Java Student Management System (MVC with JDBC)

1. Student.java (Model)

```java
public class Student {
    private int studentID;
    private String name;
    private String department;
    private double marks;

    public Student(int studentID, String name, String department, double marks) {
        this.studentID = studentID;
        this.name = name;
        this.department = department;
        this.marks = marks;
    }

    public int getStudentID() { return studentID; }
    public String getName() { return name; }
    public String getDepartment() { return department; }
    public double getMarks() { return marks; }

    public void setName(String name) { this.name = name; }
    public void setDepartment(String department) { this.department = department; }
    public void setMarks(double marks) { this.marks = marks; }
}
```

2. StudentController.java (Controller)

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

public class StudentController {
    private static final String DB_URL = "jdbc:sqlite:students.db";

    public StudentController() {
        createTable();
    }

    private void createTable() {
        try (Connection conn = DriverManager.getConnection(DB_URL);
             Statement stmt = conn.createStatement()) {
            String sql = "CREATE TABLE IF NOT EXISTS Student (" +
                         "StudentID INTEGER PRIMARY KEY," +
                         "Name TEXT NOT NULL," +
                         "Department TEXT NOT NULL," +
                         "Marks REAL NOT NULL)";
            stmt.execute(sql);
        } catch (SQLException e) {
            System.out.println("Table Error: " + e.getMessage());
        }
    }
```

```
public void addStudent(Student student) {
    String sql = "INSERT INTO Student VALUES (?, ?, ?, ?)";
    try (Connection conn = DriverManager.getConnection(DB_URL);
         PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setInt(1, student.getStudentID());
        pstmt.setString(2, student.getName());
        pstmt.setString(3, student.getDepartment());
        pstmt.setDouble(4, student.getMarks());
        pstmt.executeUpdate();
        System.out.println("Student added successfully.");
    } catch (SQLException e) {
        System.out.println("Add Error: " + e.getMessage());
    }
}

public List<Student> getAllStudents() {
    List<Student> list = new ArrayList<>();
    String sql = "SELECT * FROM Student";
    try (Connection conn = DriverManager.getConnection(DB_URL);
         Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery(sql)) {
        while (rs.next()) {
            Student s = new Student(
                rs.getInt("StudentID"),
                rs.getString("Name"),
                rs.getString("Department"),
                rs.getDouble("Marks")
            );
            list.add(s);
        }
    } catch (SQLException e) {
        System.out.println("Fetch Error: " + e.getMessage());
    }
    return list;
}

public void updateStudent(Student student) {
    String sql = "UPDATE Student SET Name=?, Department=?, Marks=? WHERE StudentID=?";
    try (Connection conn = DriverManager.getConnection(DB_URL);
         PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setString(1, student.getName());
        pstmt.setString(2, student.getDepartment());
        pstmt.setDouble(3, student.getMarks());
        pstmt.setInt(4, student.getStudentID());
        pstmt.executeUpdate();
        System.out.println("Student updated successfully.");
    } catch (SQLException e) {
        System.out.println("Update Error: " + e.getMessage());
    }
    }

    public void deleteStudent(int studentID) {
        String sql = "DELETE FROM Student WHERE StudentID=?";
        try (Connection conn = DriverManager.getConnection(DB_URL);
             PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setInt(1, studentID);
            pstmt.executeUpdate();
            System.out.println("Student deleted successfully.");
        } catch (SQLException e) {
            System.out.println("Delete Error: " + e.getMessage());
        }
    }
}
```

3. Main.java (View)

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StudentController controller = new StudentController();

        while (true) {
            System.out.println("\n--- Student Management ---");
            System.out.println("1. Add Student");
            System.out.println("2. View All Students");
            System.out.println("3. Update Student");
            System.out.println("4. Delete Student");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int 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 Department: ");
                    String dept = sc.nextLine();
                    System.out.print("Enter Marks: ");
                    double marks = sc.nextDouble();
                    controller.addStudent(new Student(id, name, dept, marks));
                }
                case 2 -> {
                    List<Student> students = controller.getAllStudents();
                    for (Student s : students) {
                        System.out.printf("%d | %s | %s | %.2f\n", s.getStudentID(), s.getName(), 
s.getDepartment(), s.getMarks());
                    }
                }
                case 3 -> {
                    System.out.print("Enter ID to update: ");
                    int id = sc.nextInt();
                    sc.nextLine();
                    System.out.print("Enter new Name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter new Department: ");
                    String dept = sc.nextLine();
                    System.out.print("Enter new Marks: ");
                    double marks = sc.nextDouble();
                    controller.updateStudent(new Student(id, name, dept, marks));
                }
                case 4 -> {
                    System.out.print("Enter ID to delete: ");
                    int id = sc.nextInt();
                    controller.deleteStudent(id);
                }
                case 5 -> {
                    System.out.println("Exiting...");
                    sc.close();
                    return;
                }
                default -> System.out.println("Invalid choice.");
            }
        }
    }
}
```