OOPS MINI PROJECT

POLICE STATION MANAGEMENT SYSTEM

AIM:

To construct a database for the police station management system and connect it with my SQL using java.

Alogrithm:

```
1. Start
2. Connect to the MySQL database.
3. Display a menu with the following options:
•   Add Officer
•   Add Case
•   Add Criminal
•   Add Complainant
•   Exit
4. Ask the user to choose an option.
5. Based on the user’s choice:
•   If Add Officer:
        ▪ Take input: Officer Name, Rank, Contact Number, and Address.
        ▪ Insert this data into the Officers table in the database.
•   If Add Case:
        ▪ Take input: Case Name, Officer ID, Date Reported, and Status.
        ▪ Check if the Officer ID exists in the Officers table.
        ▪ If valid, insert the data into the Cases table.
•   If Add Criminal:
        ▪ Take input: Criminal Name, Crime Committed, Case ID, and Arrest Date.
        ▪ Check if the Case ID exists in the Cases table.
        ▪ If valid, insert the data into the Criminals table.
•   If Add Complainant:
        ▪ Take input: Complainant Name, Contact Number, Address, and Case ID.
        ▪ Check if the Case ID exists in the Cases table.
        ▪ If valid, insert the data into the Complainants table.
•   If Exit:
        ▪ Close the database connection.
        ▪ End the program.
6. Repeat the process until the user chooses to exit.
7.  Stop
```

SQL QURIES:

Step 1: Create the database

```
CREATE DATABASE PoliceStationDB;
```

Use the database

```
USE PoliceStationDB;
```

Step 2: Create the Officers table

```
CREATE TABLE Officers (
officer_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
rank VARCHAR(50) NOT NULL,
contact_number VARCHAR(15) NOT NULL,
station_name VARCHAR(100) NOT NULL
);
```

Step 3: Create the Cases table

```
CREATE TABLE Cases (
case_id INT PRIMARY KEY AUTO_INCREMENT,
case_type VARCHAR(100) NOT NULL,
description TEXT NOT NULL,
case_status VARCHAR(50) NOT NULL,
officer_id INT,
FOREIGN KEY (officer_id) REFERENCES Officers(officer_id)
);
```

Step 4: Create the Criminals table

```
CREATE TABLE Criminals (
criminal_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
age INT NOT NULL,
gender VARCHAR(10),
crime VARCHAR(100),
case_id INT,
FOREIGN KEY (case_id) REFERENCES Cases(case_id)
);
```

Step 5: Create the Complainants table

```
CREATE TABLE Complainants (
complainant_id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
contact_number VARCHAR(15) NOT NULL,
address TEXT NOT NULL,
complaint_date DATE NOT NULL,
case_id INT,
FOREIGN KEY (case_id) REFERENCES Cases(case_id)
);
```

JAVA PROGRAM:

```
import java.sql.*;
import java.util.Scanner;

public class PoliceStationManagement {
public static void main(String[] args) {
String url = "jdbc:mysql://localhost:3306/PoliceStationManagement";
String user = "root"; // Replace with your database username
String password = "password"; // Replace with your database password

try (Connection conn = DriverManager.getConnection(url, user, password);
Scanner scanner = new Scanner(System.in)) {
System.out.println("Connected to the database!");

while (true) {
System.out.println("Choose an operation:");
System.out.println("1. Add Officer");
System.out.println("2. Add Case");
System.out.println("3. Add Criminal");
System.out.println("4. Add Complainant");
System.out.println("5. Exit");
int choice = scanner.nextInt();
scanner.nextLine(); // Consume newline

switch (choice) {
case 1 -> addOfficer(conn, scanner);
case 2 -> addCase(conn, scanner);
case 3 -> addCriminal(conn, scanner);
case 4 -> addComplainant(conn, scanner);
case 5 -> {
System.out.println("Exiting...");
return;
}
default -> System.out.println("Invalid choice. Try again.");
}
}
} catch (SQLException e) {
e.printStackTrace();
}
}

private static void addOfficer(Connection conn, Scanner scanner) throws SQLException {
System.out.println("Enter Officer Name:");
String name = scanner.nextLine();
System.out.println("Enter Rank:");
String rank = scanner.nextLine();
System.out.println("Enter Contact Number:");
String contactNumber = scanner.nextLine();
System.out.println("Enter Address:");
String address = scanner.nextLine();
String query = "INSERT INTO Officers (Name, Rank, ContactNumber, Address) VALUES (?, ?, ?, ?)";
try (PreparedStatement stmt = conn.prepareStatement(query)) {
stmt.setString(1, name);
stmt.setString(2, rank);
stmt.setString(3, contactNumber);
stmt.setString(4, address);
stmt.executeUpdate();
System.out.println("Officer added successfully!");
}
}

private static void addCase(Connection conn, Scanner scanner) throws SQLException {
System.out.println("Enter Case Name:");
String caseName = scanner.nextLine();
System.out.println("Enter Officer ID:");
int officerId = scanner.nextInt();
scanner.nextLine(); // Consume newline
System.out.println("Enter Date Reported (YYYY-MM-DD):");
String dateReported = scanner.nextLine();
System.out.println("Enter Status:");
String status = scanner.nextLine();

String query = "INSERT INTO Cases (CaseName, OfficerID, DateReported, Status) VALUES (?, ?, ?, ?)";
try (PreparedStatement stmt = conn.prepareStatement(query)) {
stmt.setString(1, caseName);
stmt.setInt(2, officerId);
stmt.setDate(3, Date.valueOf(dateReported));
stmt.setString(4, status);
stmt.executeUpdate();
System.out.println("Case added successfully!");
}
}

private static void addCriminal(Connection conn, Scanner scanner) throws SQLException {
System.out.println("Enter Criminal Name:");
String name = scanner.nextLine();
System.out.println("Enter Crime Committed:");
String crimeCommitted = scanner.nextLine();
System.out.println("Enter Case ID:");
int caseId = scanner.nextInt();
scanner.nextLine(); // Consume newline
System.out.println("Enter Arrest Date (YYYY-MM-DD):");
String arrestDate = scanner.nextLine();

String query = "INSERT INTO Criminals (Name, CrimeCommitted, CaseID, ArrestDate) VALUES (?, ?, ?,
?)";
try (PreparedStatement stmt = conn.prepareStatement(query)) {
stmt.setString(1, name);
stmt.setString(2, crimeCommitted);
stmt.setInt(3, caseId);
stmt.setDate(4, Date.valueOf(arrestDate));
stmt.executeUpdate();
System.out.println("Criminal added successfully!");
}
}

private static void addComplainant(Connection conn, Scanner scanner) throws SQLException {
System.out.println("Enter Complainant Name:");
String name = scanner.nextLine();
System.out.println("Enter Contact Number:");
String contactNumber = scanner.nextLine();
System.out.println("Enter Address:");
String address = scanner.nextLine();
System.out.println("Enter Case ID:");
int caseId = scanner.nextInt();
scanner.nextLine(); // Consume newline

String query = "INSERT INTO Complainants (Name, ContactNumber, Address, CaseID) VALUES (?, ?, ?,
?)";
try (PreparedStatement stmt = conn.prepareStatement(query)) {
stmt.setString(1, name);
stmt.setString(2, contactNumber);
stmt.setString(3, address);
stmt.setInt(4, caseId);
stmt.executeUpdate();
System.out.println("Complainant added successfully!");
}
}
}
```

INPUT:

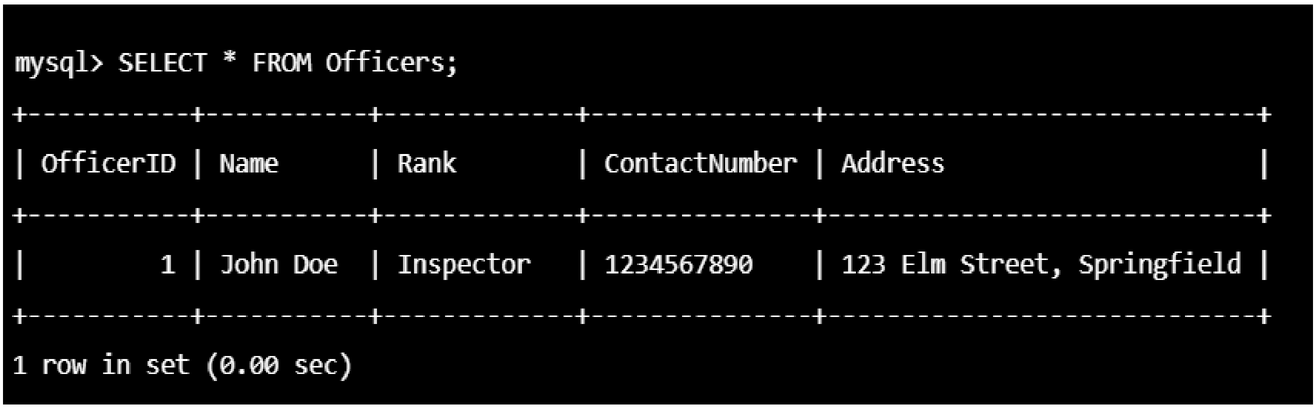
```
Choose an operation:
1. Add Officer
2. Add Case
3. Add Criminal
4. Add Complainant
5. Exit
1
Enter Officer Name: John Doe
Enter Rank: Inspector
Enter Contact Number: 9876543210
Enter Address: 123 Elm Street, Springfield
Officer added successfully!

Choose an operation:
1. Add Officer
2. Add Case
3. Add Criminal
4. Add Complainant
```

5. Exit
2
Enter Case Name: Bank Robbery
Enter Officer ID: 1
Enter Date Reported (YYYY-MM-DD): 2024-11-01
Enter Status: Open
Case added successfully!

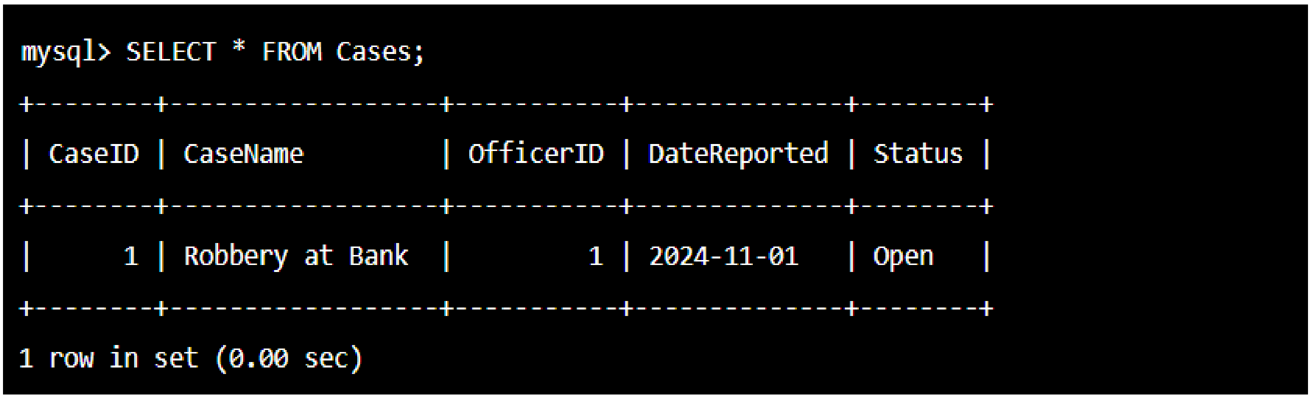
OUTPUT:

Query 1: SELECT * FROM Officers;



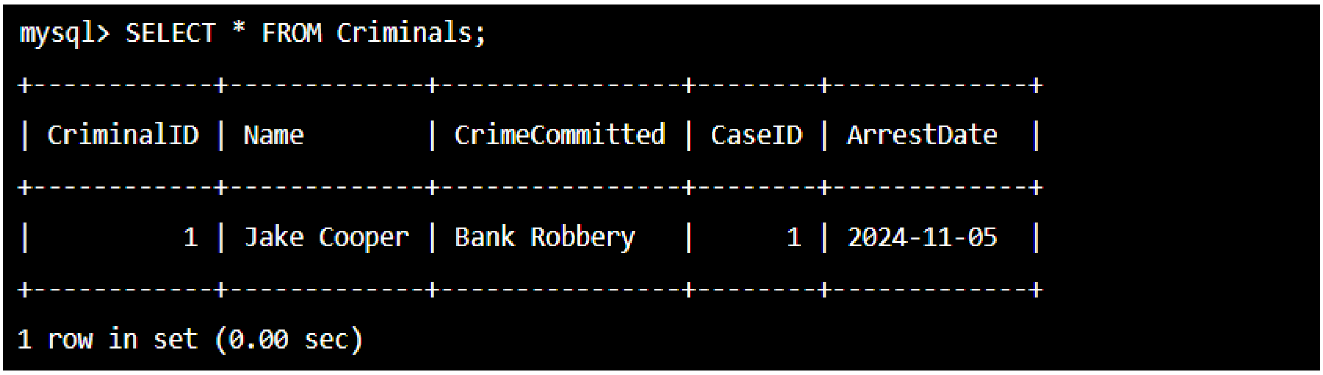
```
mysql> SELECT * FROM Officers;
+-----------+----------+-----------+---------------+-----------------------------+
| OfficerID | Name     | Rank      | ContactNumber | Address                     |
+-----------+----------+-----------+---------------+-----------------------------+
|         1 | John Doe | Inspector | 1234567890    | 123 Elm Street, Springfield |
+-----------+----------+-----------+---------------+-----------------------------+
1 row in set (0.00 sec)
```

Query 2: SELECT * FROM Cases;



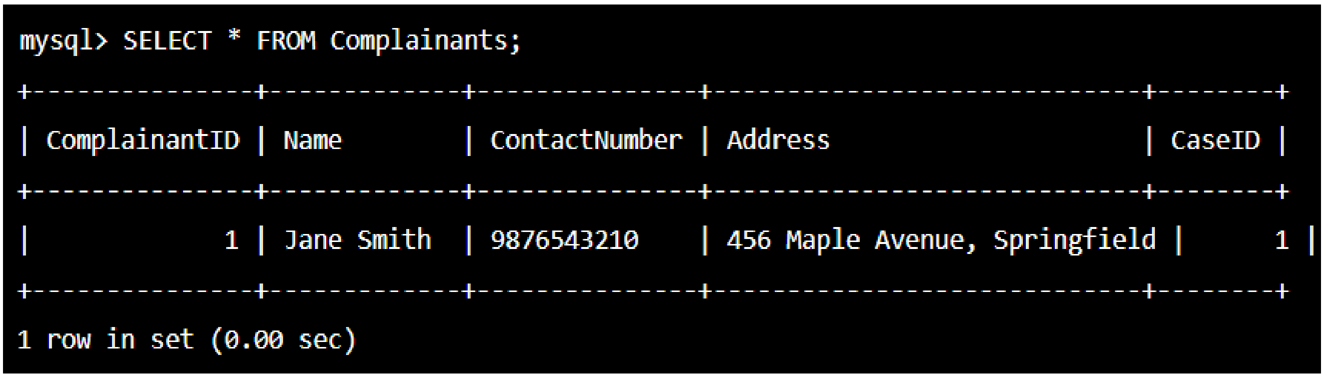
```
mysql> SELECT * FROM Cases;
+--------+------------------+-----------+--------------+--------+
| CaseID | CaseName         | OfficerID | DateReported | Status |
+--------+------------------+-----------+--------------+--------+
|      1 | Robbery at Bank  |         1 | 2024-11-01   | Open   |
+--------+------------------+-----------+--------------+--------+
1 row in set (0.00 sec)
```

Query 3: SELECT * FROM Criminals;



```
mysql> SELECT * FROM Criminals;
+------------+-------------+----------------+--------+-------------+
| CriminalID | Name        | CrimeCommitted | CaseID | ArrestDate  |
+------------+-------------+----------------+--------+-------------+
|          1 | Jake Cooper | Bank Robbery   |      1 | 2024-11-05  |
+------------+-------------+----------------+--------+-------------+
1 row in set (0.00 sec)
```

Query 4: SELECT * FROM Complainants;



```
mysql> SELECT * FROM Complainants;
+-------------+------------+---------------+-----------------------------+--------+
| ComplainantID | Name       | ContactNumber | Address                     | CaseID |
+-------------+------------+---------------+-----------------------------+--------+
|           1 | Jane Smith | 9876543210    | 456 Maple Avenue, Springfield |      1 |
+-------------+------------+---------------+-----------------------------+--------+
1 row in set (0.00 sec)
```

RESULT:

The database construction for the police station management system has been successfully complected and connected with mySQL using java.