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

CREATE TABLE employee (

emp\_id INT PRIMARY KEY,

emp\_name VARCHAR(50),

salary DECIMAL(10,2)

);

INSERT INTO employee VALUES

(1, 'Arun', 50000),

(2, 'Mani', 40000),

(3, 'Shankar', 55000),

(4, 'Charan', 60000),

(5, 'Ajay', 40000);

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 ;

**package** JDBC\_conn;

**import** java.sql.\*;

**public** **class** EmployeeProcedureDemo {

**public** **static** **void** main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String pass = "root";

**try** (Connection con = DriverManager.*getConnection*(url, user, pass)) {

// 1. Add Bonus

CallableStatement cs1 = con.prepareCall("{CALL add\_bonus()}");

cs1.execute();

System.***out***.println("Bonus added successfully.\n");

// 2. Print Employees with Duplicate Names

CallableStatement cs2 = con.prepareCall("{CALL print\_duplicate\_names()}");

ResultSet rs1 = cs2.executeQuery();

System.***out***.println("Employees with same names:");

**while** (rs1.next()) {

System.***out***.println(rs1.getString("emp\_name") + " - " + rs1.getInt("count") + " times");

}

// 3. Highest and Lowest Salary

CallableStatement cs3 = con.prepareCall("{CALL salary\_stats()}");

ResultSet rs2 = cs3.executeQuery();

**if** (rs2.next()) {

System.***out***.println("\nHighest Salary: " + rs2.getDouble("highest\_salary"));

System.***out***.println("Lowest Salary: " + rs2.getDouble("lowest\_salary"));

}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

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,

name VARCHAR(100),

ward\_no INT,

admission\_date DATE

);

drop table hospital;

INSERT INTO hospital VALUES

(1, 'vijay', 101, '2025-08-01'),

(2, 'arun', 102, '2025-08-01'),

(3, 'ajay', 101, '2025-08-01'),

(4, 'akshok', 103, '2025-08-02'),

(5, 'akshay', 101, '2025-08-02');

DELIMITER $$

CREATE PROCEDURE avg\_patient\_count\_daily()

BEGIN

SELECT AVG(cnt) AS avg\_patient\_per\_day

FROM (

SELECT admission\_date, COUNT(\*) AS cnt

FROM hospital

GROUP BY admission\_date

) AS sub;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE patients\_same\_ward()

BEGIN

SELECT h1.\*

FROM hospital h1

JOIN (

SELECT ward\_no

FROM hospital

GROUP BY ward\_no

HAVING COUNT(\*) > 1

) h2 ON h1.ward\_no = h2.ward\_no;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE patients\_by\_admission()

BEGIN

SELECT \* FROM hospital ORDER BY admission\_date;

END $$

DELIMITER ;

**package** JDBC\_conn;

**import** java.sql.\*;

**public** **class** Hospital\_data{

**public** **static** **void** main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydb";

String user = "root";

String password = "root";

**try** (Connection con = DriverManager.*getConnection*(url, user, password)) {

System.***out***.println("Connected to DB...");

// 1. Average patient count daily

CallableStatement cs1 = con.prepareCall("{CALL avg\_patient\_count\_daily()}");

ResultSet rs1 = cs1.executeQuery();

**while** (rs1.next()) {

System.***out***.println("Average patients per day: " + rs1.getDouble("avg\_patient\_per\_day"));

}

// 2. Patients in same ward

CallableStatement cs2 = con.prepareCall("{CALL patients\_same\_ward()}");

ResultSet rs2 = cs2.executeQuery();

System.***out***.println("\nPatients in same ward:");

**while** (rs2.next()) {

System.***out***.println(rs2.getInt("patient\_id") + " - " + rs2.getString("name") +

" (Ward: " + rs2.getInt("ward\_no") + ")");

}

// 3. Patients ordered by admission date

CallableStatement cs3 = con.prepareCall("{CALL patients\_by\_admission()}");

ResultSet rs3 = cs3.executeQuery();

System.***out***.println("\nPatients by admission date:");

**while** (rs3.next()) {

System.***out***.println(rs3.getInt("patient\_id") + " - " + rs3.getString("name") +

" (Admitted: " + rs3.getDate("admission\_date") + ")");

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}