DAY12_ASSIGNMENT

1.Create procedure or functions for employee table

    1. Add 5000 bonus to all employee

    2. Print same name employees

Print highest and lowest salary from employee table

A. **SQL CODE:**

```
create database mydatabase;
use mydatabase;
#create table
create table employee(
    emp_id int primary key,
    emp_name varchar(50),
    salary decimal(10,2)
);
#print information
show tables;
select * from employee;
#insert records
insert into employee values
(101, 'Neeva Sharma', 5000),
(102, 'Reeva Varma', 55000),
(103, 'Shiva Upadhyay', 45000),
(104, 'David Shah', 55000),
(105, 'Eve Kapoor', 40000);
# create procedure add_bonus
DELIMITER $$
CREATE PROCEDURE add_bonus()
BEGIN
    UPDATE employee
    SET salary = salary + 5000;
END $$
```

```sql
DELIMITER ;
# create procedure print_duplicate_names
DELIMITER $$
CREATE PROCEDURE print_duplicate_names()
BEGIN
    SELECT emp_name, COUNT(*) as count
    FROM employee
    GROUP BY emp_name
    HAVING COUNT(*) > 1;
END $$
DELIMITER ;
# create procedure salary_stats
DELIMITER $$
CREATE PROCEDURE salary_stats()
BEGIN
    SELECT
        MAX(salary) AS highest_salary,
        MIN(salary) AS lowest_salary
    FROM employee;
END $$
DELIMITER ;
```

JAVA CODE:

```java
package JDBS_conn;
import java.sql.*;
public class EmployeeProcedureApp {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        String user = "root";
        String pass = "mysql_password";
        try (Connection con = DriverManager.getConnection(url, user, pass)) {
            CallableStatement cs1 = con.prepareCall("{CALL add_bonus()}");
```

```java
        cs1.execute();
        System.out.println("Bonus added successfully.\n");
        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");
        }
        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

A.SQL CODE:

```sql
create database hospital_db;
USE hospital_db;
CREATE TABLE hospital (
   patient_id INT PRIMARY KEY,
   name VARCHAR(100),
```

```sql
    ward_no INT,
    admission_date DATE
);
drop table hospital;
INSERT INTO hospital VALUES
(1, 'Kajal devgan', 101, '2025-08-01'),
(2, 'Kareena kapoor', 102, '2025-08-01'),
(3, 'Mrunal thakur', 101, '2025-08-01'),
(4, 'Priyanka chopra', 103, '2025-08-02'),
(5, 'Kareeshama kapoor', 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 ;
```

JAVA CODE:

```java
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...");
            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"));
            }
            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")
+
```

```java
                    " (Ward: " + rs2.getInt("ward_no") + ")");
        }
        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();
    }
  }
}
```