STORED PROCEDURES

-- 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 the Worker Table
CREATE TABLE Worker (
   Worker_Id INT PRIMARY KEY,
   FirstName CHAR(25),
   LastName CHAR(25),
   Salary INT,
   JoiningDate DATETIME,
   Department CHAR(25)
);
DESC Worker;
```

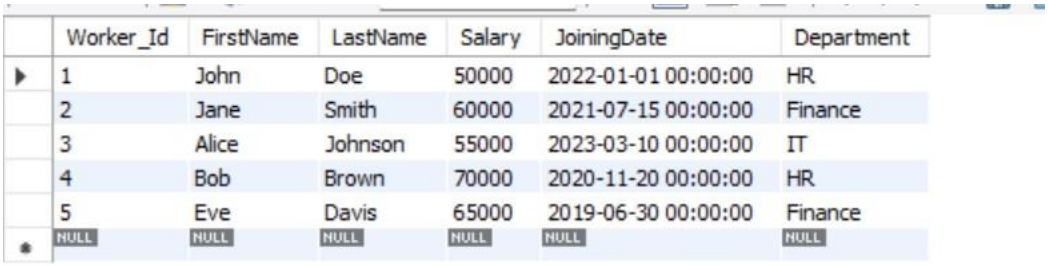
Field	Type	Null	Key	Default	Extra
Worker_Id	int	NO	PRI	NULL	
FirstName	char(25)	YES		NULL	
LastName	char(25)	YES		NULL	
Salary	int	YES		NULL	
JoiningDate	datetime	YES		NULL	
Department	char(25)	YES		NULL	

```
-- Insert  Data into the worker Table
INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)
VALUES
   (1, 'John', 'Doe', 50000, '2022-01-01', 'HR'),
   (2, 'Jane', 'Smith', 60000, '2021-07-15', 'Finance'),
   (3, 'Alice', 'Johnson', 55000, '2023-03-10', 'IT'),
```

(4, 'Bob', 'Brown', 70000, '2020-11-20', 'HR'),

(5, 'Eve', 'Davis', 65000, '2019-06-30', 'Finance');

SELECT * FROM Worker;



Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
1	John	Doe	50000	2022-01-01 00:00:00	HR
2	Jane	Smith	60000	2021-07-15 00:00:00	Finance
3	Alice	Johnson	55000	2023-03-10 00:00:00	IT
4	Bob	Brown	70000	2020-11-20 00:00:00	HR
5	Eve	Davis	65000	2019-06-30 00:00:00	Finance
NULL	NULL	NULL	NULL	NULL	NULL

-- 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

DELIMITER //

CREATE PROCEDURE NewWorker(

IN p_Worker_Id INT,

IN p_FirstName CHAR(25),

IN p_LastName CHAR(25),

IN p_Salary INT,

IN p_JoiningDate DATETIME,

IN p_Department CHAR(25)

)

BEGIN

INSERT INTO Worker (Worker_Id, FirstName, LastName, Salary, JoiningDate, Department)

VALUES (p_Worker_Id, p_FirstName, p_LastName, p_Salary, p_JoiningDate, p_Department);

```
END //
DELIMITER ;

CALL NewWorker(6, 'David', 'Johny', 45000, '2024-01-15', 'Marketing');
```

-- 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.

```
DELIMITER //
CREATE PROCEDURE GetWorkerSalary(IN p_Worker_Id INT,OUT p_Salary INT)
BEGIN
SELECT Salary INTO p_Salary FROM Worker
WHERE Worker_Id = p_Worker_Id;
END //
DELIMITER ;

SET @p_Salary = 0;
CALL GetWorkerSalary(2, @p_Salary);
SELECT @p_Salary AS WorkerSalary;
```

-- 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.

```
DELIMITER //
CREATE PROCEDURE UpdateWorkerDepartment(
   IN p_Worker_Id INT,
   IN p_Department CHAR(25)
)
BEGIN
   UPDATE Worker SET Department = p_Department WHERE Worker_Id = p_Worker_Id;
END //
DELIMITER ;

CALL UpdateWorkerDepartment(3, 'CIVIL');

select * from worker;
```

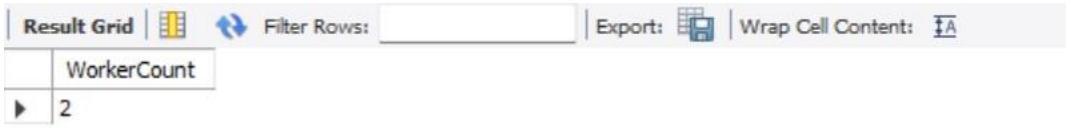
Worker_Id	FirstName	LastName	Salary	JoiningDate	Department
1	John	Doe	50000	2022-01-01 00:00:00	HR
2	Jane	Smith	60000	2021-07-15 00:00:00	Finance
3	Alice	Johnson	55000	2023-03-10 00:00:00	CIVIL
4	Bob	Brown	70000	2020-11-20 00:00:00	HR
5	Eve	Davis	65000	2019-06-30 00:00:00	Finance
6	David	Johny	45000	2024-01-15 00:00:00	Marketing
NULL	NULL	NULL	NULL	NULL	NULL

-- 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.

```
DELIMITER //
CREATE PROCEDURE GetWorkerCount(
   IN p_Department CHAR(25),
```

```
    OUT p_WorkerCount INT
)
BEGIN
    SELECT COUNT(*) INTO p_WorkerCount
    FROM Worker
    WHERE Department = p_Department;
END //
DELIMITER ;
```

```
SET @p_WorkerCount = 0;
CALL GetWorkerCount('HR', @p_WorkerCount);
SELECT @p_WorkerCount AS WorkerCount;
```



-- 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.

```
DELIMITER //
CREATE PROCEDURE GetAverageSalary(IN p_Department CHAR(25),OUT p_AvgSalary
DECIMAL(15,2))
BEGIN
    SELECT AVG(Salary) INTO p_AvgSalary
    FROM Worker
```

```
   WHERE Department = p_Department;
END //
DELIMITER ;


SET @p_AvgSalary = 0.00;
CALL GetAverageSalary('HR', @p_AvgSalary);
SELECT @p_AvgSalary AS AverageSalary;
```