Practical No :- 07

* Title :- Cursors (All type, Implict, Explicit Cursor for Loop, parameterized Cursor)

* Objective :- To learn understand and execute process of Software Application Development.

* Theory :-

Cursor - When a query returns multiple rows you can explicity define a cursor to.

1) process beyond the first row returned by the query.

2) keep track of which row is currently processed.

i) Declaring Cursor :- Declare a cursor by giving it a name and associating if with a query as the following

Syntax :- Cursor . Cursorname IS
select statement without into
clause

ii) Opening a Cursor - use the open statement to execute the query and identify the active set.

Syntax :- open Cursor_name;

Scanned with CamScanner

iii) fetching from a cursor - Fetch statement to retrives rows from the active set and specify the output host variable that will contain results.

Syntax :-

FETCH cursor-name INTO variable names / record name

iv) Closing a cursor :- When finished fetching rows from the active set you can close the cursor

Syntax :- cursor cursor-name;

* Cursor Attribute

i) %ISOPEN
ii) %NOTFOUND
iii) %FOUND
iv) %ROWCOUNT -

* Types of Cursors :-

① Implicit Cursor - Implicit cursor are also known as default cursor as SQL server. These cursor are allocated by SQL server. when the user performs DML operation.

② Explicit cursor : - Explicit cursors are created by users whenever the user required them. Explicit cursors are used for fetching data from tables in Row-by Row manners.

Scanned with CamScanner

* Cursor For loop :- The cursor for loop statement is an elegant extension of the numeric for loop statement. The numeric for loop executes the body of a loop once for every integer value in specificed range similary the cursor for loop once for each row returned by the query associated with the cursor

Syntax :—

```
For record IN Cursor-name
  Loop
    process_record statement;
  End loop;
```

* Parameterized cursor - Parameterized cursor are static cursor that can accept passed in parameter values when they are opened

Syntax :—

```
CREATE OR REPLACE PROCEDURE
myproc (in_id IN NUMBER) IS
cursor (curserid in NUMBER)
IS
select statement From table_name
Condition;
BEGIN
  For i IN
  Statement
  DBMS_OUTPUT.PUTLINE;
  END Loop;
END;
```

Scanned with CamScanner

Conclusion :- Hence we studied about the implementation of curses.

Scanned with CamScanner