**Queries And Output:**

1. Write a procedure to get details of all employee.

DELIMITER &&

CREATE PROCEDURE Emp\_Details()

BEGIN

SELECT empt\_id, Fname, Lname, dob, salary, Branch\_ID FROM Employee;

END &&

DELIMITER;

CALL Emp\_Details();

Output:



1. Write a procedure which accepts empID as parameter and display its details.

DELIMITER &&

CREATE PROCEDURE Emp\_Details\_ID(IN empID varchar(4))

BEGIN

SELECT empt\_id, Fname, Lname, dob, salary, Branch\_ID FROM Employee;

END &&

DELIMITER;

CALL Emp\_Details\_ID();

Output:



1. Write a procedure which accepts client number as parameter and display the controlling branchID and no. of employeeworking with that client.

DELIMITER &&

CREATE PROCEDURE Client\_Details(In clientNumber varchar(10))

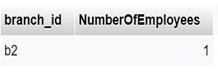
BEGIN

SELECT c.branch\_id, COUNT(ww.emp\_id) AS Numberoffmployees FROM client c LEFT JOIN works\_with ww ON c.client\_id = ww.client\_id WHERE c.client\_id clienthumber;

END &&

DELIMITER;

CALL Client\_Details('c2');



1. Write a procedure which accepts empID as an input and finds the salary and no. of client he is working .
   1. If salary > 49000 and no. of clients he is working on is > 2 increases the salary by 5% .
   2. If salary is between 49000 and 61000 and no. of clients > 2 increase salary by 2% .
   3. If salary > 74000 and no. of clients >= 1 increase salary by 1% . iv. Display empID , salary , no . of clients , increased salary.

DELIMITER &&

CREATE PROCEDURE UpdateSalary(In empID varchar(10))

BEGIN

DECLARE empsalary INT;

DECLARE empClientCount INT;

DECLARE updatedsalary INT;

SELECT salary INTO empSalary FROM employee WHERE emp\_id = empID;

SELECT COUNT(client\_id) INTO empclientCount FROM works\_with WHERE emp\_id = empID;

IF empSalary > 49000 AND empClientCount > 2 THEN SET updatedsalary = empsalary \* 1.05;

ELSEIF empsalary BETWEEN 49000 AND 61000 AND empClientCount > 2 THEN SET updatedsalary = empSalary \* 1.02;

ELSEIF empSalary > 74000 AND empClientCount >= 1 THEN SET updatedsalary = empsalary 1.01;

ELSE SET updatedsalary = empSalary;

END IF;

SELECT empID AS emp\_id, empsalary AS salary, empClientCount AS no\_of\_clients, updatedsalary AS increased\_salary;

END &&

DELIMITER;

CALL UpdateSalary('103');

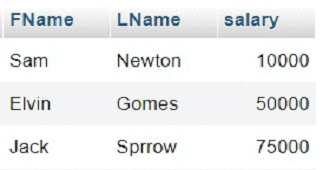
Output:



5.Write procedure which executes the same task as in procedure 4 but for all the employee in the database.

SELECT Employee.FName, Employee.LName, Employee.salary FROM Employee

Output:



DELIMITER &&

CREATE PROCEDURE Update\_All\_Salaries()

BEGIN

UPDATE employee SET salary = CASE WHEN salary > 49000 AND (SELECT COUNT(\*) FROM works\_with WHERE emp\_id = Employee.emp\_id) > 2 THEN salary \* 1.05 WHEN salary BETWEEN 49000 AND 61000 AND (SELECT COUNT(\*) FROM works\_with WHERE emp\_id = employee.emp\_id) = 2 THEN salary 1.02 WHEN salary > 74000 AND (SELECT COUNT(\*) FROM works with WHERE emp\_id = employee.emp\_id) >= 1 THEN salary \* 1.01

ELSE salary END;

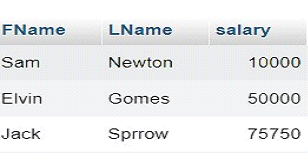
END &&

DELIMITER;

CALL Upadte\_All\_Salaries();

SELECT Employee.FName, Employee.LName, Employee.salary FROM Employee

Output:



**Conclusion**

Procedures were studied and the queries were successfully executed.