**Ex. No. 10 PL/SQL – CURSOR AND ITS IMPLEMENTATION**

**AIM**

To create and implement PL/SQL cursor

**DESCRIPTION:**

A **cursor** is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**.

You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors −

* Implicit cursors
* Explicit cursors
* **Implicit Cursor**

Whenever any DML operations occur in the database, an implicit cursor is created that holds the rows affected, in that particular operation. These cursors cannot be named and, hence they cannot be controlled or referred from another place of the code. We can refer only to the most recent cursor through the cursor attributes.

|  |  |
| --- | --- |
| **S.No** | **Attribute & Description** |
| 1 | **%FOUND**  Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE. |
| 2 | **%NOTFOUND**  The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE. |
| 3 | **%ISOPEN**  Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement. |
| 4 | **%ROWCOUNT**  Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement. |

## Explicit Cursor

Programmers are allowed to create named context area to execute their DML operations to get more control over it. The explicit cursor should be defined in the declaration section of the PL/SQL block, and it is created for the 'SELECT' statement that needs to be used in the code.

Working with an explicit cursor includes the following steps −

* Declaring the cursor for initializing the memory
* Opening the cursor for allocating the memory
* Fetching the cursor for retrieving the data
* Closing the cursor to release the 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

OPEN c\_customers;

## Fetching the Cursor

Fetching the cursor involves accessing one row at a time.

FETCH c\_customers INTO c\_id, c\_name, c\_addr;

## Closing the Cursor

Closing the cursor means releasing the allocated memory.

CLOSE c\_customers;

EXPLICIT CURSOR

SQL> declare

2 avg\_1 number(10);

3 sal emp\_5.salary%type;

4 cursor c4 is select \* from emp\_5;

5 rw c4%rowtype;

6 begin

7 select avg(salary) into avg\_1 from emp\_5;

8 dbms\_output.put\_line('the average is'||avg\_1);

9 open c4;

10 fetch c4 into rw;

11 for i in (select \* from emp\_5)

12 loop

13 if (rw.salary<=avg\_1) then

14 sal:=rw.salary+1000;

15 update emp\_5 set salary=sal

16 where empno=rw.empno;

17 end if ;

18 end loop;

19 end;

20 /

OUTPUT

the average is43000

PL/SQL procedure successfully completed.

PROGRAM

SQL> declare

2 t student\_1.total%type;

3 p student\_1.percentage%type;

4 cursor str is select \* from student\_1;

5 rw str%rowtype;

6 begin

7 open str;

8 loop

9 fetch str into rw;

10 exit when str%notfound;

11 t:=rw.m1+rw.m2+rw.m3;

12 p:=(t/300)\*100;

13 update student\_1 set total=t,

14 percentage=p where regno=rw.regno;

15 end loop;

16 end;

17 /

PL/SQL procedure successfully completed.

SQL> select \* from student\_1;

REGNO SNAME M1 M2 M3 TOTAL PERCENTAGE

---------- ---------- ---------- ---------- ---------- ---------- ----------

1 vijay 99 95 96 290 97

2 gokul 98 99 90 287 96

3 hari 80 87 85 252 84