EX: NO: 10 PL / SQL Program to implement Cursors

Description :

The oracle engine uses a work area for its internal processing in order to execute an SQL statement. This work area is private to SQL's operations and is called a cursor. A cursor is a mechanism by which you can assign a name to a "select statement" and manipulate the information within that SQL statement.

There are **two** types of Cursors.

- Implicit Cursors
- Explicit Cursors

Both the types of Cursors have following General attributes.

Attribute Name	Description
%ISOPEN	returns true if cursor is open, false otherwise
%FOUND	returns true if record was fetched successfully, false otherwise.
%NOTFOUND	returns true if record was not fetched successfully, false otherwise.
%ROWCOUNT	returns number of records processed from the cursor.

Implicit cursor attributes can be used to access information about status of last insert, update, delete or single - row select statement. This can be done by preceding the implicit cursor

attribute with cursor name (i.e. sql)

Explicit cursor is defined in the declarative part of a pl/sql block. This is done by naming the cursor and mapping it to a query. The commands used to control the cursor are **DECLARE, OPEN, FETCH and CLOSE.**

- **Oracle/PLSQL: Declare a Cursor**

A cursor is a SELECT statement that is defined within the declaration section of your PLSQL code. We'll take a look at three different syntaxes for cursors.

1. **Cursor without parameters (simplest)**

   The basic syntax for a cursor without parameters is:

   CURSOR cursor_name IS SELECT_statement;

2. **Cursor with parameters**

   The basic syntax for a cursor with parameters is:

   CURSOR cursor_name (parameter_list) IS SELECT_statement;

3. **Cursor with return clause**

The basic syntax for a cursor with a return clause is:

CURSOR cursor_name RETURN field%ROWTYPE

IS SELECT_statement;

• Oracle/PLSQL: OPEN Statement

Once you've declared your cursor, the next step is to open the cursor.

The basic syntax to OPEN the cursor is:

OPEN cursor_name;

• Oracle/PLSQL: FETCH Statement

The purpose of using a cursor, in most cases, is to retrieve the rows from your cursor so that some type of operation can be performed on the data. After declaring and opening your cursor, the next step is to FETCH the rows from your cursor.

The basic syntax for a FETCH statement is:

FETCH cursor_name INTO <list of variables>;

• Oracle/PLSQL: CLOSE Statement

The final step of working with cursors is to close the cursor once you have finished using it. The basic syntax to CLOSE the cursor is:

CLOSE cursor_name;

EX: NO: 10 (a) Write a PL/SQL code to retrieve the employee name, join_date, and designation from employee database of an employee whose number is input by the user.

SOLUTION:
Use table in Experiment-5

Pl/SQL Code :

```
declare
enum emp.empno%type;
name emp.ename%type;
begin
 enum:=&enum;
select empno,ename into enum,name from emp where
empno=enum;
dbms_output.put_line('---------output----------');
dbms_output.put_line('employee no :'||enum);
dbms_output.put_line('employee name :'||name);
end;
/
Enter value for eno: 501
---------output----------
```

employee no :501
employee name : jagadeesh
PL/SQL procedure successfully completed.

Exp 10 (b):

Write a PL/SQL program using explicit cursors to find and display the maximum salary employee details

Code:

```
declare
        cursor c_emp is select ename,salary from emp;
        name emp.ename%type;
        sal emp.salary%type;
        maxsal emp.salary%type:=0;
        maxsalname emp.ename%type;
        begin
                open c_emp;

                loop
                        fetch c_emp into name,sal;
                        if(maxsal<sal) then
                                maxsal:=sal;
                                maxsalname:=name;
                        end if;
                exit when c_emp%notfound;
                end loop;
        dbms_output.put_line('DETAILS OF THE PERSON GETTING MAXIMUM SALARY');
        dbms_output.put_line('EMPLOYEE NAME : '||maxsalname);
        dbms_output.put_line('SALARY : '||maxsal);
        close c_emp;
        end;
/
```

Triggers:

A trigger is a statement that the system executes automatically as a side effect of a modification to the database. To design a trigger mechanism, we must meet two requirements:

1. Specify when a trigger is to be executed. This is broken up into an event (INSERT, UPDATE or DELETE) that causes the trigger to be checked and a condition that must be satisfied for trigger execution to proceed.
2. Specify the actions to be taken when the trigger executes.

```
create trigger Trigger_name
(before | after)
[insert | update | delete]
on [table_name]
[for each row]
[trigger_body]
```

Experiment No. 10(c):

Write a program on triggers to make sure that when employee name is inserted or updated it must be in upper case

CODE:

```
create or replace trigger emp_insert_update before
insert or update on emp for each row
declare
begin
:new.ename:=upper(:new.ename);
end;
/
```