

### **Experiment2.3**

**Name:** Rohan Jaiswal

**UID:** 21BCS2856

**Branch:** CSE

**Section/Group:** 608 (B)

**Semester:** 3<sup>rd</sup>

**Date of Performance:** 27<sup>th</sup> Oct

**Subject Name:** DBMS

**Subject Code:** 21CSH-214

#### **Aim:**

Introduction and implementation of programs using Cursors.

#### **Objective:**

Implementing Implicit, Explicit and Parameterized Cursors.

#### **Theory:**

- **Cursor** – Predefined area in oracle engine for internal processing.
  - **Types**
    - Implicit – Automatically created.
    - Explicit – User Defined
  - **Attributes**
    - %isopen – return true if cursor is opened else false.
    - %found – return true if data is fetched from main table else false.
    - %notfound – Opposite of “%found”.
    - %rowcount – return the count of rows effected/fetched.

## Explicit Cursors

Explicit cursors are programmer defined cursors for gaining more control over the context area. An explicit cursor should be defined in the declaration section of the PL/SQL Block.

## Syntax

Working with an explicit cursor involves four steps:

- Declaring the cursor for initializing in the memory
- Opening the cursor for allocating memory
- Fetching the cursor for retrieving data
- Closing the cursor to release allocated memory

## Declaring the Cursor

Declaring the cursor defines the cursor with a name and the associated SELECT statement. For example:

```
CURSOR c_customers IS SELECT id, name, address FROM customers;
```

## Opening the Cursor

Opening the cursor allocates memory for the cursor, and makes it ready for fetching the rows returned by the SQL statement into it. For example :

```
OPEN c_customers;
```

## Fetching the Cursor

Fetching the cursor involves accessing one row at a time. For example:

```
FETCH c_customers INTO c_id, c_name, c_addr;
```

## Closing the Cursor

Closing the cursor means releasing the allocated memory. For example, we will close above opened cursor as follows:

```
CLOSE c_customers;
```

## DBMS script and Output:

- Creating table and inserting rows

```
CREATE TABLE Studinfo(  
    Sid number(10),  
    Sname varchar2(20),  
    sub1 number(5),  
    sub2 number(5),  
    sub3 number(5),  
    Total number(6)  
);  
  
INSERT into Studinfo values  
(1,'Abhishek',50,54,49,153);  
INSERT into Studinfo values  
(2,'Rohan',20,34,29,83);  
INSERT into Studinfo values  
(3,'Himanshu',13,12,25,50);  
INSERT into Studinfo values  
(4,'Aditya Ruhela',1,5,7,13);  
INSERT into Studinfo values  
(5,'Aditya Barnwal',32,45,49,126);  
INSERT into Studinfo values  
(6,'Sameer',0,1,2,3);  
INSERT into Studinfo values  
(7,'Harsh',12,13,45,70);  
INSERT into Studinfo values  
(8,'Aarush',32,45,49,126);  
INSERT into Studinfo values  
(9,'Vivek',5,2,36,43);  
INSERT into Studinfo values  
(10,'Mike',2,90,23,115);
```

- Implicit Cursor (deleting 1 row)

```
BEGIN  
    DELETE from Studinfo Where Sid=10;  
    if sql%found then  
        dbms_output.put_line('Row Deleted');  
    else  
        dbms_output.put_line('Row not Deleted');  
    END IF;  
END;
```

- Implicit Cursor (Deleting multiple rows)

```
DECLARE
    n number;
BEGIN
    DELETE from Studinfo Where Total<50;
    if sql%found then
        n:= sql%rowcount;
        dbms_output.put_line(n || ' rows Deleted');
    else
        dbms_output.put_line('No data found');
    END IF;
END;
```

- Parameterized Cursor

```
DECLARE
    tmp number;
    total_marks number:= 0;
    cursor c1(n number) is select total from Studinfo where total<n;
BEGIN
    open c1(100);
    LOOP
        fetch c1 into tmp;
        EXIT WHEN c1%notfound;

        total_marks:= total_marks + tmp;
    END LOOP;
    dbms_output.put_line('Total marks of Students having marks less than 100 is ' || total_marks);
    close c1;
END;
```

- Explicit Cursor

```
DECLARE
    cursor c1 is select Sid, Sname, total from Studinfo Where Total<100;
    rec c1%rowtype;
BEGIN
    open c1;
    LOOP
        fetch c1 into rec;
        EXIT WHEN c1%notfound;

        dbms_output.put_line('Sid: ' || rec.Sid);
        dbms_output.put_line('Sname: ' || rec.Sname);
        dbms_output.put_line('Total: ' || rec.Total);
    END LOOP;
    close c1;
END;
```