

=====

EXPERIMENT NO. 07

=====

Author : Diksha Gupta.
Roll no.: 07 [27A]
Date : 02-DECEMBER-2022.

=====

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

***** QUERY-01 *****

Write a SQL code to compile and execute an anonymous block which declares a 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.

SET SERVEROUTPUT ON;

DECLARE

V_DESIGNATION EMPP.DESIGNATION%TYPE := &DESIGNATION;
V_COUNT NUMBER(2):=0;
CURSOR FACULTY IS
SELECT EID , ENAME , DESIGNATION
FROM EMPP
WHERE UPPER(DESIGNATION) = UPPER(V_DESIGNATION);
EMPP_REC FACULTY%ROWTYPE;

BEGIN

OPEN FACULTY;
LOOP
FETCH FACULTY INTO EMPP_REC;
EXIT WHEN FACULTY%NOTFOUND ;
DBMS_OUTPUT.PUT_LINE(RPAD(EMPP_REC.EID , 8) ||' '|| RPAD(EMPP_REC.ENAME ,
20)||' '||RPAD(EMPP_REC.DESIGNATION,23));

```

        V_COUNT := V_COUNT + 1;
    END LOOP;
    IF V_COUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('NO MATCHING DATA FOUND');
    ELSE
        DBMS_OUTPUT.PUT_LINE('MATCHING DATA FOUND');
    END IF;
    CLOSE FACULTY;
    DBMS_OUTPUT.PUT_LINE('CURSOR PROCESSED');
END ;
/

```

OUTPUT :

Enter value for designation: 'HELLO'

NO MATCHING DATA FOUND

CURSOR PROCESSED

PL/SQL procedure successfully completed.

Enter value for designation: 'PrOfeSsOr'

7119 Atulya Bharat Professor

7101 Eugene Sabatini Professor

7102 Samantha Jones Professor

7103 Alexander Lloyd Professor

7104 Simon Downing Professor

MATCHING DATA FOUND

CURSOR PROCESSED

PL/SQL procedure successfully completed.

***** QUERY-02 *****

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.

SET SERVEROUTPUT ON;

DECLARE

V_DESIGNATION EMP.DESIGNATION%TYPE := &DESIGNATION;

CURSOR FACULTY_CFL IS

SELECT EID , ENAME , DESIGNATION

```

        FROM EMPP
        WHERE UPPER(DSIGNATION) = UPPER(V_DESIGNATION);
BEGIN
    DBMS_OUTPUT.PUT_LINE('THE CURSOR FOR LOOP.....');
    DBMS_OUTPUT.PUT_LINE(CHR(10));
    FOR EMPP_REC IN FACULTY_CFL
    LOOP
        DBMS_OUTPUT.PUT_LINE(RPAD(EMPP_REC.EID , 8) ||' '|| RPAD(EMPP_REC.ENAME ,
                                20)||' '||RPAD(EMPP_REC.DESIGNATION,23));
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(CHR(10));
    DBMS_OUTPUT.PUT_LINE('THE CURSOR FOR LOOP WITH %ROWCOUNT.....');
    DBMS_OUTPUT.PUT_LINE(CHR(10));

    FOR EMPP_REC IN FACULTY_CFL
    LOOP
        DBMS_OUTPUT.PUT_LINE(RPAD(FACULTY_CFL%ROWCOUNT ,4) ||' '||RPAD(EMPP_REC.EID , 8)
                                ||' '|| RPAD(EMPP_REC.ENAME , 20)||' '||RPAD(EMPP_REC.DESIGNATION,23));
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('CURSOR PROCESSED');
END ;
/

```

OUTPUT :

Enter value for designation: 'PROFESSOR'

THE CURSOR FOR LOOP.....

7119	Atulya Bharat	Professor
7101	Eugene Sabatini	Professor
7102	Samantha Jones	Professor
7103	Alexander Lloyd	Professor
7104	Simon Downing	Professor

THE CURSOR FOR LOOP WITH %ROWCOUNT.....

1	7119	Atulya Bharat	Professor
2	7101	Eugene Sabatini	Professor
3	7102	Samantha Jones	Professor
4	7103	Alexander Lloyd	Professor
5	7104	Simon Downing	Professor

CURSOR PROCESSED

PL/SQL procedure successfully completed.

***** QUERY-03 *****

EXITING A CURSOR AFTER FETCHING SPECIFIED NUMBER OF ROWS: Modify the cursor FACULTY_CFL_A to display only those many records as desired by the user. Use %ROWCOUNT to enable the cursor to ensure this.

```

SET SERVEROUTPUT ON;
DECLARE
    V_DESIGNATION EMPP.DESIGNATION%TYPE := &DESIGNATION;
    V_COUNT NUMBER(2):=&HOW_MANY_ROWS;
    CURSOR FACULTY_CFL_A IS
        SELECT EID , ENAME , DESIGNATION
        FROM EMPP
        WHERE UPPER(DESIGNATION) = UPPER(V_DESIGNATION);
BEGIN
    FOR EMPP_REC IN FACULTY_CFL_A
    LOOP
        DBMS_OUTPUT.PUT_LINE(RPAD(FACULTY_CFL_A%ROWCOUNT ,4)
        ||' '||RPAD(EMPP_REC.EID,8)
        ||' '|| RPAD(EMPP_REC.ENAME , 20)||' '||RPAD(EMPP_REC.DESIGNATION,23));

        IF(FACULTY_CFL_A%ROWCOUNT = V_COUNT) THEN
            EXIT;
        END IF;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('CURSOR PROCESSED');
END ;
/

```

OUTPUT :

```

Enter value for designation: 'PROFESSOR'
Enter value for how_many_rows: 4
1  7119  Atulya Bharat      Professor
2  7101  Eugene Sabatini     Professor
3  7102  Samantha Jones     Professor
4  7103  Alexander Lloyd    Professor
CURSOR PROCESSED

```

PL/SQL procedure successfully completed.

Enter value for designation: 'PrOfEsSOr'

Enter value for how_many_rows: 5

1	7119	Atulya Bharat	Professor
2	7101	Eugene Sabatini	Professor
3	7102	Samantha Jones	Professor
4	7103	Alexander Lloyd	Professor
5	7104	Simon Downing	Professor

CURSOR PROCESSED

PL/SQL procedure successfully completed.

***** QUERY-04 *****

Write a SQL code to compile and execute an anonymous block which declares a EMP_SAL_INFO (Salary, Designation). Let the default values for salary and designation be 75000 and "Asst. Professor" respectively. cursor – 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.

SET SERVEROUTPUT ON;

DECLARE

V_EMPLOYEE EMPLOYEE%ROWTYPE;

V_SAL_1 EMPLOYEE.SALARY%TYPE := &SPECIFIED_SALARY;

V_SAL_2 EMPLOYEE.SALARY%TYPE := &SPECIFIED_SALARY;

V_DESIGNATION_2 EMPLOYEE.DESIGNATION%TYPE := &DESIGNATION_SALARY;

CURSOR EMP_SAL_INFO(V_SAL EMPLOYEE.SALARY%TYPE := 75000 , V_DESIGNATION
EMPLOYEE.DESIGNATION%TYPE := 'Asst. Professor') IS

SELECT * FROM EMPLOYEE

WHERE SALARY > V_SAL AND DESIGNATION = V_DESIGNATION;

BEGIN

DBMS_OUTPUT.PUT_LINE('WITH DEFAULT VALUES....');

DBMS_OUTPUT.PUT_LINE(CHR(10));

OPEN EMP_SAL_INFO;

LOOP

FETCH EMP_SAL_INFO INTO V_EMPLOYEE;

EXIT WHEN EMP_SAL_INFO%NOTFOUND ;

```

        DBMS_OUTPUT.PUT_LINE(RPAD(V_EMPLOYEE.ENO , 8) ||' '||
                                RPAD(V_EMPLOYEE.FNAME||' '|| V_EMPLOYEE.LNAME,
                                20)||' '||RPAD(V_EMPLOYEE.DESIGNATION,17)||' '||
                                RPAD(V_EMPLOYEE.SALARY,23));

    END LOOP;
CLOSE EMP_SAL_INFO;

DBMS_OUTPUT.PUT_LINE(CHR(10));
DBMS_OUTPUT.PUT_LINE('WITH SOME DEFAULT VALUES.....');
DBMS_OUTPUT.PUT_LINE(CHR(10));

OPEN EMP_SAL_INFO(V_SAL_1);
    LOOP
        FETCH EMP_SAL_INFO INTO V_EMPLOYEE;
        EXIT WHEN EMP_SAL_INFO%NOTFOUND ;
        DBMS_OUTPUT.PUT_LINE(RPAD(V_EMPLOYEE.ENO , 8) ||' '||
                                RPAD(V_EMPLOYEE.FNAME||' '|| V_EMPLOYEE.LNAME,
                                20)||' '||RPAD(V_EMPLOYEE.DESIGNATION,17)||' '||
                                RPAD(V_EMPLOYEE.SALARY,23));

    END LOOP;
CLOSE EMP_SAL_INFO;

DBMS_OUTPUT.PUT_LINE('WITH SUPPLIED VALUES.....');
DBMS_OUTPUT.PUT_LINE(CHR(10));

OPEN EMP_SAL_INFO(V_SAL_2,V_DESIGNATION_2);
    LOOP
        FETCH EMP_SAL_INFO INTO V_EMPLOYEE;
        EXIT WHEN EMP_SAL_INFO%NOTFOUND ;
        DBMS_OUTPUT.PUT_LINE(RPAD(V_EMPLOYEE.ENO , 8) ||' '||
                                RPAD(V_EMPLOYEE.FNAME||' '||
                                V_EMPLOYEE.LNAME, 20)||' '||
                                RPAD(V_EMPLOYEE.DESIGNATION,17)
                                ||' '||RPAD(V_EMPLOYEE.SALARY,23));

    END LOOP;
CLOSE EMP_SAL_INFO;

END;
/

```

OUTPUT :

Enter value for specified_salary: 88000

Enter value for specified_salary: 12000

Enter value for designation_salary: 'Asso. Professor'

WITH DEFAULT VALUES.....

7109	Martina Jacobson	Asst. Professor	91000
7110	William Smithfield	Asst. Professor	86400

WITH SOME DEFAULT VALUES.....

7109	Martina Jacobson	Asst. Professor	91000
------	------------------	-----------------	-------

WITH SUPPLIED VALUES.....

7107	Christov Plutnik	Asso. Professor	127400
7105	Christina Mulboro	Asso. Professor	127400
7106	Dolly Silverline	Asso. Professor	127400
7108	Ellena Sanchez	Asso. Professor	119700

PL/SQL procedure successfully completed.

***** QUERY-05 *****

BULK COLLECT with CURSORS:-

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.

CREATE OR REPLACE PROCEDURE PRINT_EMPLOYEE(V_EMP_SALARY_INP EMPP.SALARY%TYPE)

AS

```
TYPE EMP_ENO IS VARRAY(100) OF EMPP.EID%TYPE;
TYPE EMP_ENAME IS VARRAY(100) OF EMPP.ENAME%TYPE;
TYPE EMP_SALARY IS VARRAY(100) OF EMPP.SALARY%TYPE;
V_EMP_ENO EMP_ENO;
V_EMP_ENAME EMP_ENAME;
```

```

V_EMP_SALARY EMP_SALARY;
CURSOR SAL_CURSOR IS
    SELECT EID , ENAME , SALARY
    FROM EMPP
    WHERE SALARY > V_EMP_SALARY_INP;
BEGIN
    OPEN SAL_CURSOR;
    FETCH SAL_CURSOR
        BULK COLLECT INTO V_EMP_ENO , V_EMP_ENAME , V_EMP_SALARY;
    CLOSE SAL_CURSOR;

    DBMS_OUTPUT.PUT_LINE(RPAD('EID',7) ||' '||RPAD('ENAME',20) ||' '||RPAD('SALARY',9));
    DBMS_OUTPUT.PUT_LINE('----      -----      -----');
    FOR KNT IN V_EMP_ENO.FIRST .. V_EMP_ENO.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(RPAD(V_EMP_ENO(KNT),7) ||' '
                                ||RPAD(V_EMP_ENAME(KNT),20)||' '||RPAD(V_EMP_SALARY(KNT),16));

    END LOOP;
    DBMS_OUTPUT.PUT_LINE('----      -----      -----');
    DBMS_OUTPUT.PUT_LINE('.....END BULK FETCH.....');
END;
/

```

OUTPUT :

```
CALL PRINT_EMPLOYEE(50000);
```

EID	ENAME	SALARY
----	-----	-----
7119	Atulya Bharat	162000
7101	Eugene Sabatini	150000
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
----	-----	-----
.....END BULK FETCH.....		

Call completed.


```
CALL PRINT_EMPLOYEE(125000);
```

```
EMPLOYEE SALARY HAVING > 125000
```

EID	ENAME	SALARY
7119	Atulya Bharat	162000
7101	Eugene Sabatini	150000
7102	Samantha Jones	146500
7103	Alexander Lloyd	148000
7104	Simon Downing	138400
7105	Christina Mulboro	127400
7106	Dolly Silverline	127400
7107	Christov Plutnik	127400

.....END BULK FETCH.....

Call completed.

```
CALL PRINT_EMPLOYEE(148000);
```

```
EMPLOYEE SALARY HAVING > 148000
```

EID	ENAME	SALARY
7119	Atulya Bharat	162000
7101	Eugene Sabatini	150000

.....END BULK FETCH.....

Call completed.

INFERENCES OF THE EXPERIMENT

Hence , we have successfully write and execute SQL programs for retrieving data using a cursor and to demonstrate various cursors.