**ASSIGNMENT 4**

**AIM:-**

Write a PL/SQL code to implement all types of cursor (Implicit,Explicit) and display

employee number, name and salary of 5 highest paid employees using

cursor.Employee(employee no, employee name, join\_date, designation, salary).

**OBJECTIVE:-**

To implement Implicit and Explicit Cursor.

**THEORY:-**

To process an SQL statement, ORACLE needs to create an area of memory known as the context area; this will have the information needed to process the statement.

This information includes the number of rows processed by the statement, a pointer to the parsed representation of the statement.

In a query, the active set refers to the rows that will be returned.

A cursor is a handle, or pointer, to the context area.

Through the cursor, a PL/SQL program can control the context area and what happens to it as the statement is processed.

Two important features about the cursor are

Cursors allow you to fetch and process rows returned by a SELECT statement, one row at a time.

A cursor is named so that it can be referenced.

**TYPES OF CURSOR**

There are two types of cursors:

**IMPLICIT CURSOR**

An **IMPLICIT** cursor is automatically declared by Oracle every time an SQL statement is executed. The user will not be aware of this happening and will not be able to control or process the information in an implicit cursor.

**EXPLICIT CURSOR**

An **EXPLICIT** cursor is defined by the program for any query that returns more than one row of data. That means the programmer has declared the cursor within the PL/SQL code block.

Explicit cursors are programmer-defined cursors for gaining more control over the context area.

An explicit cursor should be defined in the declaration section of the PL/SQL Block.

It is created on a SELECT Statement which returns more than one row.

The **syntax** for creating an explicit cursor is −

CURSOR cursor\_name IS select\_statement;

**STEPS:-**

**DECLARING the cursor**. This initializes the cursor into memory.

**OPENING the cursor.** The previously declared cursor can now be opened; memory is allotted.

**FETCHING the cursor**. The previously declared and opened cursor can now retrieve data; this is the process of fetching the cursor.

**CLOSING the cursor**. The previously declared, opened, and fetched cursor must now be closed to release memory allocation.

**PROGRAM:-**

**CREATE TABLE**

CREATE TABLE EMPLOYEE\_(employee\_no int(5),employee\_name varchar(20),join\_date int(10),designation varchar(10),salary int(10));

Insert Into EMPLOYEE\_ VALUES(1,'RAJ',11/11/1111,'PUNE',50000);

SELECT \* FROM EMPLOYEE\_;

**IMPLICIT CURSOR:**

Select employee\_no,employee\_name,salary

From EMPLOYEE\_

order by salary desc limit 5;

**EXPLICIT CURSOR**

delimiter /

CREATE PROCEDURE sp()

begin

DECLARE x int default 0;

DECLARE y varchar(20);

DECLARE z int(10) default 0;

DECLARE finished int default 0;

declare r

cursor for

Select employee\_no,employee\_name,salary

From EMPLOYEE\_

order by salary desc limit 5;

DECLARE CONTINUE HANDLER

FOR NOT FOUND SET finished=1;

open r;

lp:loop

FETCH r INTO x,y,z;

IF finished =1 THEN LEAVE lp;

end if;

select x,y,z;

END LOOP lp;

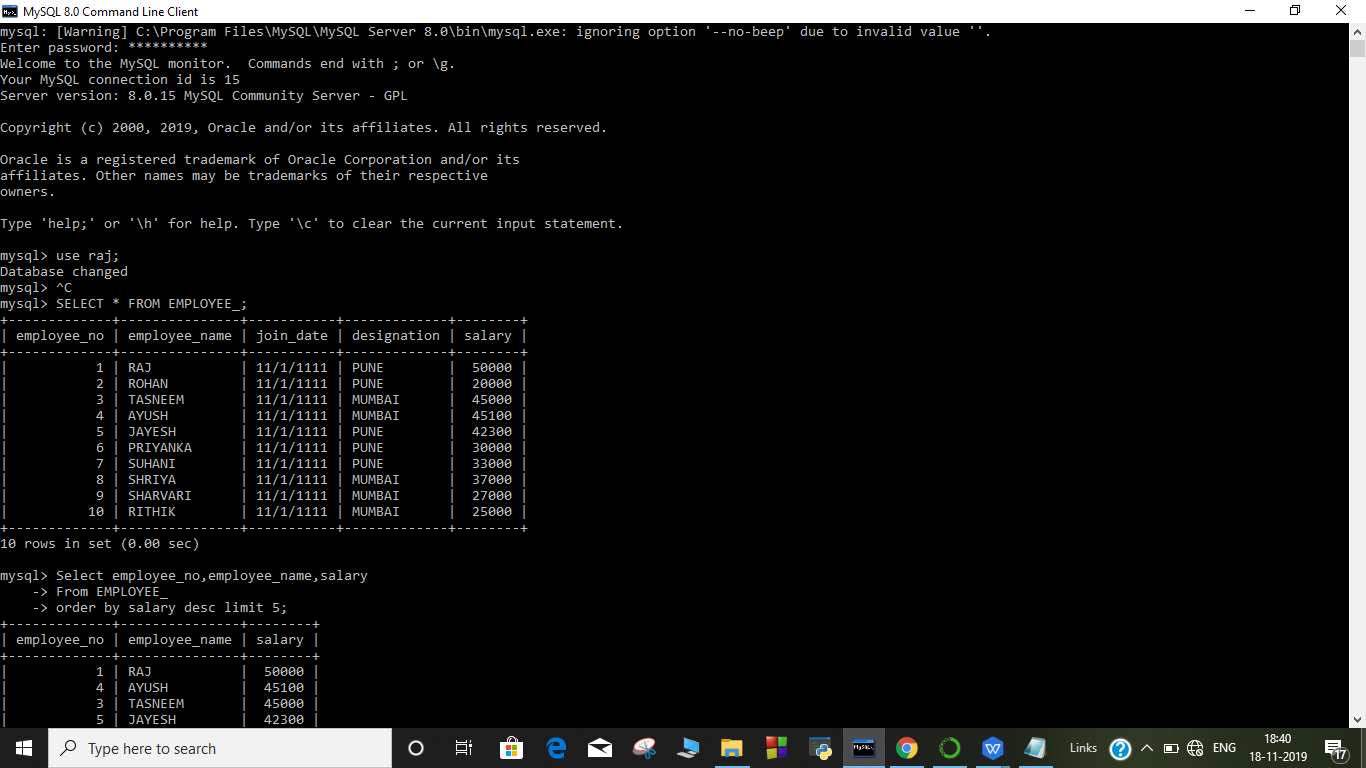
CLOSE r;

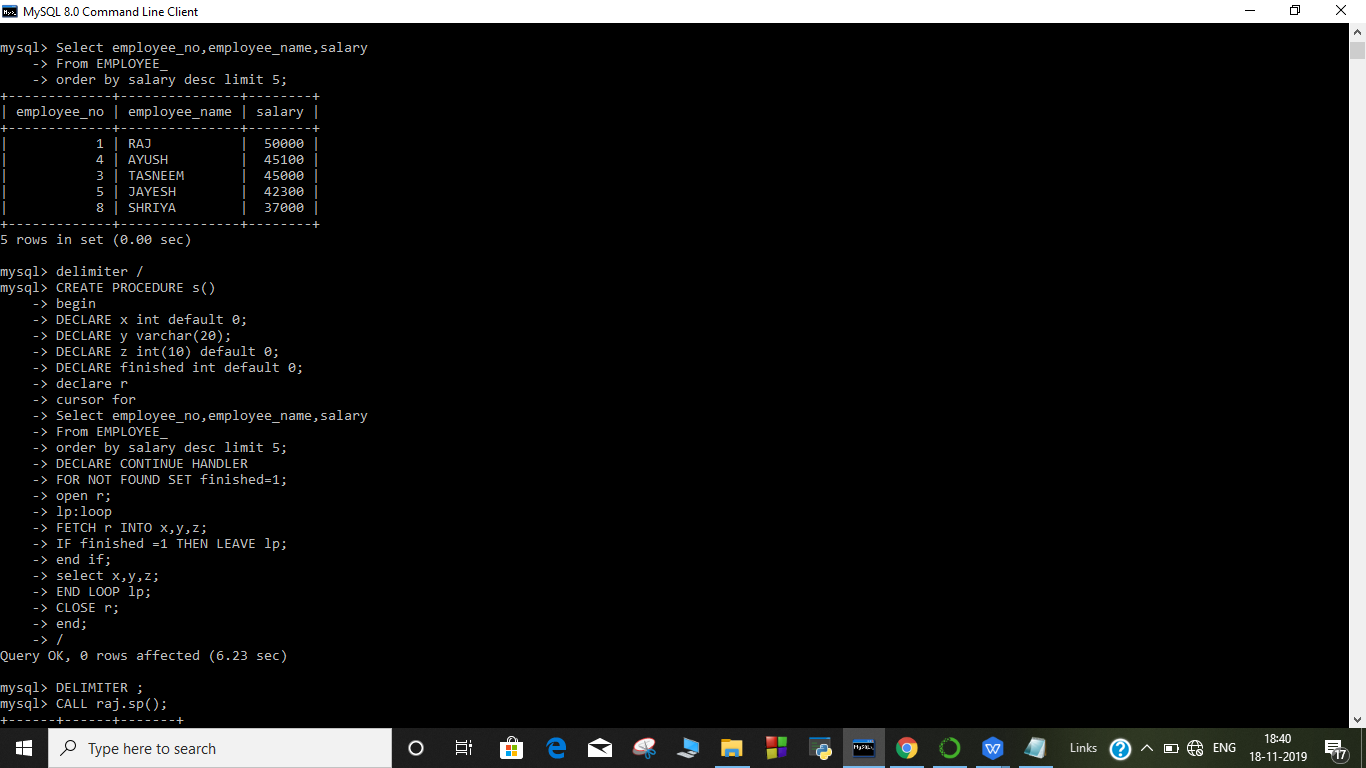
end;

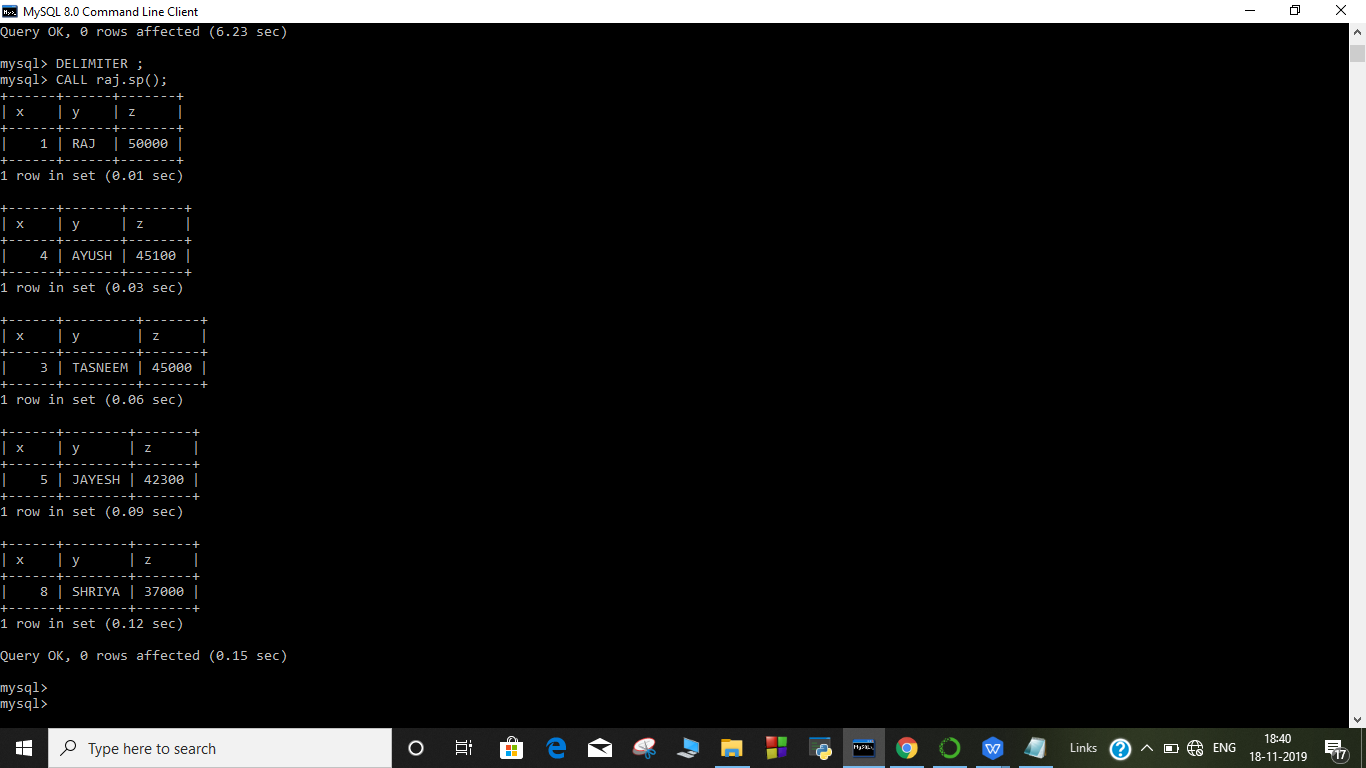
/

DELIMITER ;

CALL raj.sp();

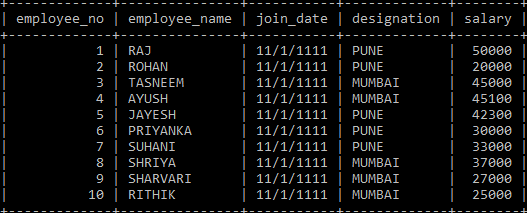




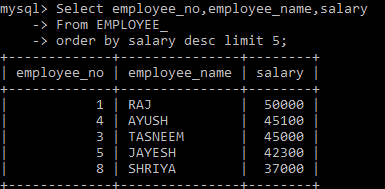


**OUTPUT:-**

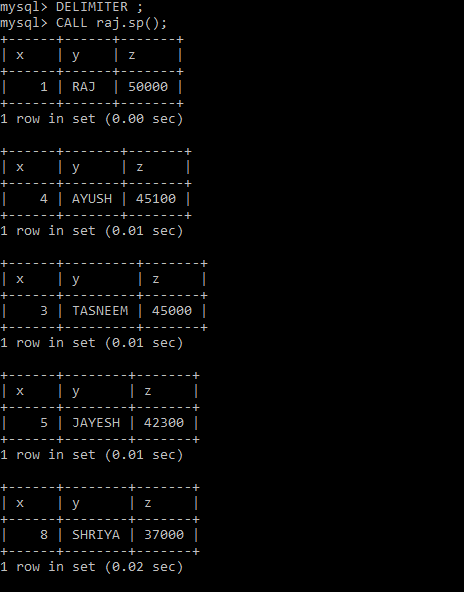
**Table EMPLOYEE\_**

****

**IMPLICIT CURSOR**

****

**EXPLICIT CURSOR:**

****