

Assign 8

```
package jdbc_sql;

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

public class sql {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        try {

            // Step 1 - Load Driver

            Class.forName("com.mysql.cj.jdbc.Driver"); // use com.mysql.cj for newer versions

            // Step 2 - Establish Connection

            Connection con = DriverManager.getConnection(

                "jdbc:mysql://localhost:3306/sql_db", "root", "chaitanya@2910");

            Statement stmt = con.createStatement();

            // Create table if not exists

            String createTable = "CREATE TABLE IF NOT EXISTS User_info ("

                + "u_id INT UNSIGNED NOT NULL AUTO_INCREMENT, "

                + "name VARCHAR(30), "

                + "email VARCHAR(50), "

                + "role ENUM('admin','user'), "

                + "created_at DATETIME, "

                + "PRIMARY KEY(u_id))";

            stmt.executeUpdate(createTable);

            int choice;

            do {

                System.out.println("\nMENU");
```

```
System.out.println("1. Insert Entry");
System.out.println("2. Read Entries");
System.out.println("3. Update Entry");
System.out.println("4. Delete Entry");
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
choice = sc.nextInt();
sc.nextLine();
```

```
switch (choice) {
    case 1:
        System.out.print("Enter name: ");
        String name = sc.nextLine();
        System.out.print("Enter email: ");
        String email = sc.nextLine();
        System.out.print("Enter role (admin/user): ");
        String role = sc.nextLine();
```

```
String insertSQL = "INSERT INTO User_info(name, email, role, created_at) VALUES (?, ?, ?, NOW())";
```

```
PreparedStatement insertStmt = con.prepareStatement(insertSQL);
insertStmt.setString(1, name);
insertStmt.setString(2, email);
insertStmt.setString(3, role);
insertStmt.executeUpdate();
System.out.println("Entry inserted successfully!");
break;
```

case 2:

```
ResultSet rs = stmt.executeQuery("SELECT * FROM User_info");

System.out.println("\n User Table");

while (rs.next()) {

    System.out.println("ID: " + rs.getInt("u_id")

        + ", Name: " + rs.getString("name")

        + ", Email: " + rs.getString("email")

        + ", Role: " + rs.getString("role")

        + ", Created At: " + rs.getTimestamp("created_at"));

}

break;
```

case 3:

```
System.out.print("Enter user ID to update: ");

int uidUpdate = sc.nextInt();

sc.nextLine();

System.out.print("Enter new email: ");

String newEmail = sc.nextLine();

String updateSQL = "UPDATE User_info SET email = ? WHERE u_id = ?";

PreparedStatement updateStmt = con.prepareStatement(updateSQL);

updateStmt.setString(1, newEmail);

updateStmt.setInt(2, uidUpdate);

int updated = updateStmt.executeUpdate();

if (updated > 0)

    System.out.println("Entry updated successfully!");

else

    System.out.println("No entry found with ID " + uidUpdate);

break;
```

```

case 4: // Delete

    System.out.print("Enter user ID to delete: ");

    int uidDelete = sc.nextInt();

    String deleteSQL = "DELETE FROM User_info WHERE u_id = ?";
    PreparedStatement deleteStmt = con.prepareStatement(deleteSQL);
    deleteStmt.setInt(1, uidDelete);
    int deleted = deleteStmt.executeUpdate();
    if (deleted > 0)
        System.out.println("Entry deleted successfully!");
    else
        System.out.println("No entry found with ID " + uidDelete);
    break;

case 5:

    System.out.println("Exiting...");

    break;

default:

    System.out.println("Invalid choice!");

}

} while (choice != 5);

con.close();

} catch (ClassNotFoundException | SQLException e) {

    System.out.println("Error: " + e);

}

}

}

```

```
module jdbc_sql {  
    requires java.sql;  
}
```

Assign 12

```
package aryan;
```

```
import com.mongodb.client.*;
```

```
import org.bson.Document;
```

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

```
import static com.mongodb.client.model.Filters.eq;
```

```
import static com.mongodb.client.model.Updates.set;
```

```
public class jdbcnew {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        // String host = "10.10.8.119";
```

```
        String dbName = "te31341_db";
```

```
        // String username = "te31341";
```

```
        // String password = "te31341";
```

```
        // connection string hai --> mongodb://username:password@host/dbname
```

```
        // String connectionString = "mongodb://" + username + ":" + password + "@" + host + "/" +  
        dbName;
```

```
        MongoClient mongoClient = MongoClient.create("mongodb://127.0.0.1:27017");
```

```
        MongoDBDatabase database = mongoClient.getDatabase(dbName);
```

```
MongoCollection<Document> movies = database.getCollection("movies");
```

```
int choice;
```

```
do {
```

```
    System.out.println("\n=== MongoDB CRUD Menu ===");
```

```
    System.out.println("1. Add Movie");
```

```
    System.out.println("2. View All Movies");
```

```
    System.out.println("3. Search Movie by Title");
```

```
    System.out.println("4. Update Movie Rating");
```

```
    System.out.println("5. Delete Movie by Title");
```

```
    System.out.println("6. Exit");
```

```
    System.out.print("Enter your choice: ");
```

```
    choice = sc.nextInt();
```

```
    sc.nextLine();
```

```
    switch (choice) {
```

```
        case 1:
```

```
            System.out.print("Enter title: ");
```

```
            String title = sc.nextLine();
```

```
            System.out.print("Enter director: ");
```

```
            String director = sc.nextLine();
```

```
            System.out.print("Enter release year: ");
```

```
            int year = sc.nextInt();
```

```
            sc.nextLine();
```

```
            System.out.print("Enter genres (comma separated): ");
```

```
            List<String> genres = Arrays.asList(sc.nextLine().split(","));
```

```
            System.out.print("Enter rating: ");
```

```
            double rating = sc.nextDouble();
```

```
            sc.nextLine();
```

```
            Document movie = new Document("title", title)
```

```
.append("director", director)
.append("releaseYear", year)
.append("genre", genres)
.append("rating", rating);
```

```
movies.insertOne(movie);
System.out.println("Movie added successfully.");
break;
```

case 2:

```
System.out.println("Movie List:");
for (Document doc : movies.find()) {
    System.out.println(doc.toJson());
}
break;
```

case 3:

```
System.out.print("Enter title to search: ");
String searchTitle = sc.nextLine();
Document found = movies.find(eq("title", searchTitle)).first();
if (found != null)
    System.out.println("Movie found: " + found.toJson());
else
    System.out.println("No movie found with that title.");
break;
```

case 4:

```
System.out.print("Enter title to update rating: ");
String updateTitle = sc.nextLine();
System.out.print("Enter new rating: ");
double newRating = sc.nextDouble();
```



```

        sc.nextLine();
//        Bson update = ;
        movies.updateMany(eq("title", updateTitle),set("rating", newRating));
        System.out.println("Movie rating updated successfully.");
        break;

    case 5:
        System.out.print("Enter title to delete: ");
        String deleteTitle = sc.nextLine();
        movies.deleteOne(eq("title", deleteTitle));
        System.out.println("Movie deleted successfully.");
        break;

//        break;

    case 6:
        System.out.println("Exiting...");
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
    }
} while (choice != 6);

mongoClient.close();
sc.close();
}
}
//
//
//
//// Testcase :
```

```
//
//// 1.Insert:
//
//// === MongoDB CRUD Menu ===
//// 1. Add Movie
//// 2. View All Movies
//// 3. Search Movie by Title
//// 4. Update Movie Rating
//// 5. Delete Movie by Title
//// 6. Exit
//// Enter your choice:
//// 1
//// Enter title: Lagan
//// Enter director: keshav
//// Enter release year: 1999
//// Enter genres (comma separated): comedy,sports
//// Enter rating: 8
//
//// output :
//
//// [
////   _id: ObjectId('69019a1a29678b40104df76a'),
////   title: 'Lagan',
////   director: 'keshav',
////   releaseYear: 1999,
////   genre: [ 'comedy', 'sports' ],
////   rating: 8
//// ]
```