

With respect to table given in the last assignment, write a program in PL/SQL which stores first\_name, last\_name, salary into cursor and it display first\_name, last\_name, salary and Grade of Employee position of employee depends on Salary. If Salary is greater than 80000 then display it as manager position, of salary is between 50000 to 80000 then

position is Associated and less than 50000 Position is Executive.

### Expected Output

first_name	last_name	salary	position
Rahul	Kumar	87000	Manager
Pravin	Nalwade	66000	Associated
Preeti	Reddy	46000	Executive

DELIMITER \$\$

CREATE PROCEDURE DisplayEmployeeDetails1()

BEGIN

-- Declare variables to store employee data fetched from the cursor

DECLARE v\_first\_name VARCHAR(50);

DECLARE v\_last\_name VARCHAR(50);

DECLARE v\_salary FLOAT;

DECLARE v\_position VARCHAR(20);

DECLARE done INT DEFAULT 0;

-- Declare a cursor to select relevant employee details

DECLARE employee\_cursor CURSOR FOR

SELECT first\_name, last\_name, salary FROM employee;

-- Declare a handler to exit the loop when no more rows are found

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

```

-- Open the cursor
OPEN employee_cursor;

-- Loop through the cursor and fetch each row
employee_loop: LOOP
    -- Fetch the row into variables
    FETCH employee_cursor INTO v_first_name, v_last_name, v_salary;

    -- Exit the loop when no more rows are available
    IF done = 1 THEN
        LEAVE employee_loop;
    END IF;

    -- Assign position based on salary using CASE statement
    SET v_position = CASE
        WHEN v_salary > 80000 THEN 'Manager'
        WHEN v_salary BETWEEN 50000 AND 80000 THEN
'Associated'
        ELSE 'Executive'
    END;

    -- Display employee details (you can customize this for other outputs)
    SELECT v_first_name AS first_name, v_last_name AS last_name,
v_salary AS salary, v_position AS position;

END LOOP;

-- Close the cursor
CLOSE employee_cursor;

END;

CALL DisplayEmployeeDetails1();

```

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, and Scripting. The left sidebar displays the 'SCHEMAS' tree with 'employees' selected. The main editor contains a SQL script that assigns a position based on salary and then displays employee details. The 'Results' tab shows a table with columns 'first\_name', 'last\_name', 'salary', and 'position', containing one row for 'John' with a salary of 8000 and position 'Assistant'. The 'Query History' tab at the bottom shows three executed queries, all of which are 'CALL DisplayEmployeeDetails()'. A tooltip on the right side of the screen reads: 'Automatic context help is disabled. Use the F1 key to manually get help for the current caret position or to toggle automatic help.'