

Develop a Menu driven java application for student information which will create the table with appropriate columns. Menu will contain the options like insert, update and delete. Based on the option, data will be inserted or updated or deleted from table based on student id (student_id will be primary key). Display appropriate message for each operation.

```
package ga.test;

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

public class JdbcApp {

    public static void main(String[] args) throws ClassNotFoundException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root",
"Gaurav@123");

            Scanner scanner = new Scanner(System.in);

            createTable(conn);

            while (true) {
                System.out.println("\nMenu:");
                System.out.println("1. Insert Student");
                System.out.println("2. Update Student");
                System.out.println("3. Delete Student");
                System.out.println("4. Exit");
                System.out.print("Enter your choice: ");
                int choice = scanner.nextInt();
```

```

switch (choice) {
    case 1:
        insertStudent(conn, scanner);
        break;
    case 2:
        updateStudent(conn, scanner);
        break;
    case 3:
        deleteStudent(conn, scanner);
        break;
    case 4:
        System.out.println("Exiting application. Goodbye!");
        return;
    default:
        System.out.println("Invalid choice. Please try again.");
}
}

} catch (SQLException e) {
    e.printStackTrace();
}

}

private static void createTable(Connection conn) throws SQLException {
    String createTableSQL = "CREATE TABLE IF NOT EXISTS students (" +

```

```

        "student_id INT PRIMARY KEY, " +
        "name VARCHAR(100), " +
        "department VARCHAR(100))";
try (Statement stmt = conn.createStatement()) {
    stmt.execute(createTableSQL);
    System.out.println("Table 'students' is ready.");
}
}

```

```

private static void insertStudent(Connection conn, Scanner scanner) {
    System.out.print("Enter Student ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Department: ");
    String department = scanner.nextLine();
}

```

```

String insertSQL = "INSERT INTO students (student_id, name, department)
VALUES (?, ?, ?)";

```

```

try (PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {
    pstmt.setInt(1, id);
    pstmt.setString(2, name);
    pstmt.setString(3, department);
    pstmt.executeUpdate();
    System.out.println("Student inserted successfully.");
}

```

```
    } catch (SQLException e) {  
        System.out.println(e);  
    }  
}
```

```
private static void updateStudent(Connection conn, Scanner scanner) {  
    System.out.print("Enter Student ID to update: ");  
    int id = scanner.nextInt();  
    scanner.nextLine();  
    System.out.print("Enter New Name: ");  
    String name = scanner.nextLine();  
    System.out.print("Enter New Department: ");  
    String department = scanner.nextLine();  
  
    String updateSQL = "UPDATE students SET name = ?, department = ?  
WHERE student_id = ?";  
    try (PreparedStatement pstmt = conn.prepareStatement(updateSQL)) {  
        pstmt.setString(1, name);  
        pstmt.setString(2, department);  
        pstmt.setInt(3, id);  
        int rowsUpdated = pstmt.executeUpdate();  
        if (rowsUpdated > 0) {  
            System.out.println("Student updated successfully.");  
        } else {  
            System.out.println("Student ID not found.");  
        }  
    } catch (SQLException e) {
```

```

        System.out.println(e);
    }
}

private static void deleteStudent(Connection conn, Scanner scanner) {
    System.out.print("Enter Student ID to delete: ");
    int id = scanner.nextInt();
    String deleteSQL = "DELETE FROM students WHERE student_id = ?";
    try (PreparedStatement pstmt = conn.prepareStatement(deleteSQL)) {
        pstmt.setInt(1, id);
        int rowsDeleted = pstmt.executeUpdate();
        if (rowsDeleted > 0) {
            System.out.println("Student deleted successfully.");
        } else {
            System.out.println("Student ID not found.");
        }
    } catch (SQLException e) {
        System.out.println(e);
    }
}
}
}

```