DAY12 ASSIGNMENT

1. 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.

Ans)

```
 CREATE TABLE employee (
   empid INT PRIMARY KEY,
   name VARCHAR(100),
   salary int
);

-- insert data
INSERT INTO employee VALUES (101, 'manasa', 30000);
INSERT INTO employee VALUES (201, 'shivani', 35000);
INSERT INTO employee VALUES (301, 'shruthi', 40000);
INSERT INTO employee VALUES (401, 'manasa', 25000);

Delimiter //
create procedure employeesalary11()
begin
update employee set salary=salary+5000;
end//
```

```
Delimiter //
create procedure sameemployeenames1()
begin
select name from employee group by name having count(*)>1;
end //

Delimiter //
create procedure highestandlowestsalary1()
begin
select max(salary),min(salary) from employee;
end //

package jdbc;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
public class collable_statement {
	public static void main(String[] args) {
			String url = "jdbc:mysql://localhost:3306/mydb";
		String user = "root";
		String password = "Nikhitha123";
```

```
            try
                (Connection con = DriverManager.getConnection(url,
user, password)){
                CallableStatement cst0 = con.prepareCall("CALL
employeesalary11()");
                cst0.execute();
                System.out.println("bonus added to all employess");
                CallableStatement cst1 = con.prepareCall("CALL
sameemployeenames1()");
                ResultSet rs=cst1.executeQuery();
                System.out.println("same employees names: ");
                while(rs.next()) {
                 System.out.println(rs.getString("name"));
                }
                CallableStatement cst2 = con.prepareCall("CALL
highestandlowestsalary1()");
                ResultSet rs1=cst2.executeQuery();
                if(rs1.next())
                {
                 System.out.println("highest salary:"+rs1.getInt(1));
                 System.out.println("highest salary:"+rs1.getInt(2));

                }
            }
```

```
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }
}
```

Output:

bonus added to all employess

same employees names:

manasa

highest salary:75000

highest salary:60000

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

Ans)

```
CREATE TABLE hospital (
   hospitalname VARCHAR(100),
   ward VARCHAR(50),
   patientname VARCHAR(100),
   admissiondate DATE
```

```
);
INSERT INTO hospital (hospitalname, ward, patientname,
admissiondate) VALUES
('Amma Hospital', 'Ward-A', 'manasa', '2025-08-01'),
('Amma Hospital', 'Ward-B', 'shruthi', '2025-08-01'),
('Amma Hospital', 'Ward-B', 'shivani', '2025-08-03'),
('Amma Hospital', 'Ward-A', 'sath', '2025-08-02');

DELIMITER //
CREATE PROCEDURE avgPatientCountPerDay()
BEGIN
   SELECT AVG(count_per_day) AS avg_patients
   FROM (
      SELECT admission_date, COUNT(*) AS count_per_day
      FROM hospital
      GROUP BY admission_date
   ) AS sub;
END //

DELIMITER //
CREATE PROCEDURE sameWardPatients()
BEGIN
   SELECT * FROM hospital
```

```
    WHERE ward IN (
        SELECT ward FROM hospital
        GROUP BY ward
        HAVING COUNT(*) > 1
    );
END //

DELIMITER //
CREATE PROCEDURE sortPatientsByAdmission()
BEGIN
    SELECT * FROM hospital
    ORDER BY admission_date ASC;
END //
```

```
package jdbc;
import java.sql.*;
public class HospitalProcedureDemo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "Nikhitha123";
        try (Connection con = DriverManager.getConnection(url, user,
password)) {
```

```
        CallableStatement cs1 = con.prepareCall("CALL
avgPatientCountPerDay()");
        ResultSet rs1 = cs1.executeQuery();
        if (rs1.next()) {
            System.out.println("Average Patients Per Day: " +
rs1.getFloat("avg_patients"));
        }
        CallableStatement cs2 = con.prepareCall("CALL
sameWardPatients()");
        ResultSet rs2 = cs2.executeQuery();
        System.out.println("Patients in Same Ward:");
        while (rs2.next()) {
            System.out.println("Hospital: " +
rs2.getString("hospitalname") +
                    ", Ward: " + rs2.getString("ward") +
                    ", Patient: " + rs2.getString("patientname") +
                    ", Admission Date: " +
rs2.getDate("admissiondate"));
        }
        CallableStatement cs3 = con.prepareCall("CALL
sortPatientsByAdmission()");
        ResultSet rs3 = cs3.executeQuery();
        System.out.println("Patients Sorted by Admission Date:");
        while (rs3.next()) {
```

```
            System.out.println("Hospital: " +
rs3.getString("hospitalname") +
                               ", Ward: " + rs3.getString("ward") +
                               ", Patient: " + rs3.getString("patientname") +
                               ", Admission Date: " +
rs3.getDate("admissiondate"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

Average Patients Per Day: 1.3333

Patients with Same Ward:

Hospital: Amma Hospital, Ward: Ward-A, Patient: manasa

Hospital: Amma Hospital, Ward: Ward-B, Patient: shruthi

Hospital: Amma Hospital, Ward: Ward-B, Patient: shivani

Hospital: Amma Hospital, Ward: Ward-A, Patient: sath

Patients Sorted by Admission Date:

Hospital: Amma Hospital, Patient: manasa, Admission Date: 2025-08-01

Hospital: Amma Hospital, Patient: shruthi, Admission Date: 2025-08-01

Hospital: Amma Hospital, Patient: sath, Admission Date: 2025-08-02

Hospital: Amma Hospital, Patient: shivani, Admission Date: 2025-08-03