**********Experiment no:-07**********

Author:Saurabh Khandagale
Roll No:46
Date :22-Oct-2020

EXPERIMENT:-07

AIM:-

To write and execute SQL programs for retrieving data using a cursor and to demonstrate various cursors.

Problem Statement:

Using the relation schemata established in Experiments - 02, 03, and 05, create and execute SQL programs for retrieving data using cursors.

==========Query-01==========

Write a SQL code to compile and execute an anonymous block which declares a cursor - FACULTY. The cursor buffers the records comprising - Employee ID, Employee Name (FNAME and LNAME combined) and Designation for the Designation entered by the user.
You may use either EMPLOYEE table or EMPP table for this cursor and print the buffered records. Use %NOTFOUND variable to enable cursor exit.

==========

```
DECLARE
ENO1 EMPP.EID%TYPE;
ENAME1 EMPP.ENAME%TYPE;
DESIGNATION1 EMPP.DESIGNATION%TYPE;


ENO2 EMPP.EID%TYPE;
ENAME2 EMPP.ENAME%TYPE;
DESIGNATION2 EMPP.DESIGNATION%TYPE;

CURSOR FACULTY(D EMPP.DESIGNATION%TYPE) IS SELECT  DISTINCT(ENO), FNAME||'
'||LNAME AS ENAME,DESIGNATION FROM EMPLOYEE WHERE DESIGNATION=INITCAP(D) ;

BEGIN
DBMS_OUTPUT.PUT_LINE('ENTER DESIGNATION ');
DESIGNATION1:='&DESIGNATION1';
OPEN FACULTY(DESIGNATION1);
DBMS_OUTPUT.PUT_LINE(-----------------------------)
LOOP
FETCH FACULTY INTO ENO1,ENAME1,DESIGNATION1;

DBMS_OUTPUT.PUT_LINE(ENO1||'  '||  ENAME2||'  '||DESIGNATION3);
EXIT WHEN FACULTY%NOTFOUND;
END LOOP;
CLOSE FACULTY;
END;


/
--Professor
```

```
Enter value for designation1: Professor
old  11: DESIGNATION1:='&DESIGNATION1';
new  11: DESIGNATION1:='Professor';
ENTER DESIGNATION
7102  Samantha Jones  Professor
7101  Eugene Sabatini  Professor
7103  Alexander Lloyd  Professor
7104  Simon Downing  Professor
7119  Atulya Bharat  Professor

PL/SQL procedure successfully completed.
```

==========Query02==========

Modify the cursor in Query-01 as FACULTY_CFL which uses the cursor FOR loop to buffering and displaying the records (as mentioned) when employee designation is entered by the user.
Use a variation of cursor FOR loop to include the ROWCOUNT variable to print serial number for the displayed records.

==========

```
DECLARE
ENO1 EMPP.EID%TYPE;

ENAME1 EMPP.ENAME%TYPE;
DESIGNATION1 EMPP.DESIGNATION%TYPE;

CURSOR FACULTY_CFL(D EMPP.DESIGNATION%TYPE) IS SELECT  EID, ENAME ,DESIGNATION
FROM EMPP WHERE DESIGNATION=INITCAP(D)  ;

BEGIN

DESIGNATION1:='&DESIGNATION';
FOR I IN  FACULTY_CFL (DESIGNATION1)
LOOP

DBMS_OUTPUT.PUT_LINE(I.EID||'  '||  I.ENAME||'  '||I.DESIGNATION);
END LOOP;
END;
/
[Output]
Enter value for designation: Professor
old  11: DESIGNATION1:='&DESIGNATION';
new  11: DESIGNATION1:='Professor';
7102  Samantha Jones  Professor
7101  Eugene Sabatini  Professor
7103  Alexander Lloyd  Professor
7104  Simon Downing  Professor
7119  Atulya Bharat  Professor
```

```
DECLARE
ENO1 EMPP.EID%TYPE;

ENAME1 EMPP.ENAME%TYPE;
DESIGNATION1 EMPP.DESIGNATION%TYPE;

CURSOR FACULTY_CFL(D EMPP.DESIGNATION%TYPE) IS SELECT  EID, ENAME ,DESIGNATION
FROM EMPP WHERE DESIGNATION=INITCAP(D)  ;
ROWCOUNT NUMBER(2):=0;
BEGIN

DESIGNATION1:='&DESIGNATION';
FOR I IN  FACULTY_CFL (DESIGNATION1)
LOOP
ROWCOUNT:=ROWCOUNT+1;
DBMS_OUTPUT.PUT_LINE(ROWCOUNT||' '||I.EID||'  '||  I.ENAME||'
'||I.DESIGNATION);
END LOOP;
END;
/
Enter value for designation: Professor
old  11: DESIGNATION1:='&DESIGNATION';
new  11: DESIGNATION1:='Professor';
1 7102  Samantha Jones  Professor
2 7101  Eugene Sabatini  Professor
3 7103  Alexander Lloyd  Professor
4 7104  Simon Downing  Professor
5 7119  Atulya Bharat  Professor

PL/SQL procedure successfully completed.


[Note:-I have use InitCap].


=====================================Query03============================
Write Sql code to list the first and last name of employees who were born
before 01-jan-1972
And who are posted in western region.
========================================================================
DECLARE
ENO1 EMPP.EID%TYPE;

ENAME1 EMPP.ENAME%TYPE;
DESIGNATION1 EMPP.DESIGNATION%TYPE;

CURSOR FACULTY_CFL(D EMPP.DESIGNATION%TYPE) IS SELECT  EID, ENAME ,DESIGNATION
FROM EMPP WHERE DESIGNATION=INITCAP(D)  ;
ROWCOUNT NUMBER(2):=0;
NO NUMBER(2);
BEGIN
DESIGNATION1:='&DESIGNATION';
NO:='&NO';
FOR I IN  FACULTY_CFL (DESIGNATION1)
LOOP
ROWCOUNT:=ROWCOUNT+1;
```

```
DBMS_OUTPUT.PUT_LINE(ROWCOUNT||' '||I.EID||'  '||  I.ENAME||'
'||I.DESIGNATION);
EXIT WHEN  ROWCOUNT=NO;

END LOOP;
END;
/
Enter value for designation: Professor
old  11: DESIGNATION1:='&DESIGNATION';
new  11: DESIGNATION1:='Professor';
Enter value for no: 2
old  12: NO:='&NO';
new  12: NO:='2';
1 7102  Samantha Jones  Professor
2 7101  Eugene Sabatini  Professor
```

==========Query04==========

Write a SQL code to compile and execute an anonymous block which declares a cursor - EMP_SAL_INFO (Salary, Designation). Let the default values for salart and designation be 75000 and "Asst. Professor" respectively.
The cursor buffers the records comprising - Employee ID, Employee Name (FNAME and LNAME combined), Designation and Salary for the Salary and Designation entered by the user.Use EMPLOYEE table for this cursor. Use this cursor to print the buffered records.

==========

```
SQL> DECLARE
 CURSOR EMP_SAL_INFO(SAL EMPLOYEE.SALARY%TYPE :=86400 ,
  DESG EMPLOYEE.DESIGNATION%TYPE := 'Asst. Professor')
  IS
  SELECT ENO,FNAME||' '||LNAME AS NAME,DESIGNATION,SALARY
  FROM EMPLOYEE WHERE
  DESIGNATION LIKE '%'||DESG AND SALARY=SAL;
  RECORDS EMP_SAL_INFO%ROWTYPE;
  BEGIN
  DBMS_OUTPUT.PUT_LINE(chr(10));
  OPEN EMP_SAL_INFO;
  DBMS_OUTPUT.PUT_LINE('With default values ... ');
  DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
  DBMS_OUTPUT.PUT_LINE(RPAD('EID',4,' ')||' '
||RPAD('EMPLOYEE NAME',20,' ')||' '||RPAD('DESIGNATION',15,' ')||' '
||RPAD('SALARY',10,' '));
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 LOOP
 FETCH EMP_SAL_INFO INTO RECORDS;
 EXIT WHEN EMP_SAL_INFO%NOTFOUND;
 DBMS_OUTPUT.PUT_LINE(RECORDS.ENO||'
'||RPAD(RECORDS.NAME,20,' ')||
 ' '||RPAD(RECORDS.DESIGNATION,15,' ')||'
'||RECORDS.SALARY);
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 CLOSE EMP_SAL_INFO;

 DBMS_OUTPUT.PUT_LINE(chr(10));
```

```
 OPEN EMP_SAL_INFO(&specified_salary);
 DBMS_OUTPUT.PUT_LINE('With some default values ... ');
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 DBMS_OUTPUT.PUT_LINE(RPAD('EID',4,' ')||' '
||RPAD('EMPLOYEE NAME',20,' ')||' '||RPAD('DESIGNATION',15,' ')||' '
||RPAD('SALARY',10,' '));
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 LOOP
 FETCH EMP_SAL_INFO INTO RECORDS;
 EXIT WHEN EMP_SAL_INFO%NOTFOUND;
 DBMS_OUTPUT.PUT_LINE(RECORDS.ENO||'
'||RPAD(RECORDS.NAME,20,' ')||
 ' '||RPAD(RECORDS.DESIGNATION,15,' ')||'
'||RECORDS.SALARY);
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 CLOSE EMP_SAL_INFO;

 DBMS_OUTPUT.PUT_LINE(chr(10));
 OPEN
EMP_SAL_INFO(&specified_salary,&specified_designation);
 DBMS_OUTPUT.PUT_LINE('With all supplied values ..... ');
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 DBMS_OUTPUT.PUT_LINE(RPAD('EID',4,' ')||' '
||RPAD('EMPLOYEE NAME',20,' ')||' '||RPAD('DESIGNATION',15,' ')||' '
||RPAD('SALARY',10,' '));
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 LOOP
 FETCH EMP_SAL_INFO INTO RECORDS;
 EXIT WHEN EMP_SAL_INFO%NOTFOUND;
 DBMS_OUTPUT.PUT_LINE(RECORDS.ENO||'
'||RPAD(RECORDS.NAME,20,' ')||' '
 ||RPAD(RECORDS.DESIGNATION,15,' ')||'
'||RECORDS.SALARY);
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('---- ------------------- -----------
---- ----------');
 CLOSE EMP_SAL_INFO;
 END;
 /
```

```
Enter value for specified_salary: 91000
old 27: OPEN EMP_SAL_INFO(&specified_salary);
new 27: OPEN EMP_SAL_INFO(91000);
Enter value for specified_salary: 127400
Enter value for specified_designation: 'Asso. Professor'
old 42: OPEN
EMP_SAL_INFO(&specified_salary,&specified_designation);
new 42: OPEN EMP_SAL_INFO(127400,'Asso. Professor');
With default values ...
---- ------------------- -------------- ----------
EID EMPLOYEE NAME DESIGNATION SALARY
---- ------------------- -------------- ----------
7110 William Smithfield Asst. Professor 86400
---- ------------------- -------------- ----------
With some default values ...
---- ------------------- -------------- ----------
EID EMPLOYEE NAME DESIGNATION SALARY
---- ------------------- -------------- ----------
7109 Martina Jacobson Asst. Professor 91000
---- ------------------- -------------- ----------
With all supplied values .....
---- ------------------- -------------- ----------
EID EMPLOYEE NAME DESIGNATION SALARY
---- ------------------- -------------- ----------
7107 Christov Plutnik Asso. Professor 127400
7105 Christina Mulboro Asso. Professor 127400
7106 Dolly Silverline Asso. Professor 127400
---- ------------------- -------------- ----------
PL/SQL procedure successfully completed.

====================================Query05================================
      Write SQL code to compile and execute a procedure - PRINT_EMPLOYEE
      which receives employee salary as input and prints the following
      particulars - employee number, employee name and salary, for
      employees whose salary exceeds the inputted salary.
      You must use a cursor - SAL_CURSOR, to buffer required result-
      set for bulk collect. Use TYPE statement to declare and instantiate
      array variables.
      You may also try using %ROWCOUNT. Use EMPP table as source. You may
      also use EMPLOYEE table.
===========================================================================
SQL> CREATE OR REPLACE PROCEDURE PRINT_EMPLOYEE(SAL NUMBER)
  AS
  TYPE NUM_ARRAY IS VARRAY(10000) OF NUMBER;
  TYPE STR_ARRAY IS VARRAY(10000) OF VARCHAR2(25);
  TYPE SAL_ARRAY IS VARRAY(10000) OF NUMBER;

  ENO_ARR NUM_ARRAY;
  ENAME_ARR STR_ARRAY;
  SAL_ARR SAL_ARRAY;
  CURSOR SAL_CURSOR IS
  SELECT EID,ENAME,SALARY FROM EMPP
  WHERE SALARY>SAL;
  BEGIN
  OPEN SAL_CURSOR;
  FETCH SAL_CURSOR
  BULK COLLECT INTO ENO_ARR,ENAME_ARR,SAL_ARR;
  CLOSE SAL_CURSOR;
```

```
  DBMS_OUTPUT.PUT_LINE('---- ------------------- ----------');
 DBMS_OUTPUT.PUT_LINE(RPAD('EID',4,' ')||' ' ||RPAD('EMPLOYEE
NAME',20,' ')||' '||RPAD('SALARY',10,' '));
 DBMS_OUTPUT.PUT_LINE('---- ------------------- ----------');
 FOR KNT IN ENO_ARR.FIRST .. ENO_ARR.LAST LOOP
 DBMS_OUTPUT.PUT_LINE(RPAD(ENO_ARR(KNT),4,' ')||'
'||RPAD(ENAME_ARR(KNT),20,' ')||' '||RPAD(SAL_ARR(KNT),10,' '));
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('---- ------------------- ----------');
 END;
 /
Procedure created.
SQL> CALL PRINT_EMPLOYEE(20000);
---- ------------------- ----------
EID EMPLOYEE NAME SALARY
---- ------------------- ----------
7102 Samantha Jones 146500
7103 Alexander Lloyd 148000
7104 Simon Downing 138400
7105 Christina Mulboro 127400
7106 Dolly Silverline 127400
7107 Christov Plutnik 127400
7108 Ellena Sanchez 119700
7109 Martina Jacobson 91000
7110 William Smithfield 86400
7111 Albert Greenfield 48200
7112 James Washington 44600
7113 Julia Martin 35600
7114 Larry Gomes 32850
7115 Svetlana Sanders 30000
7116 Lovelyn Brendon 30000
7117 Hector Hercules 32200
7118 Prachi Bhanuse 25000
---- ------------------- ----------
Call completed.

********************************************************************************
Viva Voice
==========

**********Question-1**********
      What is a cursor? List the steps associated with implementing a
      cursor.

Ans:-

      Ans:
      In computer science, a database cursor is a control structure that
      enables traversal over the records in a database. Cursors facilitate
      subsequent processing in conjunction with the traversal, such as
      retrieval, addition and removal of database records.

      There are four steps in using an Explicit Cursor.
      1. DECLARE the cursor in the Declaration section.
      2. OPEN the cursor in the Execution Section.
      3. FETCH the data from the cursor into PL/SQL variables or records in
      the
         Execution Section.
```

4. CLOSE the cursor in the Execution Section before you end the PL/SQL Block.

**********Question-2**********

What is an "active set"?

Ans:
Active-set methods are defined for a particular primal and dual formulation of a QP with general equality constraints and simple lower bounds on the variables. In the first part of the paper, two methods are proposed, one primal and one dual. These methods generate a sequence of iterates that are feasible with respect to the equality constraints associated with the optimality conditions of the primal–dual form. The primal method maintains feasibility of the primal inequalities while driving the infeasibilities of the dual inequalities to zero. The dual method maintains feasibility of the dual inequalities while moving to satisfy the primal inequalities.

**********Question-3**********

What is a cursor FOR loop? Why it is advantageous?

Ans:
The cursor FOR LOOP statement implicitly declares its loop index as a record variable of the row type that a specified cursor returns, and then opens a cursor. With each iteration, the cursor FOR LOOP statement fetches a row from the result set into the record.

ADVANTAGES OF CURSORS USING FOR LOOP
1.No need to open the cursor.
2.Fetch the records automatically.
3.It automatically checks the end of rows.
4.It automatically closes the cursor.
5.No need to declare the variables.
6.code size will be decreased.
7.execution will be faster.
8.less fetching time.
9.It is collection of information from cursor to a variable.

**********Question-4**********

Why it is a good practice to close a cursor?

Ans:
When you declare a cursor in a package (that is, not inside a subprogram of the package) and the cursor is opened, it will stay open until you explicitly close it or your session is terminated.
When the cursor is declared in a declaration section (and not in a package), Oracle Database will also automatically close it when the block in which it is declared terminates. It is still, however, a good idea to explicitly close the cursor yourself. If the cursor is moved to a package, you will have the now necessary CLOSE already in place. And if it is local, then including a CLOSE statement will also show other developers and your manager that you are paying attention.
****

Inference:-
==========

1. LEARN TO USE CURSOR AND BIND VARIAVLE.
2. Cursor is use to fetch multiple values.