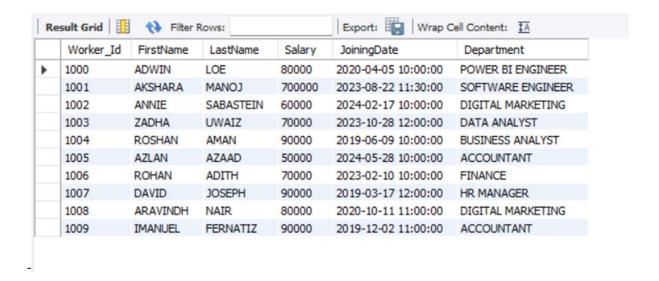
TASK - 8

Consider the Worker table with following fields:

- Worker_Id INT
- FirstName CHAR(25),
- LastName CHAR(25),
- Salary INT(15),
- JoiningDate DATETIME,
- Department CHAR(25))

```
CREATE DATABASE work;
2 •
      USE work;
3
4 • ⊖ CREATE TABLE worker (
          Worker Id INT,
5
6
          FirstName CHAR(25),
7
          LastName CHAR(25),
8
          Salary INT,
           JoiningDate DATETIME,
9
10
           Department CHAR(25)
     );
11
12
      INSERT INTO worker VALUES
13 •
       (1000, 'ADWIN', 'LOE', 80000, '2020-04-05 10:00:00', 'POWER BI ENGINEER'),
14
       (1001, 'AKSHARA', 'MANOJ', 700000, '2023-08-22 11:30:00', 'SOFTWARE ENGINEER'),
15
       (1002, 'ANNIE', 'SABASTEIN', 60000, '2024-02-17 10:00:00', 'DIGITAL MARKETING'),
16
       (1003, 'ZADHA', 'UWAIZ', 70000, '2023-10-28 12:00:00', 'DATA ANALYST'),
17
       (1004, 'ROSHAN', 'AMAN', 90000, '2019-06-09 10:00:00', 'BUSINESS ANALYST'),
18
       (1005, 'AZLAN', 'AZAAD', 50000, '2024-05-28 10:00:00', 'ACCOUNTANT'),
19
       (1006, 'ROHAN', 'ADITH', 70000, '2023-02-10 10:00:00', 'FINANCE'),
20
       (1007, 'DAVID', 'JOSEPH', 90000, '2019-03-17 12:00:00', 'HR MANAGER'),
21
       (1008, 'ARAVINDH', 'NAIR', 80000, '2020-10-11 11:00:00', 'DIGITAL MARKETING'),
22
       (1009, 'IMANUEL', 'FERNATIZ', 90000, '2019-12-02 11:00:00', 'ACCOUNTANT');
23
25 •
      SELECT * FROM worker;
26
```



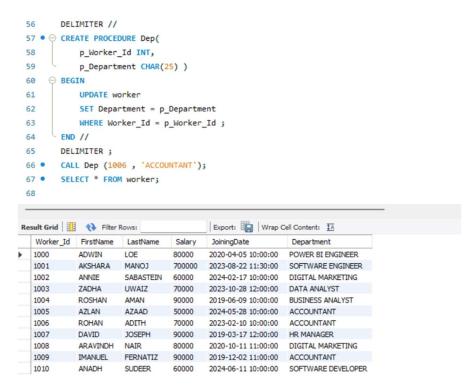
1. Create a stored procedure that takes in IN parameters for all the columns in the Worker table and adds a new record to the table and then invokes the procedure call.



2. Write stored procedure takes in an IN parameter for WORKER_ID and an OUT parameter for SALARY. It should retrieve the salary of the worker with the given ID and returns it in the p_salary parameter. Then make the procedure call.

```
43
       DELIMITER //
 44 • ⊖ CREATE PROCEDURE Sal(
 45
           IN p_Worker_Id INT,
 46
           OUT p Salary INT )
 47
           SELECT Salary INTO p_Salary FROM worker
 48
           WHERE Worker_Id = p_Worker_Id;
 49
      END //
 50
       DELIMITER ;
 51
 52
 53 • CALL Sal (1001, @Sal);
 54 • SELECT @Sal;
Export: Wrap Cel
   @Sal
700000
```

3. Create a stored procedure that takes in IN parameters for WORKER_ID and DEPARTMENT. It should update the department of the worker with the given ID. Then make a procedure call.



4. Write a stored procedure that takes in an IN parameter for DEPARTMENT and an OUT parameter for p_workerCount. It should retrieve the number of workers in the given department and returns it in the p_workerCount parameter. Make procedure call.

```
74
        DELIMITER //
 75
 76 • ⊖ CREATE PROCEDURE W Count(
 77
            IN p Department CHAR(25),
            OUT p_WorkersCount INT)
 78
 79

⊖ BEGIN

            SELECT COUNT(*) INTO p_WorkersCount FROM worker
 80
 81
            WHERE Department = p_Department;
       END //
 82
 83
 84
        DELIMITER ;
 85
        CALL W_Count('ACCOUNTANT', @W_count);
 86 •
        SELECT @W_count;
                                      Export: Wrap Cell Content: 1A
@W_count
) 3
```

5. Write a stored procedure that takes in an IN parameter for DEPARTMENT and an OUT parameter for p_avgSalary. It should retrieve the average salary of all workers in the given department and returns it in the p_avgSalary parameter and call the procedure.

```
88
         DELIMITER //
 89 • ○ CREATE PROCEDURE Avg_sal(
             IN p Department CHAR(25),
 90
 91
             OUT p_AvgSalary INT )

⊖ BEGIN

 92
             SELECT AVG(Salary) INTO p_AvgSalary FROM worker
 93
             WHERE Department = p_Department;
 94
        END //
 95
         DELIMITER ;
 96
 97
 98 •
         CALL Avg_sal('ACCOUNTANT', @AvgSal);
         SELECT @AvgSal;
 99 •
100
Result Grid Filter Rows:
                                         Export: Wrap Cell Content:
   @AvgSal
70000
```