Sql

```
package jdbc_sql;


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


class sql {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Connection con = null;


      try {
         // Step 1: Load driver
         Class.forName("com.mysql.cj.jdbc.Driver");
         // Step 2: Connect to MySQL
         con = DriverManager.getConnection("jdbc:mysql://localhost:3306/sql_db", "root",
"chaitanya@2910");


         Statement st = con.createStatement();
         st.executeUpdate("CREATE TABLE IF NOT EXISTS student(roll INT PRIMARY KEY, name
VARCHAR(30), branch VARCHAR(20))");


         int choice;
         do {
            System.out.println("\n===== STUDENT DATABASE MENU =====");
            System.out.println("1. Insert Record");
            System.out.println("2. Display Records");
            System.out.println("3. Update Record");
            System.out.println("4. Delete Record");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
```

```
            choice = sc.nextInt();
            sc.nextLine(); // clear buffer

            switch (choice) {
               case 1:
                  System.out.print("Enter Roll No: ");
                  int roll = sc.nextInt();
                  sc.nextLine();
                  System.out.print("Enter Name: ");
                  String name = sc.nextLine();
                  System.out.print("Enter Branch: ");
                  String branch = sc.nextLine();

                  st.executeUpdate("INSERT INTO student VALUES(" + roll + ", '" + name + "', '" + branch +
"')");
                  System.out.println("Record inserted successfully!");
                  break;

               case 2:
                  ResultSet rs = st.executeQuery("SELECT * FROM student");
                  System.out.println("\nROLL\tNAME\tBRANCH");
                  System.out.println("-------------------------");
                  while (rs.next()) {
                     System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3));
                  }
                  break;

               case 3:
                  System.out.print("Enter Roll No to update: ");
                  int upRoll = sc.nextInt();
                  sc.nextLine();
```

```
                System.out.print("Enter New Branch: ");
                String newBranch = sc.nextLine();
                st.executeUpdate("UPDATE student SET branch='" + newBranch + "' WHERE roll=" +
upRoll);
                System.out.println("Record updated successfully!");
                break;

            case 4:
                System.out.print("Enter Roll No to delete: ");
                int delRoll = sc.nextInt();
                st.executeUpdate("DELETE FROM student WHERE roll=" + delRoll);
                System.out.println("Record deleted successfully!");
                break;

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

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

    } while (choice != 5);

} catch (Exception e) {
    System.out.println("Error: " + e);
} finally {
    try {
        if (con != null) con.close();
    } catch (Exception e) {
        System.out.println("Error closing connection: " + e);
```

```
        }
      }
    }
  }
```

```
package aryan;

//import com.mongodb.*;
import com.mongodb.client.*;
import org.bson.Document;
import static com.mongodb.client.model.Filters.eq;
import static com.mongodb.client.model.Updates.set;


import java.util.*;

public class jdbcnew {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MongoClient mongoClient = MongoClients.create("mongodb://127.0.0.1:27017");
        MongoDatabase db = mongoClient.getDatabase("te31341_db");
        MongoCollection<Document> col = db.getCollection("movies");

        int choice;
        do {
            System.out.println("\n===== MONGODB STUDENT DATABASE MENU =====");
            System.out.println("1. Insert Record");
            System.out.println("2. Display Records");
            System.out.println("3. Update Record");
            System.out.println("4. Delete Record");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            sc.nextLine(); // clear buffer
```

```
switch (choice) {
    case 1:
        System.out.print("Enter Roll No: ");
        int roll = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Branch: ");
        String branch = sc.nextLine();

        Document doc = new Document("roll", roll)
                .append("name", name)
                .append("branch", branch);
        col.insertOne(doc);
        System.out.println("Record inserted successfully!");
        break;

    case 2:
        System.out.println("\nAll Students:");
        for (Document d : col.find()) {
            System.out.println(d.toJson());
        }
        break;

    case 3:
        System.out.print("Enter Roll No to update: ");
        int upRoll = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter New Branch: ");
        String newBranch = sc.nextLine();
```

```
                col.updateOne(eq("roll", upRoll),
                        new Document("$set", new Document("branch", newBranch)));
                System.out.println("Record updated successfully!");
                break;

            case 4:
                System.out.print("Enter Roll No to delete: ");
                int delRoll = sc.nextInt();
                col.deleteOne(eq("roll", delRoll));
                System.out.println("Record deleted successfully!");
                break;

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

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

      } while (choice != 5);

      mongoClient.close();
   }
}
```