

-----  
Author : Pradyot Patil

Roll No : 53 [6B]

Date : 5-April-2018  
-----

\*\*\*\*\*

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. Use this cursor to print the buffered records. Use %NOFOUND variable to enable cursor exit.

\*\*\*\*\*

DECLARE

```
    EMP_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP_DESIGNATION;  
    CURSOR FACULTY IS SELECT EID,(FNAME||' '||LNAME) AS NAME,DESIGNATION  
        FROM EMPLOYEE WHERE UPPER(DESIGNATION) LIKE '%'||EMP_DES||'%';  
    V_PART FACULTY%ROWTYPE;
```

BEGIN

```
    OPEN FACULTY ;  
    LOOP  
        FETCH FACULTY INTO V_PART ;  
        EXIT WHEN FACULTY%NOTFOUND ;  
        DBMS_OUTPUT.PUT_LINE(V_PART.EID||' '||V_PART.NAME||'  
        '||V_PART.DESIGNATION);  
    END LOOP ;  
    CLOSE FACULTY ;
```

END;

Enter value for emp\_designation: 'LECTURER'

old 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP\_DESIGNATION;

new 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :='LECTURER';

```
7102 Albert Greenfield  Sr. Lecturer  
7103 Julia Martin      Lecturer  
7108 James Washington  Sr. Lecturer  
7109 Larry Gomes       Lecturer  
7110 Svetlana Sanders  Lecturer
```

PL/SQL procedure successfully completed.

\*\*\*\*\*

QUERY-02 CURSOR FOR LOOP: 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

EMP\_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP\_DESIGNATION;

CURSOR FACULTY\_CFL IS SELECT EID,(FNAME||' '||LNAME) AS NAME,DESIGNATION  
FROM EMPLOYEE WHERE UPPER(DESIGNATION) LIKE '%'||EMP\_DES||'%';

V\_PART FACULTY\_CFL%ROWTYPE;

BEGIN

FOR REC IN FACULTY\_CFL LOOP

EXIT WHEN FACULTY\_CFL%NOTFOUND ;

DBMS\_OUTPUT.PUT\_LINE(FACULTY\_CFL%ROWCOUNT||' '||REC.EID||' '||REC.NAME||'  
'||REC.DESIGNATION);

END LOOP ;

END;

Enter value for emp\_designation: 'PROFESSOR'

old 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP\_DESIGNATION;

new 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :='PROFESSOR';

7101 Samantha Jones	Professor
7104 Martina Jacobson	Asst. Professor
7105 Alexander Lloyd	Professor
7106 William Smithfield	Asst. Professor
7107 Eugene Sabatini	Professor

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.

\*\*\*\*\*

DECLARE

```
EMP_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP_DESIGNATION;
CURSOR FACULTY_CFL IS SELECT EID,(FNAME||' '||LNAME) AS NAME,DESIGNATION
    FROM EMPLOYEE WHERE UPPER(DESIGNATION) LIKE '%'||EMP_DES||'%';
V_NUM NUMBER :=&HOW_MANY_ROWS;
```

BEGIN

```
FOR REC IN FACULTY_CFL LOOP
    IF FACULTY_CFL%ROWCOUNT>V_NUM THEN
        EXIT ;
    END IF ;
    EXIT WHEN FACULTY_CFL%NOTFOUND ;
    DBMS_OUTPUT.PUT_LINE(FACULTY_CFL%ROWCOUNT||' '||REC.EID||' '||REC.NAME||'
    '||REC.DESIGNATION);
END LOOP ;
```

END;

Enter value for emp\_designation: 'PROFESSOR'

old 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP\_DESIGNATION;

new 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :='PROFESSOR';

Enter value for how\_many\_rows: 4

old 5: V\_NUM NUMBER :=&HOW\_MANY\_ROWS;

new 5: V\_NUM NUMBER :=4;

1 7101 Samantha Jones Professor

2 7104 Martina Jacobson Asst. Professor

3 7105 Alexander Lloyd Professor

4 7106 William Smithfield Asst. Professor

PL/SQL procedure successfully completed.

Enter value for emp\_designation: 'PROFESSOR'

old 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :=&EMP\_DESIGNATION;

new 2: EMP\_DES EMPLOYEE.DESIGNATION%TYPE :='PROFESSOR';

Enter value for how\_many\_rows: 8

old 5: V\_NUM NUMBER :=&HOW\_MANY\_ROWS;

new 5: V\_NUM NUMBER :=8;

1 7101 Samantha Jones Professor

2 7104 Martina Jacobson Asst. Professor

3 7105 Alexander Lloyd      Professor  
4 7106 William Smithfield Asst. Professor  
5 7107 Eugene Sabatini      Professor

PL/SQL procedure successfully completed.

\*\*\*\*\*

QUERY-04: PARAMETERIZED CURSOR WITH DEFAULT VALUES: Write a SQL code to compile and execute an anonymous block which declares a cursor - EMP\_SAL\_INFO (Salary, Designation). 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.

\*\*\*\*\*

DECLARE

```
CURSOR EMP_SAL_INFO(sal EMPLOYEE.SALARY%TYPE :=13000,desgn
EMPLOYEE.DESIGNATION%TYPE  := 'LECTURER')IS
SELECT  EID,(FNAME||' '||LNAME) AS NAME,DESIGNATION,SALARY FROM EMPLOYEE
WHERE SALARY>=sal AND UPPER(DESIGNATION) LIKE '%'||desgn||'%';
V_PART EMP_SAL_INFO%ROWTYPE;
```

BEGIN

```
dbms_output.put_line('With Default Values: ');
OPEN EMP_SAL_INFO ;
LOOP
    FETCH EMP_SAL_INFO INTO V_PART ;
    EXIT WHEN EMP_SAL_INFO%NOTFOUND ;
    DBMS_OUTPUT.PUT_LINE(V_PART.EID||' '||V_PART.NAME||'
    '||V_PART.SALARY||' '||V_PART.DESIGNATION);
END LOOP ;
CLOSE EMP_SAL_INFO ;
```

```
dbms_output.put_line('With few Default Values: ');
OPEN EMP_SAL_INFO(&m_salary1) ;
LOOP
    FETCH EMP_SAL_INFO INTO V_PART ;
    EXIT WHEN EMP_SAL_INFO%NOTFOUND ;
    DBMS_OUTPUT.PUT_LINE(V_PART.EID||' '||V_PART.NAME||'
    '||V_PART.SALARY||' '||V_PART.DESIGNATION);
END LOOP ;
CLOSE EMP_SAL_INFO ;
```

```
dbms_output.put_line('With specified Values: ');
OPEN EMP_SAL_INFO(&m_salary2,&m_designation) ;
LOOP
    FETCH EMP_SAL_INFO INTO V_PART ;
```

```

        EXIT WHEN EMP_SAL_INFO%NOTFOUND ;
        DBMS_OUTPUT.PUT_LINE(V_PART.EID||' '||V_PART.NAME||'
        '||V_PART.SALARY||' '||V_PART.DESIGNATION);
    END LOOP ;
CLOSE EMP_SAL_INFO ;

```

END ;

Enter value for m\_salary1: **14000**

old 16:                OPEN EMP\_SAL\_INFO(&m\_salary1) ;

new 16:                OPEN EMP\_SAL\_INFO(14000) ;

Enter value for m\_salary2: **15000**

Enter value for m\_designation: '**PROFESSOR**'

old 25:                OPEN EMP\_SAL\_INFO(&m\_salary2,&m\_designation) ;

new 25:                OPEN EMP\_SAL\_INFO(15000,'PROFESSOR') ;

With Default Values:

7102 Albert Greenfield 14200 Sr. Lecturer

7103 Julia Martin 13320 Lecturer

7108 James Washington 14000 Sr. Lecturer

7109 Larry Gomes 13650 Lecturer

With few Default Values:

7102 Albert Greenfield 14200 Sr. Lecturer

7108 James Washington 14000 Sr. Lecturer

With specified Values:

7101 Samantha Jones 16500 Professor

7104 Martina Jacobson 15550 Asst. Professor

7105 Alexander Lloyd 17500 Professor

7106 William Smithfield 15660 Asst. Professor

7107 Eugene Sabatini 16500 Professor

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 EMP table as source. You may also use EMPLOYEE table

\*\*\*\*\*

DECLARE

V\_SALARY NUMBER :=&M\_SALARY ;

PROCEDURE PRINT\_EMPLOYEE(x IN number) IS

TYPE SAL\_CURSOR

IS TABLE OF EMPLOYEE%ROWTYPE

INDEX BY PLS\_INTEGER;

l\_employees SAL\_CURSOR;

BEGIN

dbms\_output.put\_line('EMPLOYEES HAVING SALARY > '||x);

SELECT \*

BULK COLLECT INTO l\_employees

FROM EMPLOYEE WHERE SALARY>x;

FOR indx IN 1 .. l\_employees.COUNT

LOOP

dbms\_output.put\_line(indx||' '||l\_employees(indx).EID||' '||

l\_employees(indx).FNAME||' '||l\_employees(indx).LNAME||' '||

l\_employees(indx).SALARY);

END LOOP;

END;

BEGIN

PRINT\_EMPLOYEE(V\_SALARY);

dbms\_output.put\_line('END OF BULK FETCH ');

END;

Enter value for m\_salary: 15000

old 2: V\_SALARY NUMBER :=&M\_SALARY ;

new 2: V\_SALARY NUMBER :=15000 ;

EMPLOYEES HAVING SALARY > 15000

```
1 7101 Samantha Jones 16500
2 7104 Martina Jacobson 15550
3 7105 Alexander Lloyd 17500
4 7106 William Smithfield 15660
5 7107 Eugene Sabatini 16500
END OF BULK FETCH
PL/SQL procedure successfully completed.
```

```
Enter value for m_salary: 16000
old 2:      V_SALARY NUMBER :=&M_SALARY ;
new 2:      V_SALARY NUMBER :=16000 ;
EMPLOYEES HAVING SALARY > 16000
1 7101 Samantha Jones 16500
2 7105 Alexander Lloyd 17500
3 7107 Eugene Sabatini 16500
END OF BULK FETCH
```

```
PL/SQL procedure successfully completed.
```