1. Write a Java program to connect to a MySQL database using JDBC.

Program :

```
package JDBC_conn;
import java.sql.Connection;
import java.sql.DriverManager;
public class DB_Conn {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            // Load MySQL JDBC Driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Establish connection
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("Connection created");

            // Close connection
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : Connection created

2. Create a Java class to insert student records into a database table.

Program :

```
package JDBC_conn;
import java.sql.*;
```

```
public class Insert_data {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            String query = "INSERT INTO student(id, name, percentage) VALUES (1, 'Dev', 85.5)";
            Statement st = con.createStatement();
            int rows = st.executeUpdate(query);
            System.out.println(rows + " record inserted successfully!");
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : 1 record inserted successfully

3. Write a JDBC program to fetch and display all student records from the database.

Program :

```
package JDBC_conn;
import java.sql.*;
public class Fetch_data {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            String query = "SELECT * FROM student";
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(query);
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") +
                                    ", Name: " + rs.getString("name") +
                                    ", Percentage: " + rs.getDouble("percentage"));
            }
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : ID: 1, Name: Dev, Percentage: 85.5

4. Develop a program to search a student by ID using JDBC.

Program :

```
package JDBC_conn;
import java.sql.*;

public class Search {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);

            int searchId = 1;
            String query = "SELECT * FROM student WHERE id = " + searchId;
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(query);

            if (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") +
                                   ", Name: " + rs.getString("name") +
                                   ", Percentage: " + rs.getDouble("percentage"));
            } else {
                System.out.println("No student found with ID " + searchId);
            }
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : ID: 1, Name: Dev, Percentage: 85.5

5. Implement an update operation to modify student details in the database using JDBC.

Program :

```
package JDBC_conn;
import java.sql.*;
public class UpdateStudent {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            String query = "UPDATE student SET percentage = 90.0 WHERE id = 1";
            Statement st = con.createStatement();
            int rows = st.executeUpdate(query);
            System.out.println(rows + " record updated successfully!");
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : 1 record updated successfully!

6. Write a Java program to delete a student record from the database using JDBC.

Program :

```
package JDBC_conn;
import java.sql.*;
public class DeleteStudent {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            String query = "DELETE FROM student WHERE id = 1";
            Statement st = con.createStatement();
            int rows = st.executeUpdate(query);
            System.out.println(rows + " record deleted successfully!");
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : 1 record deleted successfully!

7. Design a Java application to perform all CRUD (Create, Read, Update, Delete) operations on an **Employee** table using JDBC.

Program :

```
package JDBC_conn;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
public class CRUD_operations{
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("Connection created");
            Statement stmt = con.createStatement();
            String insertQuery = "INSERT INTO emp VALUES (7, 'Rohit', 55000)";
            int rowsInserted = stmt.executeUpdate(insertQuery);
            System.out.println("Rows Inserted: " + rowsInserted);
            System.out.println("Employee Records");
```

```
      ResultSet rs = stmt.executeQuery("SELECT * FROM emp");
      while (rs.next()) {
        System.out.println(rs.getInt("id") + " | " +
                  rs.getString("name") + " | " +
                  rs.getInt("salary"));
      }
      String updateQuery = "UPDATE emp SET salary = 60000 WHERE id = 2";
      int rowsUpdated = stmt.executeUpdate(updateQuery);
      System.out.println("Rows Updated: " + rowsUpdated);
      String deleteQuery = "DELETE FROM emp WHERE id = 6";
      int rowsDeleted = stmt.executeUpdate(deleteQuery);
      System.out.println("Rows Deleted: " + rowsDeleted);
      con.close();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

```
Output : Connection created
Rows Inserted: 1
Employee Records
1 | Dev | 80000
2 | Yoga | 90000
3 | Muktha | 85000
4 | Dev | 95000
5 | Yoga | 100000
6 | Sai | 75000
7 | Rohit | 55000

Rows Updated: 1
Rows Deleted: 1
```

8. Create a JDBC-based program to count the total number of rows in a table.

Program :

```java
package JDBC_conn;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class CountRows {
  public static void main(String[] args) {
    String url = "jdbc:mysql://localhost:3306/mydb";
    String user = "root";
    String password = "798167";
    try {
      Class.forName("com.mysql.cj.jdbc.Driver");
      Connection con = DriverManager.getConnection(url, user, password);
      System.out.println("Connection created");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT COUNT(*) AS total FROM emp");
      if (rs.next()) {
        System.out.println("rows : " + rs.getInt("total"));
      }
      con.close();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

```
Output : Connection created
rows : 6
```

9. Develop a program to sort student data in ascending order by name using SQL in JDBC.

Program :

```
package JDBC_conn;
import java.sql.*;

public class SortStudents {
   public static void main(String[] args) {
      String url = "jdbc:mysql://localhost:3306/mydb";
      String user = "root";
      String password = "798167";

      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         Connection con = DriverManager.getConnection(url, user, password);

         String query = "SELECT * FROM student ORDER BY name ASC";
         Statement stmt = con.createStatement();
         ResultSet rs = stmt.executeQuery(query);
         System.out.println("ID\tName\t\tPercentage");
         while (rs.next()) {
            System.out.println(rs.getInt("id") + "\t" +
                         rs.getString("name") + "\t\t" +
                         rs.getFloat("percentage"));
         }
         con.close();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

Output :

```
ID                  Name            Percentage

5                   Arun            70.0

1                   Dev             80.5

3                   Muktha          75.0

4                   Sai             88.5

2                   Yoga            90.0
```

10. Write a program to display all students whose percentage is greater than 75 using JDBC and SQL WHERE clause.

Program :

```
package JDBC_conn;

import java.sql.*;


public class Student_Cond {

  public static void main(String[] args) {

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

    String user = "root";

    String password = "798167";


    try {

      Class.forName("com.mysql.cj.jdbc.Driver");

      Connection con = DriverManager.getConnection(url, user, password);


      String query = "SELECT * FROM student WHERE percentage > 75";

      Statement stmt = con.createStatement();

      ResultSet rs = stmt.executeQuery(query);


      System.out.println("ID\tName\t\tPercentage");

      while (rs.next()) {

        System.out.println(rs.getInt("id") + "\t" +
                                    rs.getString("name") + "\t\t" +
                                    rs.getFloat("percentage"));
            }

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

Output :

```
ID   Name              Percentage
1    Dev               80.5
2    Yoga              90.0
4    Sai               88.5
```

11. Use **PreparedStatement** to insert multiple student records into the database.

Program :

```
package JDBC_conn;
import java.sql.*;
import java.util.Scanner;
public class PSInsert {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        Scanner sc = new Scanner(System.in);
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            String query = "INSERT INTO student (id, name, percentage) VALUES (?, ?, ?)";
            PreparedStatement pstmt = con.prepareStatement(query);
            System.out.print("Enter number of students to insert: ");
            int n = sc.nextInt();
            sc.nextLine();
            for (int i = 1; i <= n; i++) {
                System.out.println("Enter details for student " + i);
                System.out.print("ID: ");
                int id = sc.nextInt();
                sc.nextLine();
                System.out.print("Name: ");
                String name = sc.nextLine();
                System.out.print("Percentage: ");
                float perc = sc.nextFloat();
                sc.nextLine();

                pstmt.setInt(1, id);
                pstmt.setString(2, name);
                pstmt.setFloat(3, perc);
                pstmt.executeUpdate();
            }
            System.out.println("Records inserted successfully!");
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        sc.close();
    }
}
```

```
Output : Enter number of students to insert: 2
Enter details for student 1
ID: 6
```

```
Name: Kiran

Percentage: 78.5

Enter details for student 2

ID: 7

Name: Priya

Percentage: 88.0

Records inserted successfully!
```

12. Implement a program using **transaction management** in JDBC (i.e., commit and rollback).

Program :

```
package JDBC_conn;

import java.sql.*;


public class Transaction {

    public static void main(String[] args) {

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

        String user = "root";

        String password = "798167";


        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection con = DriverManager.getConnection(url, user, password);

            con.setAutoCommit(false);


            String query = "INSERT INTO student (id, name, percentage) VALUES (?, ?, ?)";

            PreparedStatement pstmt = con.prepareStatement(query);

            pstmt.setInt(1, 8);

            pstmt.setString(2, "Anjali");

            pstmt.setFloat(3, 82.5f);

            pstmt.executeUpdate();
            pstmt.setInt(1, 8);
            pstmt.setString(2, "Rahul");
            pstmt.setFloat(3, 91.0f);
            pstmt.executeUpdate();
            con.commit();
            System.out.println("Both records inserted successfully!");
            con.close();
        } catch (Exception e) {
            System.out.println("Error occurred: " + e.getMessage());
            try {
                Connection con = DriverManager.getConnection(url, user, password);
                con.rollback();
                System.out.println("Transaction rolled back.");
            } catch (SQLException se) {
                se.printStackTrace();
            }
        }
    }
}
```

```
Output : Error occurred: Duplicate entry '8' for key 'PRIMARY'
Transaction rolled back.
```

13. Write a JDBC program to handle exceptions (like invalid ID, connection errors) gracefully.

Program :

```
package JDBC_conn;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class ExceptionHandle {
    public static void main(String[] args) throws ClassNotFoundException {
        String url = "jdbc:mysql://localhost:3306/myd";
        String user = "root";
        String password = "798167";
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("Connection created");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM emp WHERE salary > 40000");
            System.out.println("\nEmployees");
            while (rs.next()) {
                System.out.println(rs.getInt("id") + " | " +
                                  rs.getString("name") + " | " +
                                  rs.getInt("salary"));
            }
            con.close();
        } catch (SQLException e) {
            System.out.println("Database not found");
        }
    }
}
```

Output : Database not found

14. Create a login system using JDBC where user credentials are verified from the database.

Program :

```
package JDBC_conn;
import java.sql.*;
import java.util.*;
```

```
public class Login {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter username: ");
        String username = sc.nextLine();
        System.out.print("Enter password: ");
        String password = sc.nextLine();
        try {
            // 1. Load and register JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");
            // 2. Connect to DB
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root", "798167");
            // 3. Create PreparedStatement
            String query = "select * from users where username = ? and password = ?";
            PreparedStatement pstmt = con.prepareStatement(query);
            pstmt.setString(1, username);
            pstmt.setString(2, password);
            // 4. Execute query
            ResultSet rs = pstmt.executeQuery();
            if (rs.next()) {
                System.out.println("Login Successful!");
            } else {
                System.out.println("Invalid username or password.");
            }
            // 5. Close connections
            rs.close();
            pstmt.close();
            con.close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        sc.close();
    }
}
```

```
Ouput  : Enter username: admin
Enter password: admin123
Login Successful!
```

15. Implement a Java application to take dynamic input from the user and perform insertion, search, or update using menu-driven logic.

Program :

```
package JDBC_conn;
import java.sql.*;
import java.util.Scanner;
public class Test1 {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        Scanner sc = new Scanner(System.in);
        Connection con = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, password);
            int choice;
            do {
                System.out.println("\nMENU ”);
                System.out.println("1. Insert Student");
                System.out.println("2. Search Student by ID");
                System.out.println("3. Update Student Percentage");
                System.out.println("4. Exit");
                System.out.print("Enter choice: ");
                choice = sc.nextInt();
                sc.nextLine();
                switch (choice) {
                    case 1:
                        System.out.print("Enter ID: ");
                        int id = sc.nextInt();
                        sc.nextLine();
                        System.out.print("Enter Name: ");
                        String name = sc.nextLine();
                        System.out.print("Enter Percentage: ");
                        float perc = sc.nextFloat();

                        String insertSQL = "INSERT INTO students (id, name, percentage) VALUES (?, ?, ?)";
                        try (PreparedStatement pstmt = con.prepareStatement(insertSQL)) {
                            pstmt.setInt(1, id);
                            pstmt.setString(2, name);
                            pstmt.setFloat(3, perc);
                            int rows = pstmt.executeUpdate();
                            System.out.println(rows + " record(s) inserted.");
                        }
                        break;

                    case 2:
                        System.out.print("Enter Student ID to search: ");
                        int searchId = sc.nextInt();
                        String searchSQL = "SELECT * FROM students WHERE id = ?";
                        try (PreparedStatement pstmt = con.prepareStatement(searchSQL)) {
                            pstmt.setInt(1, searchId);
                            ResultSet rs = pstmt.executeQuery();
                            if (rs.next()) {
                                System.out.println("ID: " + rs.getInt("id"));
                                System.out.println("Name: " + rs.getString("name"));
                                System.out.println("Percentage: " + rs.getFloat("percentage"));
                            } else {
                                System.out.println("Student not found.");
                            }
                        }
                        break;

                    case 3:
                        System.out.print("Enter Student ID to update: ");
                        int updateId = sc.nextInt();
                        System.out.print("Enter new Percentage: ");
                        float newPerc = sc.nextFloat();

                        String updateSQL = "UPDATE students SET percentage = ? WHERE id = ?";
                        try (PreparedStatement pstmt = con.prepareStatement(updateSQL)) {
                            pstmt.setFloat(1, newPerc);
                            pstmt.setInt(2, updateId);
                            int rowsUpdated = pstmt.executeUpdate();
                            System.out.println(rowsUpdated + " record(s) updated.");
                        }
                        break;

                    case 4:
                        System.out.println("Exiting program...");
                        break;

                    default:
                        System.out.println("Invalid choice! Try again.");
```

```
                }
            } while (choice != 4);

        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}
```

```
Output :  MENU
1. Insert Student
2. Search Student by ID
3. Update Student Percentage
4. Exit
Enter choice: 2
Enter Student ID to search: 2
ID: 2
Name: Yoga
Percentage: 82.3
MENU
1. Insert Student
2. Search Student by ID
3. Update Student Percentage
4. Exit
Enter choice : 4
Exiting Program…
```

16. Design the schema for a **Library Management System** and write JDBC programs for:

Adding a book, Viewing all books, Issuing a book to a member, Returning a book

Program :

```
package JDBC_conn;
import java.sql.*;
import java.util.Scanner;
public class Library {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";
        Scanner sc = new Scanner(System.in);
        Connection con = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, password);

            int choice;
            do {
                System.out.println("\nLIBRARY MENU");
                System.out.println("1. Add Book");
                System.out.println("2. View All Books");
                System.out.println("3. Issue Book");
                System.out.println("4. Return Book");
                System.out.println("5. Exit");
                System.out.print("Enter choice: ");
                choice = sc.nextInt();
                sc.nextLine();

                switch (choice) {
                    case 1:
                        System.out.print("Enter Book ID: ");
                        int id = sc.nextInt();
                        sc.nextLine();

                        System.out.print("Enter Title: ");
                        String title = sc.nextLine();

                        System.out.print("Enter Author: ");
                        String author = sc.nextLine();

                        String insertSQL = "INSERT INTO books (book_id, title, author, available) VALUES (?, ?, ?, TRUE)";
                        try (PreparedStatement pstmt = con.prepareStatement(insertSQL)) {
                            pstmt.setInt(1, id);
                            pstmt.setString(2, title);
                            pstmt.setString(3, author);
                            int rows = pstmt.executeUpdate();
                            System.out.println(rows + " book(s) added.");
                        }
                        break;

                    case 2:
                        String viewSQL = "SELECT * FROM books";
                        try (Statement stmt = con.createStatement();
                             ResultSet rs = stmt.executeQuery(viewSQL)) {
                            System.out.println("\n--- Book List ---");
                            while (rs.next()) {
                                System.out.printf("%d | %s | %s | %s%n",
                                        rs.getInt("book_id"),
                                        rs.getString("title"),
                                        rs.getString("author"),
                                        rs.getBoolean("available") ? "Available" : "Issued");
                            }
                        }
                        break;
```

```
case 3:
    System.out.print("Enter Book ID to issue: ");
    int issueId = sc.nextInt();

    String checkSQL = "SELECT available FROM books WHERE book_id = ?";
    try (PreparedStatement pstmt = con.prepareStatement(checkSQL)) {
        pstmt.setInt(1, issueId);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            if (rs.getBoolean("available")) {
                String issueSQL = "UPDATE books SET available = FALSE WHERE book_id = ?";
                try (PreparedStatement upstmt = con.prepareStatement(issueSQL)) {
                    upstmt.setInt(1, issueId);
                    upstmt.executeUpdate();
                    System.out.println("Book issued successfully.");
                }
            } else {
                System.out.println("Book is already issued.");
            }
        } else {
            System.out.println("Book not found.");
        }
    }
    break;

case 4:
    System.out.print("Enter Book ID to return: ");
    int returnId = sc.nextInt();

    String returnSQL = "UPDATE books SET available = TRUE WHERE book_id = ?";
    try (PreparedStatement pstmt = con.prepareStatement(returnSQL)) {
```

```
                            pstmt.setInt(1, returnId);
                            int updated = pstmt.executeUpdate();
                            if (updated > 0) {
                                System.out.println("Book returned successfully.");
                            } else {
                                System.out.println("Book not found.");
                            }
                        }
                        break;

                    case 5:
                        System.out.println("Exiting Library System...");
                        break;

                    default:
                        System.out.println("Invalid choice! Try again.");
                }
            } while (choice != 5);

        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}
```

Output :

```
LIBRARY MENU
1. Add Book
2. View All Books
3. Issue Book
4. Return Book
```

```
5. Exit
Enter choice: 2

--- Book List ---
1 | Java Programming | James Gosling | Available
2 | Effective Java | Joshua Bloch | Available
3 | Clean Code | Robert C. Martin | Available

LIBRARY MENU
1. Add Book
2. View All Books
3. Issue Book
4. Return Book
5. Exit
Enter choice: 5
Exiting Library System...
```

17. Create a **Hospital Management System** database. Using JDBC, implement:

Register new patient, Assign doctor, Generate billing

Program :

```
package JDBC_conn;

import java.sql.*;

public class HospitalManagement {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";

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

```
Class.forName("com.mysql.cj.jdbc.Driver");

String insertPatient = "INSERT INTO patient VALUES (?, ?, ?, ?)";
try (PreparedStatement ps = con.prepareStatement(insertPatient)) {
    ps.setInt(1, 1);
    ps.setString(2, "Yoga");
    ps.setInt(3, 30);
    ps.setString(4, "Flu");
    ps.executeUpdate();
    System.out.println("Patient Registered");
}

String insertDoctor = "INSERT INTO doctor VALUES (?, ?, ?)";
try (PreparedStatement ps = con.prepareStatement(insertDoctor)) {
    ps.setInt(1, 101);
    ps.setString(2, "Dr. Smith");
    ps.setString(3, "General Physician");
    ps.executeUpdate();
    System.out.println("Doctor Assigned");
}

String insertBill = "INSERT INTO billing VALUES (?, ?, ?)";
try (PreparedStatement ps = con.prepareStatement(insertBill)) {
    ps.setInt(1, 1001);
    ps.setInt(2, 1);
    ps.setDouble(3, 500.00);
    ps.executeUpdate();
    System.out.println("Bill Generated");
}

System.out.println("\nPatient Records:");
```

```
            try (Statement st = con.createStatement();
                 ResultSet rs = st.executeQuery("SELECT * FROM patient")) {
                while (rs.next()) {
                    System.out.println(rs.getInt("patient_id") + " | " +
                                       rs.getString("name") + " | " +
                                       rs.getInt("age") + " | " +
                                       rs.getString("disease"));
                }
            }

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

```
Output : Patient Registered
Doctor Assigned
Bill Generated

Patient Records:
1 | Yoga | 30 | Flu
```

18. Write a JDBC-based report generator that exports data from a MySQL table to a text or CSV file.

Program :

```
package JDBC_conn;
import java.sql.*;
import java.io.FileWriter;
public class ReportGenerator {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "798167";

        try (Connection con = DriverManager.getConnection(url, user, password)) {
            Class.forName("com.mysql.cj.jdbc.Driver");

            String query = "SELECT * FROM emp";
            try (Statement st = con.createStatement();
                 ResultSet rs = st.executeQuery(query);
                 FileWriter fw = new FileWriter("emp_report.csv")) {

                fw.append("ID,Name,Salary\n");

                while (rs.next()) {
                    fw.append(rs.getInt("id") + ",")
                      .append(rs.getString("name") + ",")
                      .append(rs.getInt("salary") + "\n");
                }
                System.out.println("done");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output : done