Relational Databases with MySQL Week 10 Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your Java project code to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

In this week's coding activity, you will create a menu driven application backed by a MySQL database.

To start, choose one item that you like. It could be vehicles, sports, foods, etc....

Create a new Java project in Eclipse.

Create a SQL script in the project to create a database with one table. The table should be the item you picked.

Write a Java menu driven application that allows you to perform all four CRUD operations on your table.

Tips:

The application does not need to be as complex as the example in the video curriculum.

You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

Remember that both parameters on PreparedStatements and the ResultSet columns are based on indexes that start with 1, not 0.

Screenshots of Code:

```
1  package application;
2
3  public class Application {
4
5      public static void main(String[] args) {
6          Menu menu = new Menu();
7          menu.Start();
8      }
9  }
```

```
1  package application;
2
3  import java.sql.SQLException;
10
11 public class Menu {
12
13     private AlbumDao albumDao = new AlbumDao();
14     Scanner scanner = new Scanner(System.in);
15     private List<String> selections = Arrays.asList( "Quit Menu", "Display Albums", "Create Album", "Update Album", "Delete Album");
16     private List<String> updateSelections = Arrays.asList("Name","Artist","Record Date", "Release Date");
17
18     public void Start() {
19         String selection;
20         displayMenu();
21         do {
22             selection = scanner.nextLine();
23             try {
24                 switch(selection) {
25                 case "0":
26                     System.out.println("Menu Closed.");
27                     break;
28                 case "1":
29                     displayAlbums();
30                     break;
31                 case "2":
32                     createAlbum();
33                     break;
34                 case "3":
35                     updateAlbum();
36                     break;
37                 case "4":
38                     deleteAlbum();
39                 }
40             }catch(SQLException e) {
41                 e.printStackTrace();
42             }
43         }while(!selection.equals("0"));
44     }
45
46     private void displayMenu() {
47         System.out.println("Please choose a selection:");
48         for(int i = 0; i < selections.size(); i++) {
49             System.out.println(i + ") " + selections.get(i));
50         }
```

```
        System.out.println("Press Enter After Selection...");
    }

    private String correct(String oldString) {
        String space = "                    ";
        String newString = "||";
        newString += oldString;
        newString += space;
        return newString.substring(0,20);
    }

    private String correct(int integer) {
        String oldInt = Integer.toString(integer);
        String space = "                    ";
        String newInt = "||";
        newInt += oldInt;
        newInt += space;
        return newInt.substring(0,20);
    }

    private void displayAlbums() throws SQLException {
        List<Album> albums = albumDao.getAlbums();
        System.out.println("||Id----------||Name----------||Artist----------||Record Date-------||Release Date");
        for(Album album : albums) {
            System.out.println(correct(album.getAlbumId()) + correct(album.getName()) + correct(album.getArtist()) + correct(album.getRecordDate()) + correct(album.getReleaseDate()));
        }
    }

    private void createAlbum() throws SQLException {
        System.out.print("Enter New Album Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Artist Name For New Album:");
        String artist = scanner.nextLine();
        System.out.print("Enter Record Date For New Album (YYYY-MM-DD):");
        String recordDate = scanner.nextLine();
        System.out.print("Enter Release Date For New Album (YYYY-MM-DD): ");
        String releaseDate = scanner.nextLine();
        albumDao.createAlbum(name, artist, recordDate, releaseDate);
        System.out.println("New Album Created!");
    }

    private void deleteAlbum() throws SQLException {
        System.out.print("Enter Album Id to Delete:");
        int id = Integer.parseInt(scanner.nextLine());
        albumDao.deleteAlbum(id);
        System.out.println("Album with Id '" + id + "' Deleted!");
    }

    private void updateAlbum() throws SQLException{
        System.out.print("Enter The Id of Album to Update: ");
        int id = Integer.parseInt(scanner.nextLine());
        System.out.println("Select a Value to Change:");
        for(int i = 0; i < updateSelections.size(); i++) {
            System.out.println((i + 1) + ") " + updateSelections.get(i));
        }
        String valueToChange = updateSelection(id, scanner.nextLine());
        System.out.println("Album " + id + "'s " + valueToChange.substring(0, 1).toUpperCase() + valueToChange.substring(1) + " Updated!");
    }

    private  String updateSelection(int id, String valueToChange) throws SQLException {
        String choice = "";
        String newValue = "";
        switch(valueToChange) {
        case "1":
            choice = "name";
            System.out.print("Enter New Value For Name: ");
            newValue = scanner.nextLine();
            albumDao.updateAlbumName(id, newValue);
            break;

        case "2":
            choice = "artist";
            System.out.print("Enter New Value For Artist: ");
            newValue = scanner.nextLine();
            albumDao.updateAlbumArtist(id, newValue);
            break;

        case "3":
            choice = "recordDate";
            System.out.print("Enter New Value For Record Date: ");
            newValue = scanner.nextLine();
            albumDao.updateAlbumRecordDate(id, newValue);
            break;

        case "4":
            choice = "releaseDate";
            System.out.print("Enter New Value For Release Date: ");
            newValue = scanner.nextLine();
                albumDao.updateAlbumReleaseDate(id, newValue);
                break;
            }

            return choice;
        }
}
```

```java
package entities;

public class Album {
    private Integer albumId;
    private String name;
    private String artist;
    private String recordDate;
    private String releaseDate;

    public Album(Integer albumId, String name, String artist, String recordDate, String releaseDate) {
        this.setAlbumId(albumId);
        this.setName(name);
        this.setArtist(artist);
        this.setRecordDate(recordDate);
        this.setReleaseDate(releaseDate);
    }

    public Integer getAlbumId() {
        return albumId;
    }

    public void setAlbumId(Integer id) {
        this.albumId = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getArtist() {
        return artist;
    }

    public void setArtist(String artist) {
        this.artist = artist;
    }

    public String getRecordDate() {
        return recordDate;
    }

    public void setRecordDate(String recordDate) {
        this.recordDate = recordDate;
    }

    public String getReleaseDate() {
        return releaseDate;
    }

    public void setReleaseDate(String releaseDate) {
        this.releaseDate = releaseDate;
    }
}
```

```java
package dao;

import java.sql.Connection;

public class DBConnection {



    private final static String URL = "jdbc:mysql://localhost:3306/week10codingassignment";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "root";
    private static Connection connection;
    private static DBConnection instance;

    private DBConnection(Connection connection) {
        DBConnection.connection = connection;
    }

    public static Connection getConnection() {

        if(instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                instance = new DBConnection(connection);
                System.out.println("Connection Successful");
            }catch(SQLException e) {
                e.printStackTrace();
            }
        }
        return DBConnection.connection;
    }
}
```

```java
package dao;

import java.sql.Connection;

public class AlbumDao {

    private Connection connection = DBConnection.getConnection();
    private final String GET_ALBUMS_QUERY = "SELECT * FROM albums";
    private final String CREATE_ALBUM_QUERY = "INSERT INTO albums(name, artist, recordDate, releaseDate) VALUES (?,?,?,?)" ;
    private final String DELETE_ALBUM_QUERY = "DELETE FROM albums WHERE albumId = ?";
    private final String UPDATE_ALBUM_NAME_QUERY = "UPDATE albums SET name = ? WHERE albumId = ?";
    private final String UPDATE_ALBUM_ARTIST_QUERY = "UPDATE albums SET artist = ? WHERE albumId = ?";
    private final String UPDATE_ALBUM_RECORD_DATE_QUERY = "UPDATE albums SET recordDate = ? WHERE albumId = ?";
    private final String UPDATE_ALBUM_RELEASE_DATE_QUERY = "UPDATE albums SET releaseDate = ? WHERE albumId = ?";

    public AlbumDao() {
        connection = DBConnection.getConnection();
    }

    public List<Album> getAlbums() throws SQLException{

        ResultSet rs = connection.prepareStatement(GET_ALBUMS_QUERY).executeQuery();
        List<Album> albums = new ArrayList<Album>();

        while(rs.next()) {
            albums.add(populateAlbums(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));
        }
        return albums;
    }

    public void createAlbum(String name, String artist, String recordDate, String releaseDate) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_ALBUM_QUERY);
        ps.setString(1,name);
        ps.setString(2, artist);
        ps.setString(3, recordDate);
        ps.setString(4, releaseDate);
        ps.executeUpdate();
    }

    public void deleteAlbum(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_ALBUM_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }

    public void updateAlbumName(int id, String newValue) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_ALBUM_NAME_QUERY);
        ps.setString(1, newValue);
        ps.setInt(2, id);
        ps.executeUpdate();
    }

    public void updateAlbumArtist(int id, String newValue) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_ALBUM_ARTIST_QUERY);
        ps.setString(1, newValue);
        ps.setInt(2, id);
        ps.executeUpdate();
    }

    public void updateAlbumRecordDate(int id, String newValue) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_ALBUM_RECORD_DATE_QUERY);
        ps.setString(1, newValue);
        ps.setInt(2, id);
        ps.executeUpdate();
    }

    public void updateAlbumReleaseDate(int id, String newValue) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_ALBUM_RELEASE_DATE_QUERY);
        ps.setString(1, newValue);
        ps.setInt(2, id);
        ps.executeUpdate();
    }

    private Album populateAlbums(Integer albumId, String name, String artist, String recordDate, String releaseDate) {

        return new Album(albumId, name, artist, recordDate, releaseDate);
    }
}
```

Screenshots of Running Application:

```
Connection Successful
Please choose a selection:
0) Quit Menu
1) Display Albums
2) Create Album
3) Update Album
4) Delete Album
Press Enter After Selection...
1
||Id----------------||Name--------------||Artist------------||Record Date-------||Release Date
||1                 ||Bad               ||Michael Jackson   ||1987-07-01        ||1987-08-31
||2                 ||The Wall          ||Pink Floyd        ||1979-11-01        ||1979-11-30
||3                 ||Chicago           ||Chicago           ||1969-08-01        ||1970-01-26
||4                 ||Time Out          ||Dave Brubeck Quart||1959-08-18        ||1959-12-14
2
Enter New Album Name: Humanz
Enter Artist Name For New Album:Gorillaz
Enter Record Date For New Album (YYYY-MM-DD):2016-12-01
Enter Release Date For New Album (YYYY-MM-DD): 2017-04-28
New Album Created!
3
Enter The Id of Album to Update: 5
Select a Value to Change:
1) Name
2) Artist
3) Record Date
4) Release Date
3
Enter New Value For Record Date: 2016-12-02
Album 5's RecordDate Updated!
1
||Id----------------||Name--------------||Artist------------||Record Date-------||Release Date
||1                 ||Bad               ||Michael Jackson   ||1987-07-01        ||1987-08-31
||2                 ||The Wall          ||Pink Floyd        ||1979-11-01        ||1979-11-30
||3                 ||Chicago           ||Chicago           ||1969-08-01        ||1970-01-26
||4                 ||Time Out          ||Dave Brubeck Quart||1959-08-18        ||1959-12-14
||5                 ||Humanz            ||Gorillaz          ||2016-12-02        ||2017-04-28
4
Enter Album Id to Delete:5
Album with Id '5' Deleted!
1
||Id----------------||Name--------------||Artist------------||Record Date-------||Release Date
||1                 ||Bad               ||Michael Jackson   ||1987-07-01        ||1987-08-31
||2                 ||The Wall          ||Pink Floyd        ||1979-11-01        ||1979-11-30
||3                 ||Chicago           ||Chicago           ||1969-08-01        ||1970-01-26
||4                 ||Time Out          ||Dave Brubeck Quart||1959-08-18        ||1959-12-14
0
Menu Closed.
```

Expanded Update Album Selection:

```
Connection Successful
Please choose a selection:
0) Quit Menu
1) Display Albums
2) Create Album
3) Update Album
4) Delete Album
Press Enter After Selection...
1
||Id----------------||Name--------------||Artist------------||Record Date-------||Release Date
||1                 ||Bad               ||Michael Jackson   ||1987-07-01        ||1987-08-31
||2                 ||The Wall          ||Pink Floyd        ||1979-11-01        ||1979-11-30
||3                 ||Chicago           ||Chicago           ||1969-08-01        ||1970-01-26
||4                 ||Time Out          ||Dave Brubeck Quart||1959-08-18        ||1959-12-14
3
Enter The Id of Album to Update: 1
Select a Value to Change:
1) Name
2) Artist
3) Record Date
4) Release Date
1
Enter New Value For Name: Good
Album 1's Name Updated!
3
Enter The Id of Album to Update: 2
Select a Value to Change:
1) Name
2) Artist
3) Record Date
4) Release Date
2
Enter New Value For Artist: Blue Floyd
Album 2's Artist Updated!
3
Enter The Id of Album to Update: 3
Select a Value to Change:
1) Name
2) Artist
3) Record Date
4) Release Date
4
Enter New Value For Release Date: 1970-01-27
Album 3's ReleaseDate Updated!
1
||Id----------------||Name--------------||Artist------------||Record Date-------||Release Date
||1                 ||Good              ||Michael Jackson   ||1987-07-01        ||1987-08-31
||2                 ||The Wall          ||Blue Floyd        ||1979-11-01        ||1979-11-30
||3                 ||Chicago           ||Chicago           ||1969-08-01        ||1970-01-27
||4                 ||Time Out          ||Dave Brubeck Quart||1959-08-18        ||1959-12-14
```

URL to GitHub Repository:

DaltonCash/PT-WK10 (github.com)