Relational Databases with MySQL Week 5 Coding 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, with your Java project code, to the 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.

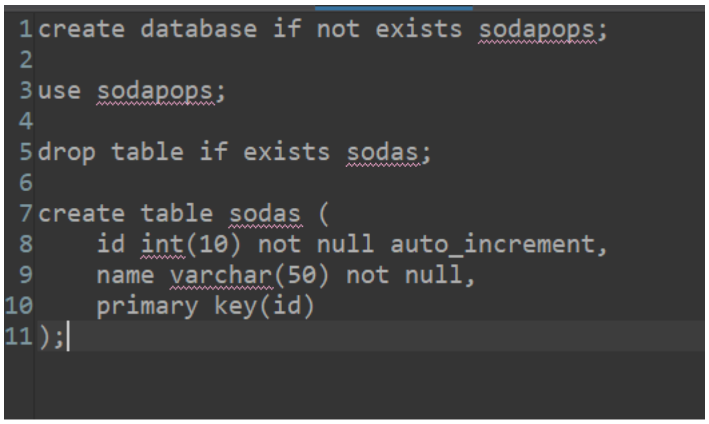
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

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.

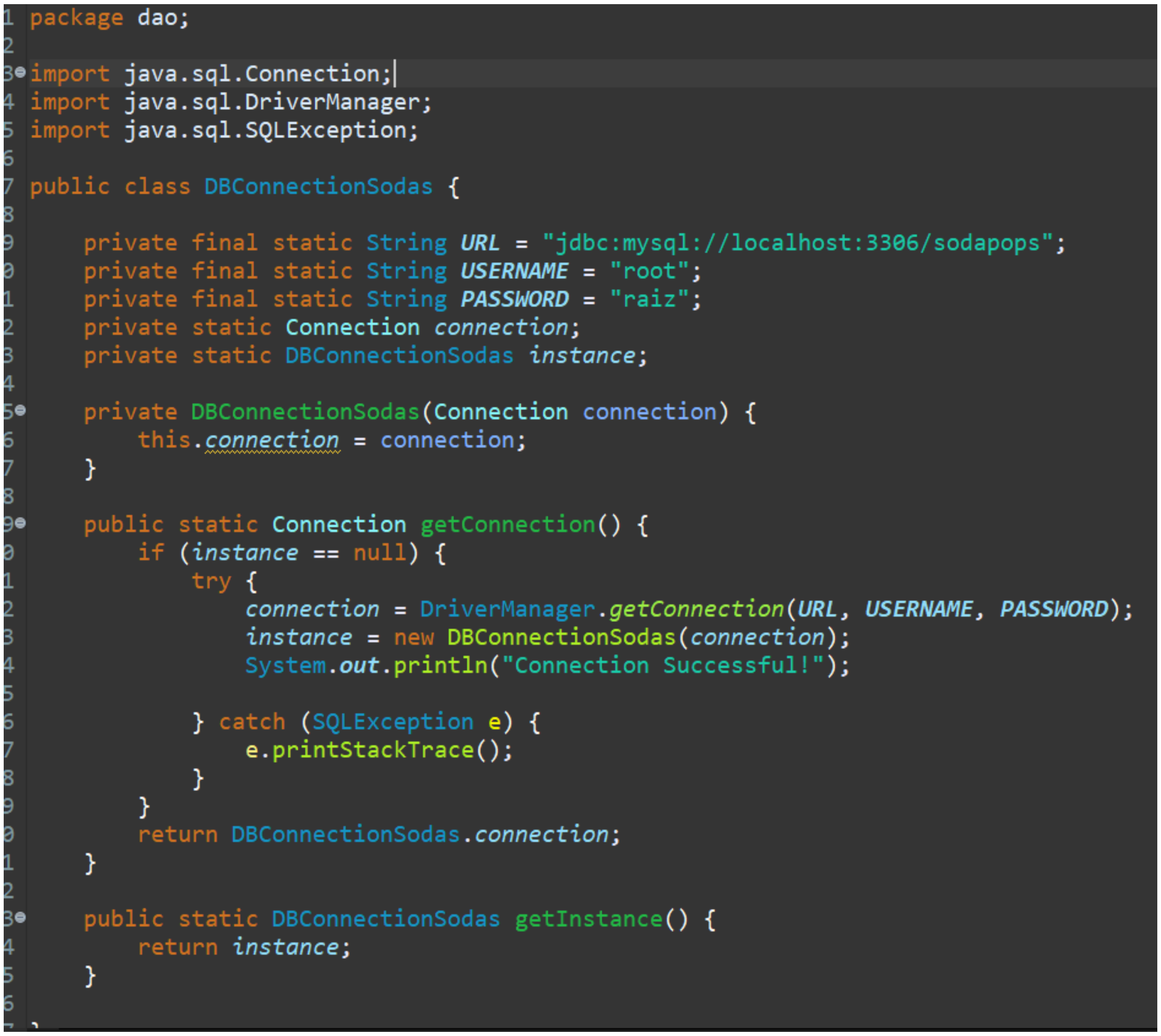
Screenshots of Code:

Script to create the database (my chosen item is varieties of soda/pop):



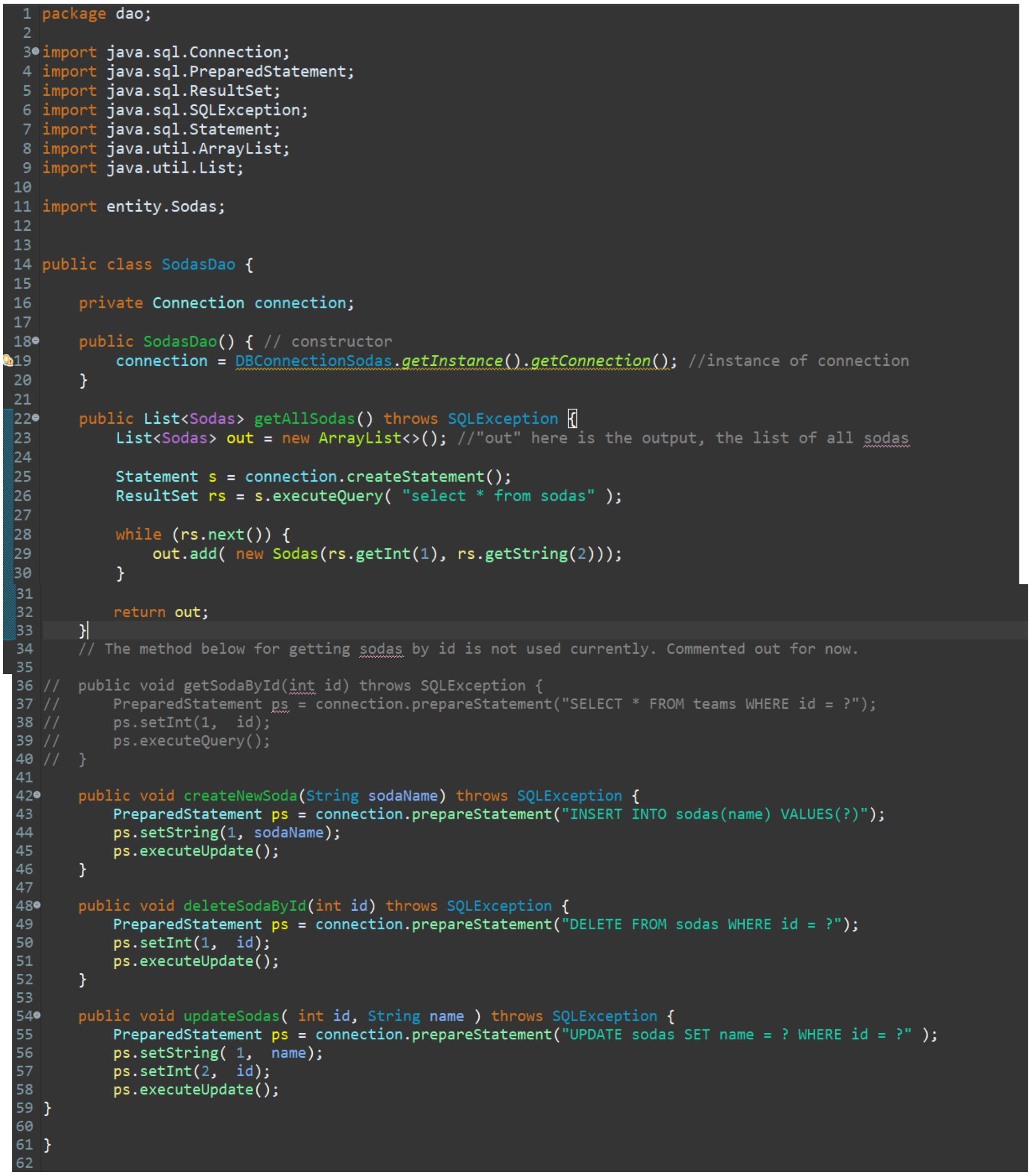
```sql
create database if not exists sodapops;

use sodapops;

drop table if exists sodas;

create table sodas (
    id int(10) not null auto_increment,
    name varchar(50) not null,
    primary key(id)
);
```

Database Connection Class:



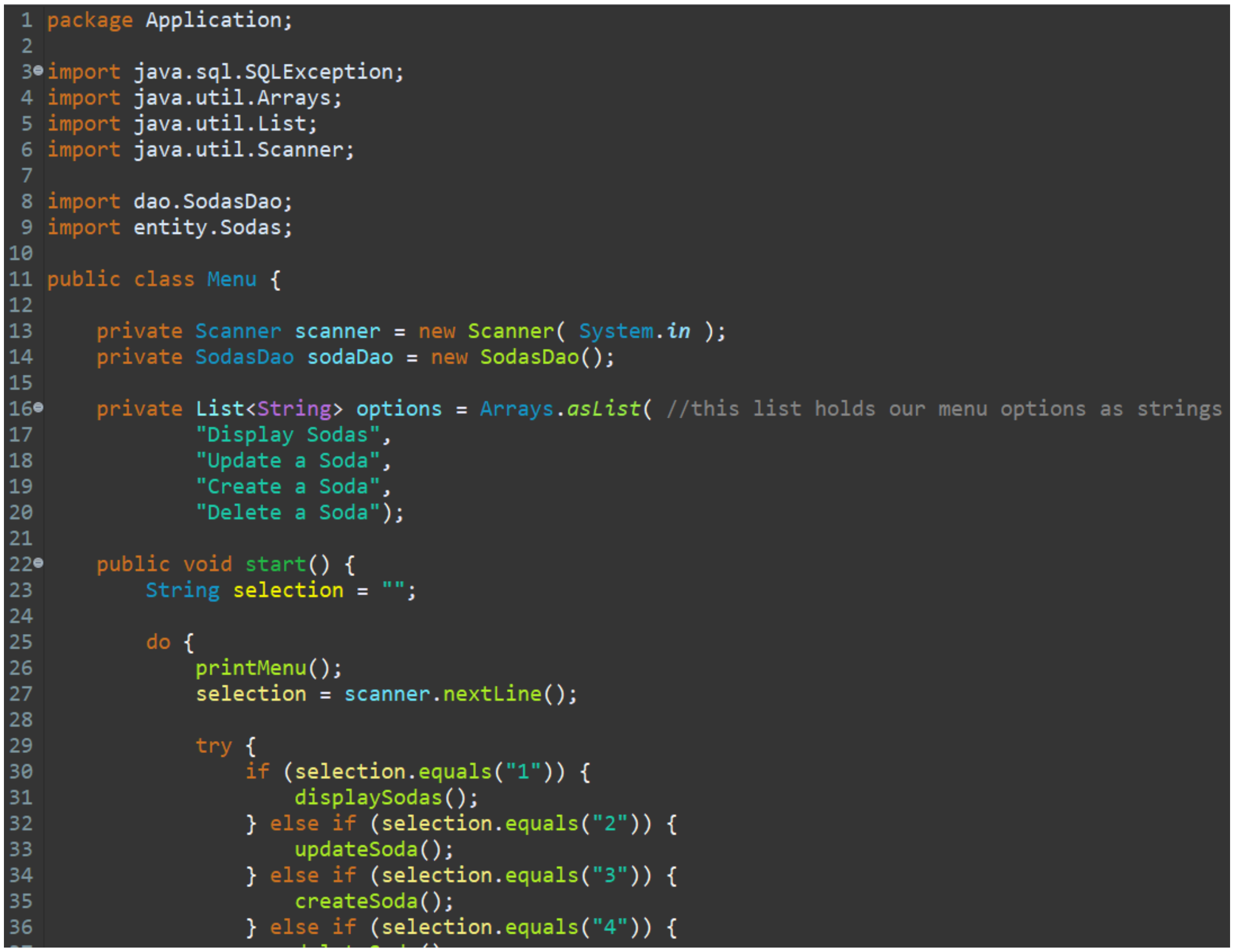
```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnectionSodas {

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

    private DBConnectionSodas(Connection connection) {
        this.connection = connection;
    }

    public static Connection getConnection() {
        if (instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                instance = new DBConnectionSodas(connection);
                System.out.println("Connection Successful!");

            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return DBConnectionSodas.connection;
    }

    public static DBConnectionSodas getInstance() {
        return instance;
    }

```

Database Access Object (DAO)



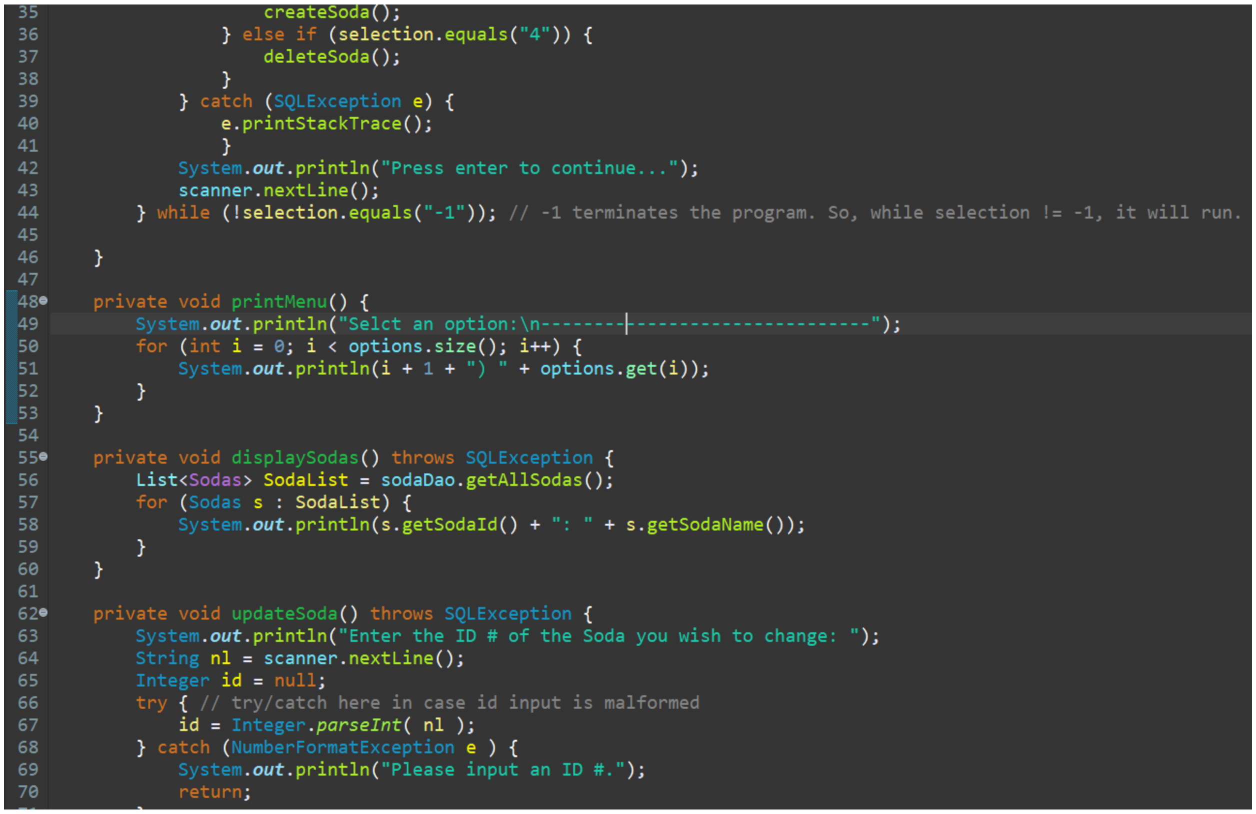
```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import entity.Sodas;


public class SodasDao {

    private Connection connection;

    public SodasDao() { // constructor
        connection = DBConnectionSodas.getInstance().getConnection(); //instance of connection
    }

    public List<Sodas> getAllSodas() throws SQLException {
        List<Sodas> out = new ArrayList<>(); //"out" here is the output, the list of all sodas

        Statement s = connection.createStatement();
        ResultSet rs = s.executeQuery( "select * from sodas" );

        while (rs.next()) {
            out.add( new Sodas(rs.getInt(1), rs.getString(2)));
        }

        return out;
    }
    // The method below for getting sodas by id is not used currently. Commented out for now.

//  public void getSodaById(int id) throws SQLException {
//      PreparedStatement ps = connection.prepareStatement("SELECT * FROM teams WHERE id = ?");
//      ps.setInt(1,  id);
//      ps.executeQuery();
//  }

    public void createNewSoda(String sodaName) throws SQLException {
        PreparedStatement ps = connection.prepareStatement("INSERT INTO sodas(name) VALUES(?)");
        ps.setString(1, sodaName);
        ps.executeUpdate();
    }

    public void deleteSodaById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement("DELETE FROM sodas WHERE id = ?");
        ps.setInt(1,  id);
        ps.executeUpdate();
    }

    public void updateSodas( int id, String name ) throws SQLException {
        PreparedStatement ps = connection.prepareStatement("UPDATE sodas SET name = ? WHERE id = ?" );
        ps.setString( 1,  name);
        ps.setInt(2,  id);
        ps.executeUpdate();
}

}
```

Menu Class:



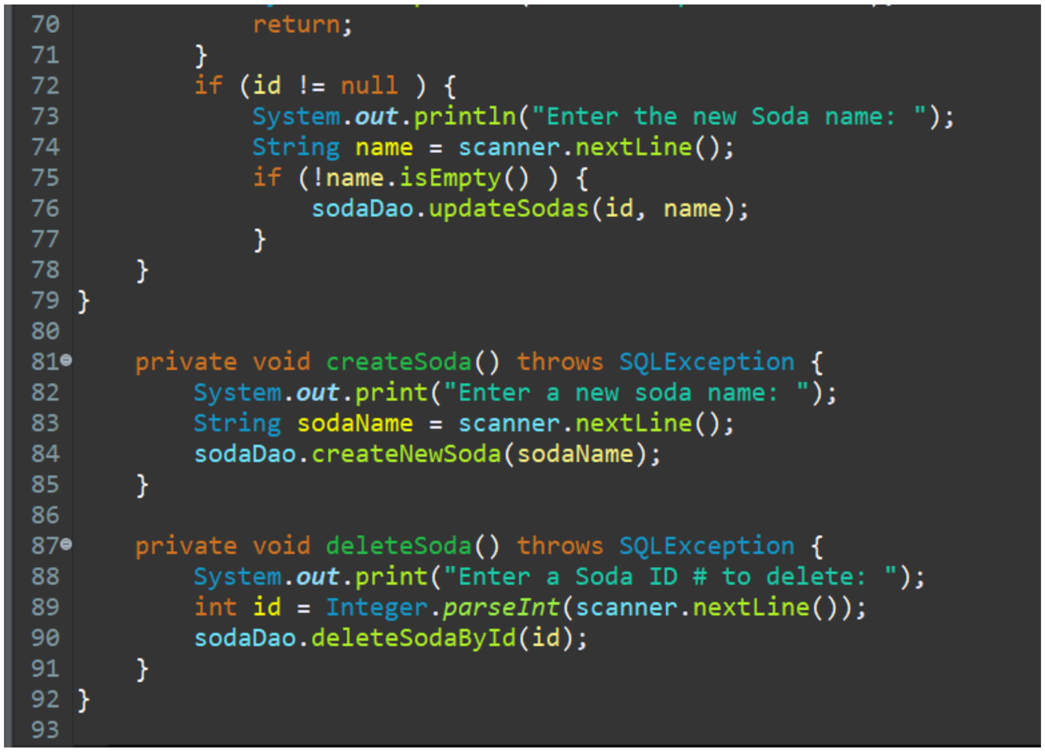
```java
package Application;

import java.sql.SQLException;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

import dao.SodasDao;
import entity.Sodas;

public class Menu {

    private Scanner scanner = new Scanner( System.in );
    private SodasDao sodaDao = new SodasDao();

    private List<String> options = Arrays.asList( //this list holds our menu options as strings
            "Display Sodas",
            "Update a Soda",
            "Create a Soda",
            "Delete a Soda");

    public void start() {
        String selection = "";

        do {
            printMenu();
            selection = scanner.nextLine();

            try {
                if (selection.equals("1")) {
                    displaySodas();
                } else if (selection.equals("2")) {
                    updateSoda();
                } else if (selection.equals("3")) {
                    createSoda();
                } else if (selection.equals("4")) {
```

Menu pt. 2:



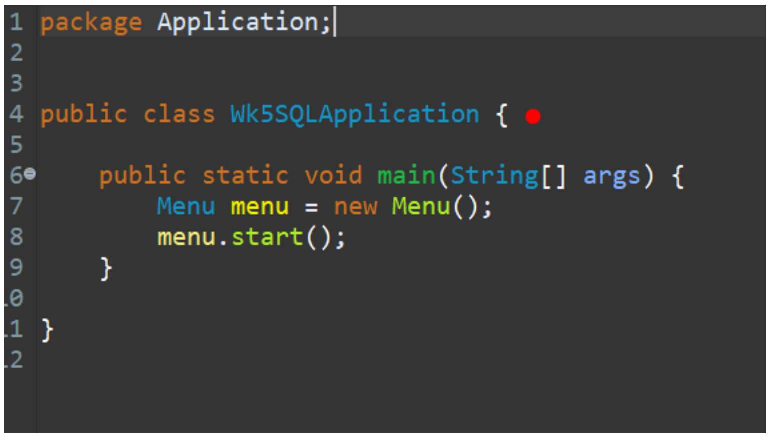
```java
                    createSoda();
                } else if (selection.equals("4")) {
                    deleteSoda();
                }
            } catch (SQLException e) {
                e.printStackTrace();
                }
            System.out.println("Press enter to continue...");
            scanner.nextLine();
        } while (!selection.equals("-1")); // -1 terminates the program. So, while selection != -1, it will run.

    }

    private void printMenu() {
        System.out.println("Selct an option:\n--------------------------------");
        for (int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + ") " + options.get(i));
        }
    }

    private void displaySodas() throws SQLException {
        List<Sodas> SodaList = sodaDao.getAllSodas();
        for (Sodas s : SodaList) {
            System.out.println(s.getSodaId() + ": " + s.getSodaName());
        }
    }

    private void updateSoda() throws SQLException {
        System.out.println("Enter the ID # of the Soda you wish to change: ");
        String nl = scanner.nextLine();
        Integer id = null;
        try { // try/catch here in case id input is malformed
            id = Integer.parseInt( nl );
        } catch (NumberFormatException e ) {
            System.out.println("Please input an ID #.");
            return;
```

Menu pt. 3:



```java
            return;
        }
        if (id != null ) {
            System.out.println("Enter the new Soda name: ");
            String name = scanner.nextLine();
            if (!name.isEmpty() ) {
                sodaDao.updateSodas(id, name);
            }
    }
}

    private void createSoda() throws SQLException {
        System.out.print("Enter a new soda name: ");
        String sodaName = scanner.nextLine();
        sodaDao.createNewSoda(sodaName);
    }

    private void deleteSoda() throws SQLException {
        System.out.print("Enter a Soda ID # to delete: ");
        int id = Integer.parseInt(scanner.nextLine());
        sodaDao.deleteSodaById(id);
    }
}
```

Application Class:

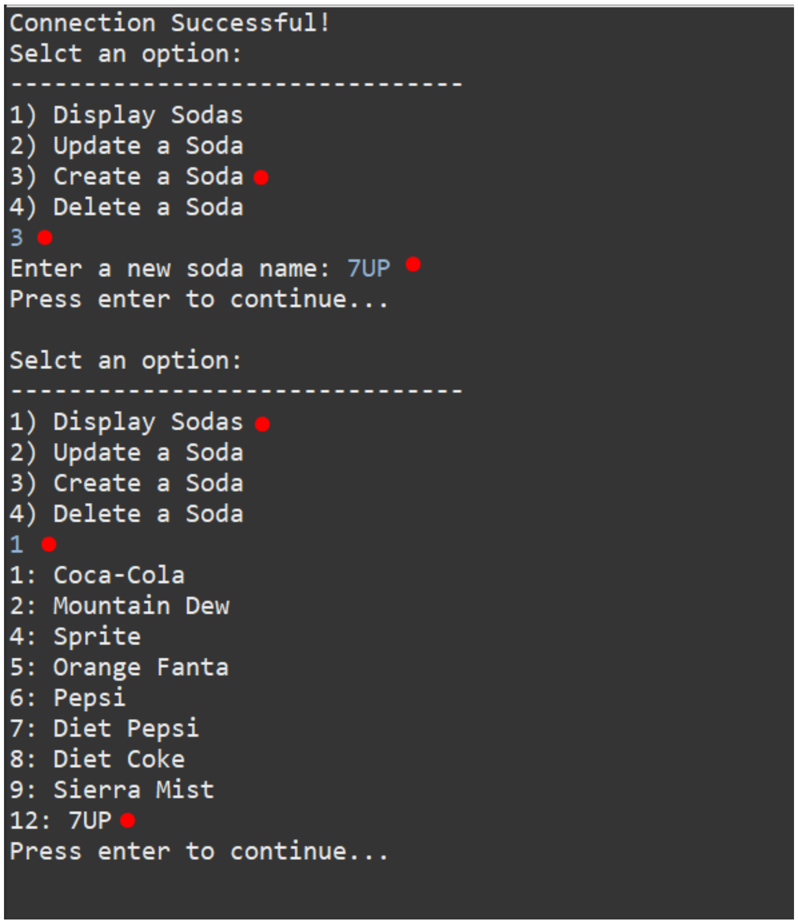


```java
package Application;


public class Wk5SQLApplication {

    public static void main(String[] args) {
        Menu menu = new Menu();
        menu.start();
    }

}

```

Screenshots of Running Application:

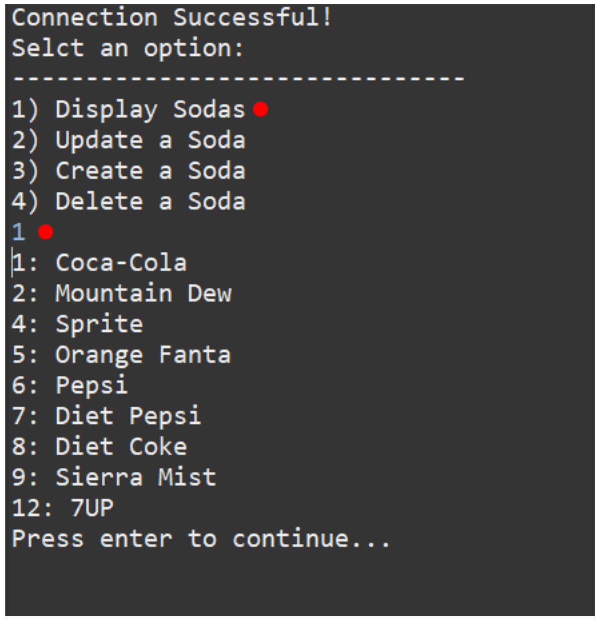
(apologies in advance for misspelling of “Select.” I fixed it in my code only after taking all these screenshots -_-)

1. Create:



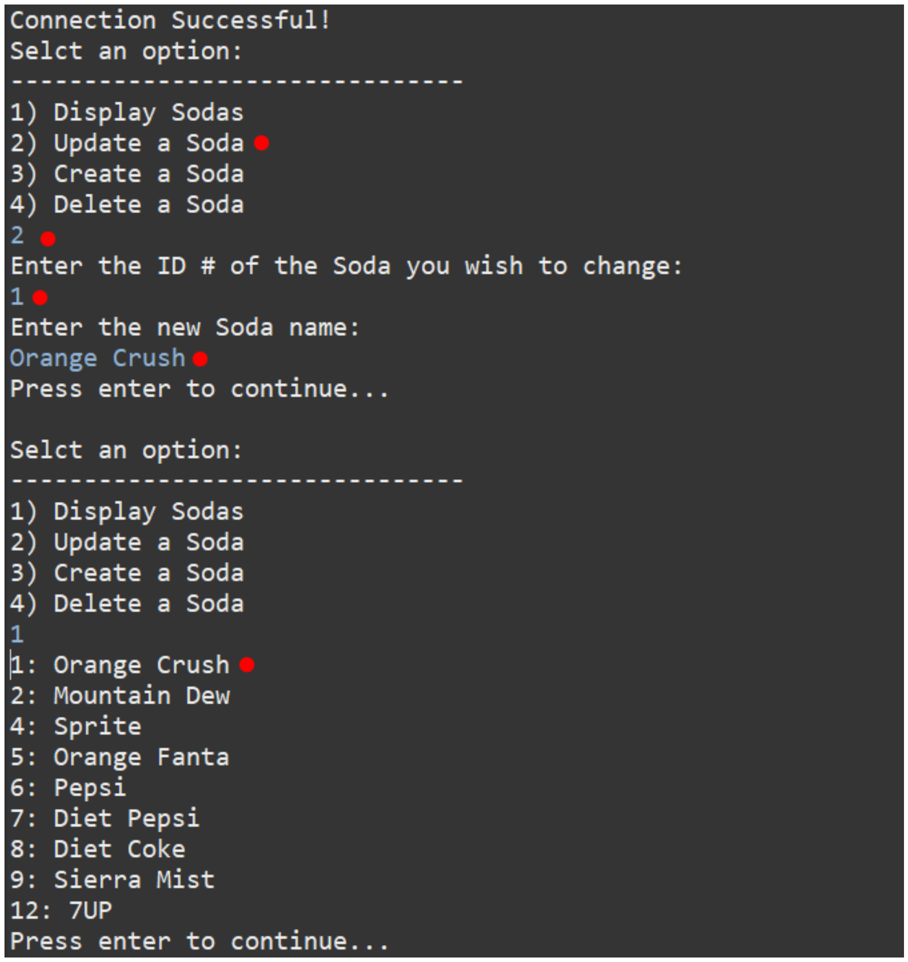
```
Connection Successful!
Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda
3
Enter a new soda name: 7UP
Press enter to continue...

Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda
1
1: Coca-Cola
2: Mountain Dew
4: Sprite
5: Orange Fanta
6: Pepsi
7: Diet Pepsi
8: Diet Coke
9: Sierra Mist
12: 7UP
Press enter to continue...
```

2. Read:



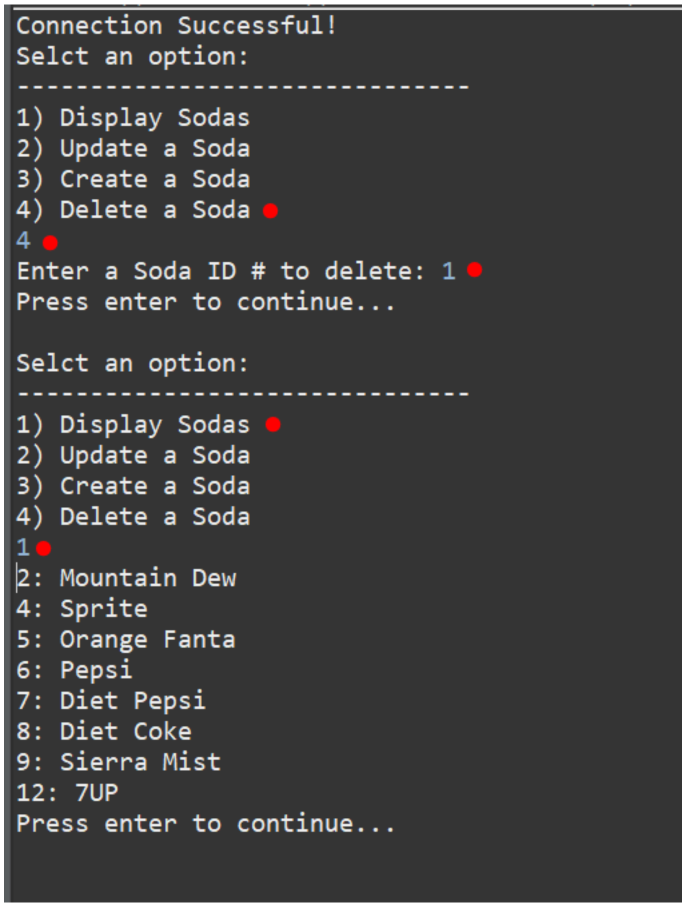
```
Connection Successful!
Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda
1
1: Coca-Cola
2: Mountain Dew
4: Sprite
5: Orange Fanta
6: Pepsi
7: Diet Pepsi
8: Diet Coke
9: Sierra Mist
12: 7UP
Press enter to continue...
```

3. Update



```
Connection Successful!
Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda
2
Enter the ID # of the Soda you wish to change:
1
Enter the new Soda name:
Orange Crush
Press enter to continue...

Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda
1
1: Orange Crush
2: Mountain Dew
4: Sprite
5: Orange Fanta
6: Pepsi
7: Diet Pepsi
8: Diet Coke
9: Sierra Mist
12: 7UP
Press enter to continue...
```

4. Delete



```
Connection Successful!
Selct an option:
-------------------------------
1) Display Sodas
2) Update a Soda
3) Create a Soda
4) Delete a Soda ●
4 ●
Enter a Soda ID # to delete: 1 ●
Press enter to continue...

Selct an option:
-------------------------------
1) Display Sodas ●
2) Update a Soda
3) Create a Soda
4) Delete a Soda
1 ●
2: Mountain Dew
4: Sprite
5: Orange Fanta
6: Pepsi
7: Diet Pepsi
8: Diet Coke
9: Sierra Mist
12: 7UP
Press enter to continue...
```

URL to GitHub Repository:

https://github.com/wlindstrom55/MYSQL-wk5-coding-assignments