

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:

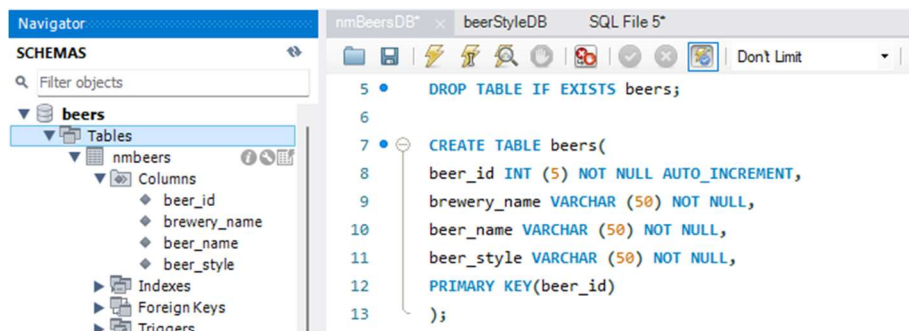
The application does not need to be as complex as the example in the video curriculum.

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

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

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

Screenshots of Code:



nmBeersDB* beerStyleDB SQL File 5* x

1 • SELECT * FROM beers;

2 |

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: I A

beer_id	brewery_name	beer_name	beer_style
1	La Cumbre Brewing Company	Elevated IPA	IPA
NULL	NULL	NULL	NULL

nmBeersDB* beerStyleDB SQL File 5* x

1 • SELECT * FROM beers;

2 |

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: I A

beer_id	brewery_name	beer_name	beer_style
1	La Cumbre Brewing Company	Project Dank	Imperial IPA
NULL	NULL	NULL	NULL

nmBeersDB* beerStyleDB SQL File 5* x

1 • SELECT * FROM beers;

2 | Execute the selected portion of the script or everything, if there is no selection |

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: I A

beer_id	brewery_name	beer_name	beer_style
NULL	NULL	NULL	NULL

beers 54 x Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:11:47	SELECT * FROM beers	1 row(s) returned	0.000 sec / 0.000 sec
2	03:17:51	SELECT * FROM beers	0 row(s) returned	0.000 sec / 0.000 sec
3	03:17:56	SELECT * FROM beers	0 row(s) returned	0.000 sec / 0.000 sec
4	03:18:01	CREATE DATABASE IF NOT EXISTS beers	1 row(s) affected, 1 warning(s): 1007 Can't create database 'beers'; database exists	0.016 sec
5	03:18:01	USE beers	0 row(s) affected	0.000 sec
6	03:18:01	DROP TABLE IF EXISTS beers	0 row(s) affected	0.078 sec
7	03:18:02	CREATE TABLE beers(beer_id INT (5) NOT NULL AUTO_INCREMENT, brewery_name VARCHAR (50) NOT NULL, beer_name VARCHAR (50) ...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.	0.047 sec
8	03:19:56	SELECT * FROM beers	1 row(s) returned	0.000 sec / 0.000 sec
9	03:22:13	SELECT * FROM beers	1 row(s) returned	0.000 sec / 0.000 sec
10	03:26:01	SELECT * FROM beers	0 row(s) returned	0.000 sec / 0.000 sec

```

2
3*import java.sql.SQLException;[]
10 S
11 public class Menu {
12     private BeerDao beerDao = new BeerDao();
13     private Scanner userInput = new Scanner (System.in);
14     private List<String> choices = Arrays.asList("Create A Beer",
15                                                 "Display All Beers",
16                                                 "Update A Beer",
17                                                 "Delete A Beer",
18                                                 "To Exit Press 0");
19
20     public void start(){
21         String userSelection = "";
22
23         do {
24             printBeerMenuOptions();
25             userSelection = userInput.nextLine();
26             try {
27                 if (userSelection.equals("1")){
28                     createABeer();
29                 }else if (userSelection.equals("2")) {
30                     displayAllBeers();
31
32                 }else if (userSelection.equals("3")) {
33                     updateABeer();
34
35                 }else if (userSelection.equals("4")) {
36                     deleteABeer();
37                 }
38             }catch (SQLException e) {
39                 e.printStackTrace();
40             }
41
42             System.out.println("***** Press Enter To Continue *****");
43             userInput.nextLine();
44
45         }while (!userSelection.equals("0"));
46     }
47
48     private void printBeerMenuOptions() {
49         System.out.println("***** Your Guide To Beer ***** \n" + "\n Select An Option Below \n");
50         for(int i =0; i< choices.size(); i++) {
51             System.out.println(i + 1 +") " + choices.get(i));
52         }
53     }
54
55     private void displayAllBeers() throws SQLException {
56         List<Beers> allBeers = BeerDao.getBeers();
57         for(Beers beer: allBeers) {
58             System.out.println("Beer Id: " + beer.getBeer_id() +"\n"
59                             + "Brewery Name:" + beer.getBrewery_name() +"\n"
60                             + "Beer Name:" + beer.getBeer_name()+ "\n"
61                             + "Beer Style:" + beer.getBeer_style());
62         }
63

```

```

63
64
65 }
66 private void createABeer() throws SQLException{
67     System.out.println("Enter Brewery Name");
68     String brewery_name = userInput.nextLine();
69
70     System.out.println("Enter Beer Name");
71     String beer_name = userInput.nextLine();
72
73     System.out.println("Enter Beer Style");
74     String beer_style = userInput.nextLine();
75
76     beerDao.createABeer(brewery_name, beer_name, beer_style);
77
78 }
79
80
81 private void updateABeer() throws SQLException {
82
83     System.out.print("Enter Beer ID To Update");
84     int beer_id = Integer.parseInt(userInput.nextLine());
85
86     System.out.print("Enter New Beer Name ");
87     String beer_name = userInput.nextLine();
88
89     System.out.print("Enter New Beer Style ");
90     String beer_style = userInput.nextLine();
91
92
93     beerDao.updateABeer(beer_id, beer_name, beer_style);
94
95
96 }
97
98
99 private void deleteABeer() throws SQLException {
00     System.out.print("Enter Beer Id To Delete");
01     int beer_id = Integer.parseInt(userInput.nextLine());
02
03     beerDao.deleteABeerById(beer_id);
04
05 }
06 }
07

```

```

Menu.java Application.java BeerDBConnection.java BeerDao.java Beers.java
1 package com.promineotech.application;
2
3 public class Application {
4
5     public static void main(String[] args){
6
7         Menu userMenu = new Menu();
8         userMenu.start();
9
10    }
11
12 }
13

```

```
*Menu.java Application.java BeerDBConnection.java x BeerDao.java Beers.java
1 package com.promineotech.dao;
2
3 import java.sql.Connection;
4
5
6 public class BeerDBConnection {
7     private final static String URL = "jdbc:mysql://localhost:3306/beers";
8     private final static String USERNAME = "root";
9     private final static String PASSWORD = "password";
10    private static Connection connection;
11    private static BeerDBConnection instance;
12
13
14    private BeerDBConnection(Connection connection) {
15        this.connection = connection;
16    }
17
18
19    public static Connection getConnection() {
20        if (instance == null) {
21
22            try {
23                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
24                instance = new BeerDBConnection(connection);
25                System.out.println("The Connection Was Successful");
26
27            } catch (SQLException e) {
28                e.printStackTrace();
29            }
30        }
31        return BeerDBConnection.connection;
32    }
33 }
34
```

```

1 package com.promineotech.dao;
2
3 import java.sql.Connection;
4
5 public class BeerDao {
6
7     private static Connection connection;
8     private final static String GET_ALL_BEERS_QUERY = "SELECT * FROM beers";
9     private final static String CREATE_A_NEW_BEER_QUERY = "INSERT INTO beers (brewery_name, beer_name, beer_style) VALUES(?,?,?)";
10    private final static String UPDATE_A_BEER_QUERY = "UPDATE beers SET beer_name = ?, beer_style= ? WHERE Beer_id = ?";
11    private final static String DELETE_A_BEER_QUERY = "DELETE FROM beers WHERE beer_id = ?";
12
13    public BeerDao() {
14        connection = BeerDBConnection.getConnection();
15    }
16
17    public static List<Beers> getBeers() throws SQLException{
18        Statement statement = connection.createStatement();
19        ResultSet rs = statement.executeQuery(GET_ALL_BEERS_QUERY);
20        List<Beers> beers = new ArrayList<Beers>();
21
22        while(rs.next()) {
23            beers.add(populateBeers(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4)));
24        }
25
26        return beers;
27    }
28
29    public void createABeer(String brewery_name, String beer_name, String beer_style) throws SQLException {
30        PreparedStatement ps = connection.prepareStatement(CREATE_A_NEW_BEER_QUERY);
31        ps.setString(1, brewery_name);
32        ps.setString(2, beer_name);
33        ps.setString(3, beer_style);
34        ps.executeUpdate();
35    }
36
37    public void updateABeer(int beer_id, String beer_name, String beer_style) throws SQLException{
38        PreparedStatement ps = connection.prepareStatement(UPDATE_A_BEER_QUERY);
39        ps.setInt(3, beer_id);
40        ps.setString(2, beer_style);
41        ps.setString(1, beer_name);
42        ps.executeUpdate();
43    }
44
45    public void deleteABeerById(int beer_id) throws SQLException {
46        PreparedStatement ps = connection.prepareStatement(DELETE_A_BEER_QUERY);
47        ps.setInt(1, beer_id);
48        ps.executeUpdate();
49    }
50
51    private static Beers populateBeers(int beer_id, String brewery_name, String beer_name, String beer_style) {
52        return new Beers (beer_id, brewery_name, beer_name, beer_style);
53    }
54 }

```

Screenshots of Running Application:

```
Application (16) [Java Application] C:\Users\horo\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (May 5, 2022, 3:19:11 AM)
The Connection Was Successful
***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
1
Enter Brewery Name
La Cumbre Brewing Company
Enter Beer Name
Elevated IPA
Enter Beer Style
IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Elevated IPA
Beer Style:IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
3
Enter Beer ID To Update1
Enter New Beer Name Project Dank
Enter New Beer Style Imperial IPA
***** Press Enter To Continue *****
```



```
Application (16) [Java Application] C:\Users\haro\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (May 5, 2022, 3:19:11 AM)

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
1
Enter Brewery Name
La Cumbre Brewing Company
Enter Beer Name
Elevated IPA
Enter Beer Style
IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Elevated IPA
Beer Style:IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
3
Enter Beer ID To Update1
Enter New Beer Name Project Dank
Enter New Beer Style Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Project Dank
Beer Style:Imperial IPA
***** Press Enter To Continue *****
```

```
Application [16] [Java Application] C:\Users\haro\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (May 5, 2022, 3:19:11 AM)
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Elevated IPA
Beer Style:IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
3
Enter Beer ID To Update1
Enter New Beer Name Project Dank
Enter New Beer Style Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Project Dank
Beer Style:Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
4
Enter Beer Id To Delete1
***** Press Enter To Continue *****
```

```
Application [16] [Java Application] C:\Users\harof\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (May 5, 2022, 3:19:11 AM)
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Elevated IPA
Beer Style:IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
3
Enter Beer ID To Update1
Enter New Beer Name Project Dank
Enter New Beer Style Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Project Dank
Beer Style:Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
4
Enter Beer Id To Delete1
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
***** Press Enter To Continue *****
```

```
Console x
<terminated> Application (16) [Java Application] C:\Users\harol\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (May 5, 2022, 3:19:11 AM - 3:27:47 AM)
4) Delete A Beer
5) To Exit Press 0
3
Enter Beer ID To Update1
Enter New Beer Name Project Dank
Enter New Beer Style Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
Beer Id: 1
Brewery Name:La Cumbre Brewing Company
Beer Name:Project Dank
Beer Style:Imperial IPA
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
4
Enter Beer Id To Deletel
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
2
***** Press Enter To Continue *****

***** Your Guide To Beer *****

Select An Option Below

1) Create A Beer
2) Display All Beers
3) Update A Beer
4) Delete A Beer
5) To Exit Press 0
0
***** Press Enter To Continue *****
```

URL to GitHub Repository:

<https://github.com/HaroldLujan/Week10CodingAssignment.git>

NOTE: THIS ASSIGNMENT WAS TURNED IN LATE DUE TO A DEATH IN THE FAMILY. I WAS TOLD TO LEAVE A COMMENT BY LISA SMITH ON THIS PAGE.