Create procedure or functions for employee table

1. Add 5000 bonus to all employee

2. Print same name employees

3. Print highest and lowest salary from employee table

--🡪DELIMITER $$

CREATE PROCEDURE add\_bonus()

BEGIN

UPDATE employee

SET salary = salary + 5000;

END $$

DELIMITER ;

---🡪 DELIMITER $$

CREATE PROCEDURE print\_duplicate\_names()

BEGIN

SELECT emp\_name, COUNT(\*) AS count

FROM employee

GROUP BY emp\_name

HAVING COUNT(\*) > 1;

END $$

DELIMITER ;

-🡪 DELIMITER $$

CREATE PROCEDURE salary\_stats()

BEGIN

SELECT

MAX(salary) AS highest\_salary,

MIN(salary) AS lowest\_salary

FROM employee;

END $$

DELIMITER ;

**CREATE PROCEDURE add\_bonus()**

CREATE PROCEDURE add\_bonus\_amount(IN bonus DECIMAL(10,2))

BEGIN

UPDATE employee SET salary = salary + bonus;

END;

**CREATE PROCEDURE print\_duplicate\_names()**

SELECT \*

FROM employee

WHERE emp\_name IN (

SELECT emp\_name

FROM employee

GROUP BY emp\_name

HAVING COUNT(\*) > 1

);

2. Create procedure or functions for Hospital table

1. print avg patient count on daily basis

2. print all the patients whose belong to same ward

3. arrange the patients list according their admission date

CREATE TABLE hospital (

patient\_id INT PRIMARY KEY,

patient\_name VARCHAR(50),

ward VARCHAR(50),

admission\_date DATE,

daily\_patient\_count INT

);

INSERT INTO hospital VALUES

(1, 'John', 'A', '2025-08-10', 5),

(2, 'Alice', 'B', '2025-08-11', 3),

(3, 'Bob', 'A', '2025-08-12', 4),

(4, 'Eve', 'C', '2025-08-10', 2),

(5, 'Charlie', 'B', '2025-08-11', 3);

-🡪 DELIMITER $$

CREATE PROCEDURE avg\_daily\_patient\_count()

BEGIN

SELECT AVG(daily\_patient\_count) AS avg\_patient\_count

FROM hospital;

END $$

DELIMITER ;

🡪 DELIMITER $$

CREATE PROCEDURE patients\_same\_ward()

BEGIN

SELECT ward, patient\_name

FROM hospital

WHERE ward IN (

SELECT ward

FROM hospital

GROUP BY ward

HAVING COUNT(\*) > 1

)

ORDER BY ward, patient\_name;

END $$

DELIMITER ;