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.

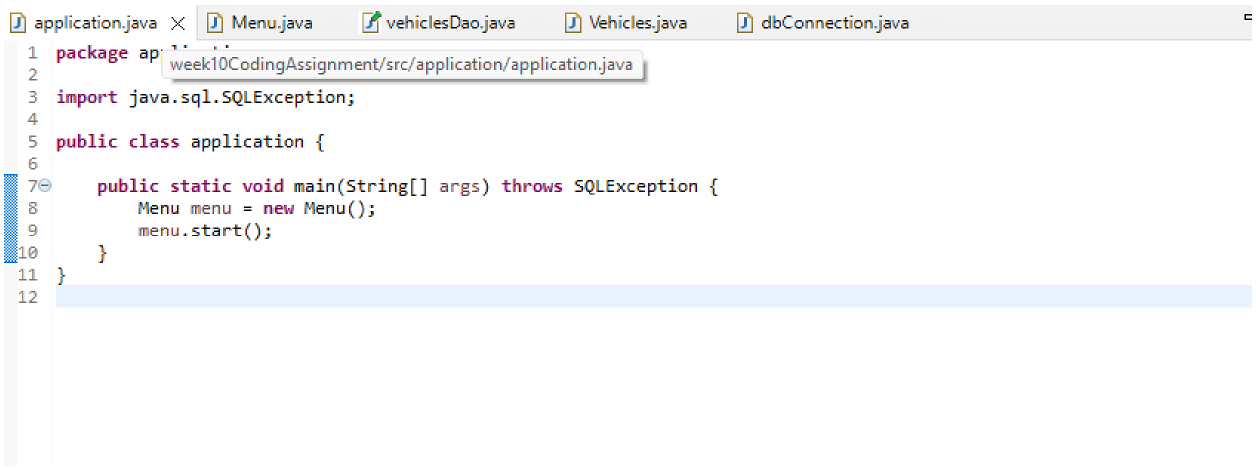
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

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.

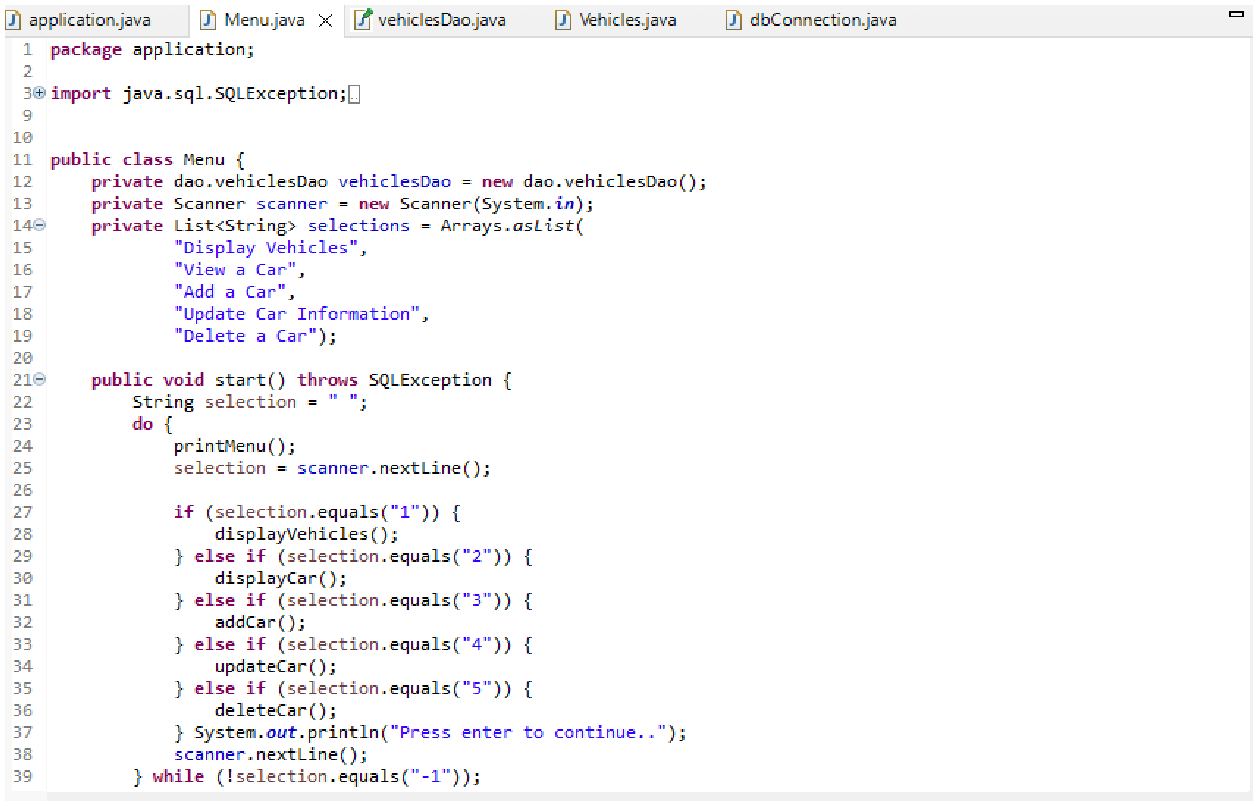
<u>Screenshots of Code:</u>

Application class:

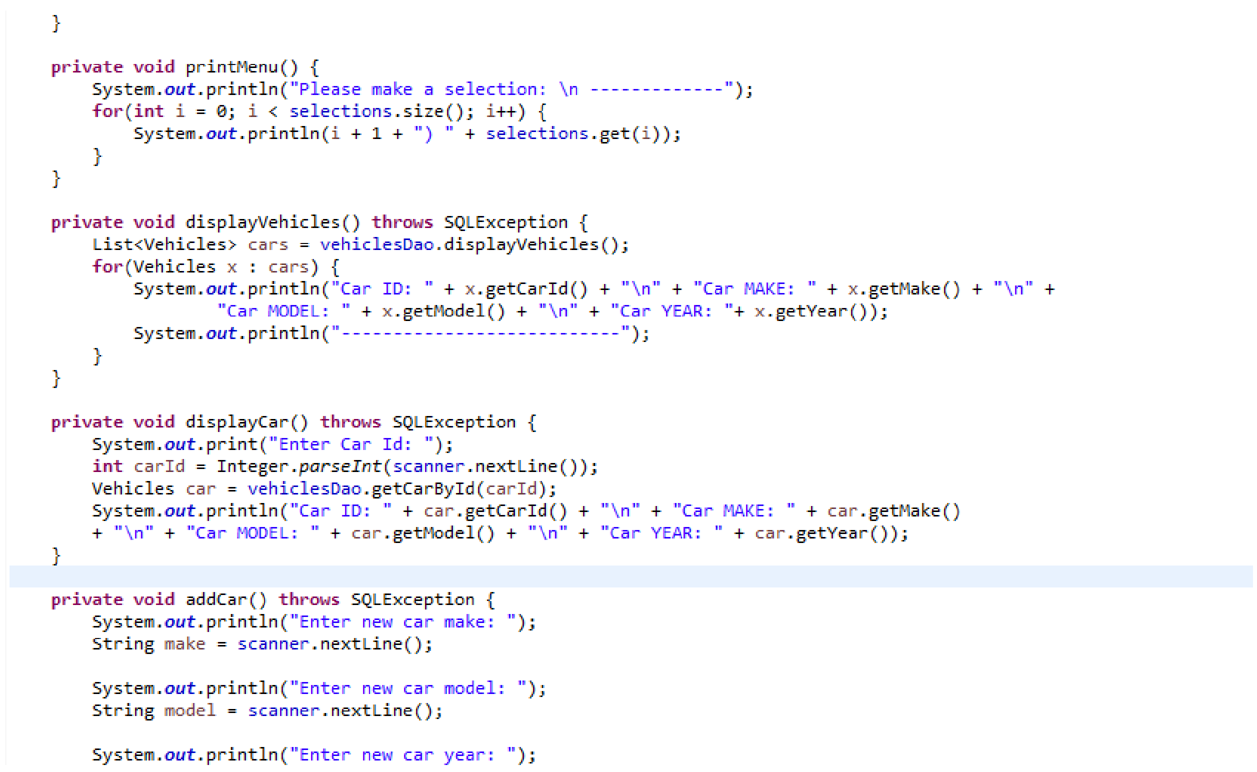


```
package ap

import java.sql.SQLException;

public class application {

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

```

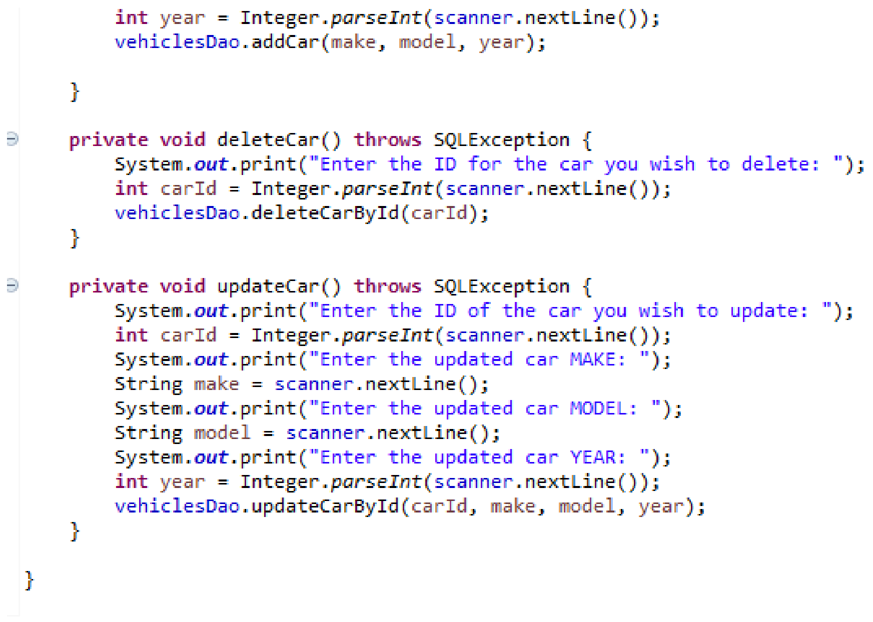
Menu:



```java
package application;

import java.sql.SQLException;


public class Menu {
    private dao.vehiclesDao vehiclesDao = new dao.vehiclesDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> selections = Arrays.asList(
            "Display Vehicles",
            "View a Car",
            "Add a Car",
            "Update Car Information",
            "Delete a Car");

    public void start() throws SQLException {
        String selection = " ";
        do {
            printMenu();
            selection = scanner.nextLine();

            if (selection.equals("1")) {
                displayVehicles();
            } else if (selection.equals("2")) {
                displayCar();
            } else if (selection.equals("3")) {
                addCar();
            } else if (selection.equals("4")) {
                updateCar();
            } else if (selection.equals("5")) {
                deleteCar();
            } System.out.println("Press enter to continue..");
            scanner.nextLine();
        } while (!selection.equals("-1"));
```

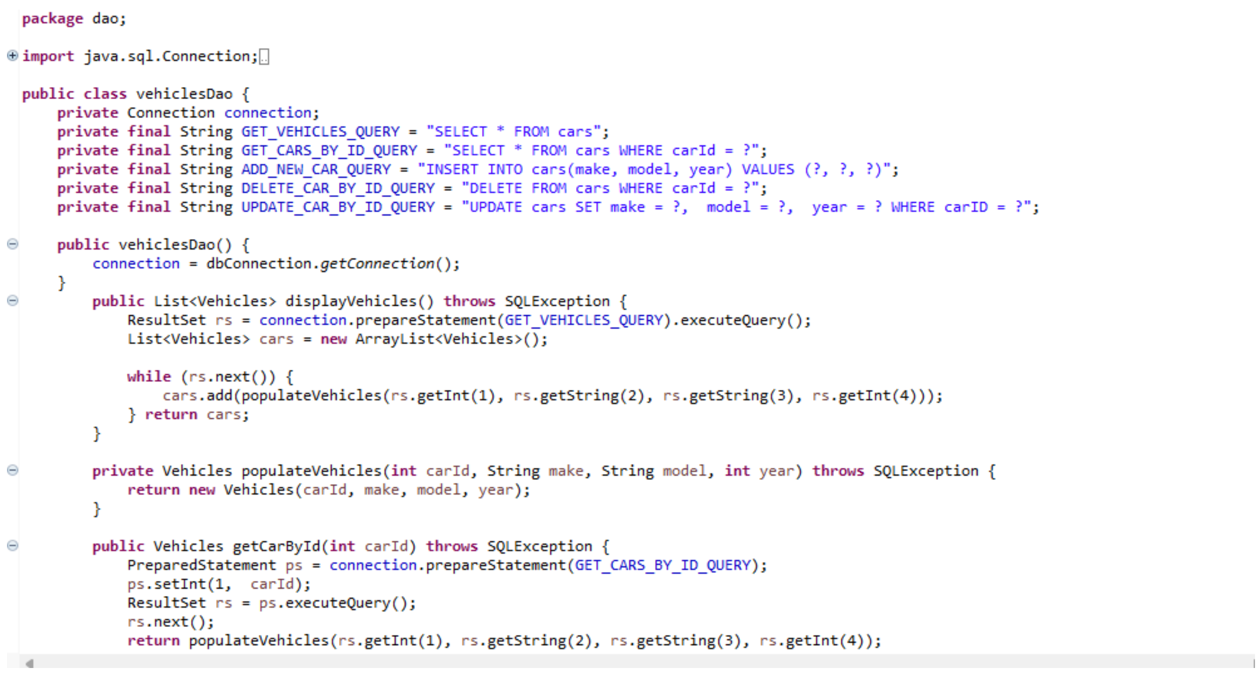


```java
    }

    private void printMenu() {
        System.out.println("Please make a selection: \n -------------");
        for(int i = 0; i < selections.size(); i++) {
            System.out.println(i + 1 + ") " + selections.get(i));
        }
    }

    private void displayVehicles() throws SQLException {
        List<Vehicles> cars = vehiclesDao.displayVehicles();
        for(Vehicles x : cars) {
            System.out.println("Car ID: " + x.getCarId() + "\n" + "Car MAKE: " + x.getMake() + "\n" +
                    "Car MODEL: " + x.getModel() + "\n" + "Car YEAR: "+ x.getYear());
            System.out.println("---------------------------");
        }
    }

    private void displayCar() throws SQLException {
        System.out.print("Enter Car Id: ");
        int carId = Integer.parseInt(scanner.nextLine());
        Vehicles car = vehiclesDao.getCarById(carId);
        System.out.println("Car ID: " + car.getCarId() + "\n" + "Car MAKE: " + car.getMake()
        + "\n" + "Car MODEL: " + car.getModel() + "\n" + "Car YEAR: " + car.getYear());
    }

    private void addCar() throws SQLException {
        System.out.println("Enter new car make: ");
        String make = scanner.nextLine();

        System.out.println("Enter new car model: ");
        String model = scanner.nextLine();

        System.out.println("Enter new car year: ");
```

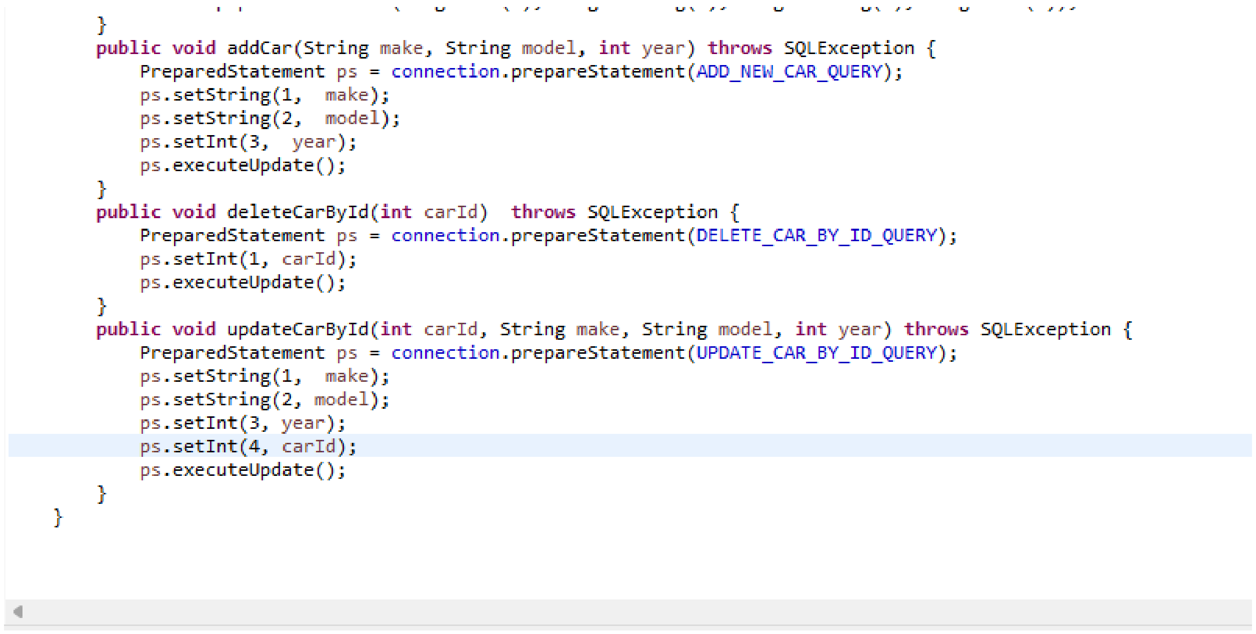


```
        int year = Integer.parseInt(scanner.nextLine());
        vehiclesDao.addCar(make, model, year);

    }

    private void deleteCar() throws SQLException {
        System.out.print("Enter the ID for the car you wish to delete: ");
        int carId = Integer.parseInt(scanner.nextLine());
        vehiclesDao.deleteCarById(carId);
    }

    private void updateCar() throws SQLException {
        System.out.print("Enter the ID of the car you wish to update: ");
        int carId = Integer.parseInt(scanner.nextLine());
        System.out.print("Enter the updated car MAKE: ");
        String make = scanner.nextLine();
        System.out.print("Enter the updated car MODEL: ");
        String model = scanner.nextLine();
        System.out.print("Enter the updated car YEAR: ");
        int year = Integer.parseInt(scanner.nextLine());
        vehiclesDao.updateCarById(carId, make, model, year);
    }

}
```

VehiclesDao:

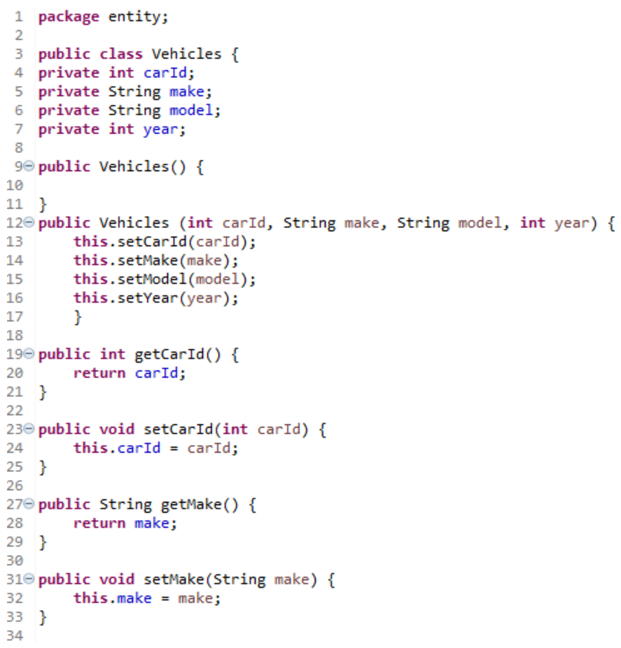


```
package dao;

import java.sql.Connection;

public class vehiclesDao {
    private Connection connection;
    private final String GET_VEHICLES_QUERY = "SELECT * FROM cars";
    private final String GET_CARS_BY_ID_QUERY = "SELECT * FROM cars WHERE carId = ?";
    private final String ADD_NEW_CAR_QUERY = "INSERT INTO cars(make, model, year) VALUES (?, ?, ?)";
    private final String DELETE_CAR_BY_ID_QUERY = "DELETE FROM cars WHERE carId = ?";
    private final String UPDATE_CAR_BY_ID_QUERY = "UPDATE cars SET make = ?,  model = ?,  year = ? WHERE carID = ?";

    public vehiclesDao() {
        connection = dbConnection.getConnection();
    }
        public List<Vehicles> displayVehicles() throws SQLException {
            ResultSet rs = connection.prepareStatement(GET_VEHICLES_QUERY).executeQuery();
            List<Vehicles> cars = new ArrayList<Vehicles>();

            while (rs.next()) {
                cars.add(populateVehicles(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4)));
            } return cars;
        }

        private Vehicles populateVehicles(int carId, String make, String model, int year) throws SQLException {
            return new Vehicles(carId, make, model, year);
        }

        public Vehicles getCarById(int carId) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(GET_CARS_BY_ID_QUERY);
            ps.setInt(1,  carId);
            ResultSet rs = ps.executeQuery();
            rs.next();
            return populateVehicles(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getInt(4));
```

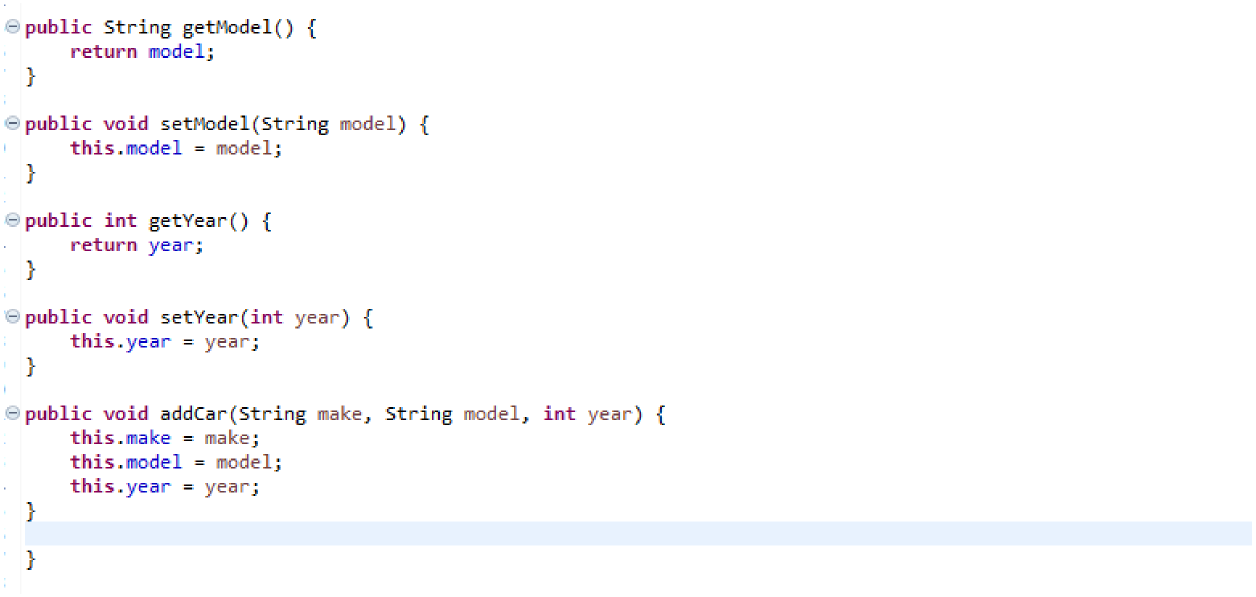


```java
        }
        public void addCar(String make, String model, int year) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(ADD_NEW_CAR_QUERY);
            ps.setString(1,  make);
            ps.setString(2,  model);
            ps.setInt(3,  year);
            ps.executeUpdate();
        }
        public void deleteCarById(int carId)  throws SQLException {
            PreparedStatement ps = connection.prepareStatement(DELETE_CAR_BY_ID_QUERY);
            ps.setInt(1, carId);
            ps.executeUpdate();
        }
        public void updateCarById(int carId, String make, String model, int year) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(UPDATE_CAR_BY_ID_QUERY);
            ps.setString(1,  make);
            ps.setString(2, model);
            ps.setInt(3, year);
            ps.setInt(4, carId);
            ps.executeUpdate();
        }
    }
```

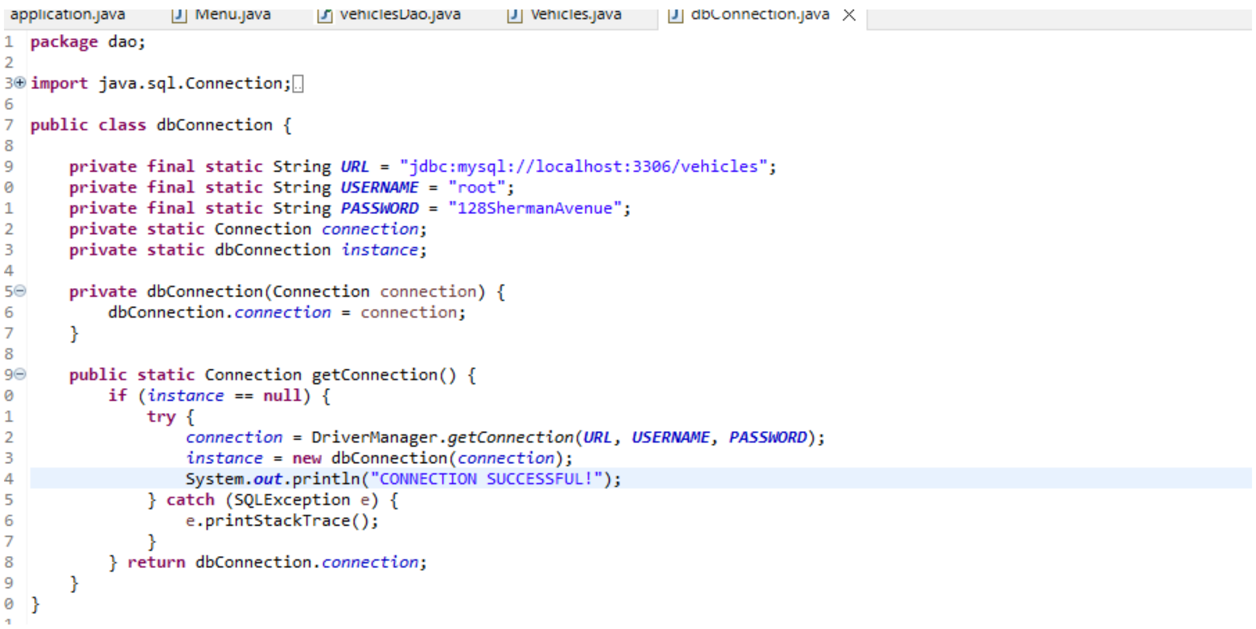
Vehicles(entity) class:



```java
package entity;

public class Vehicles {
private int carId;
private String make;
private String model;
private int year;

public Vehicles() {

}
public Vehicles (int carId, String make, String model, int year) {
    this.setCarId(carId);
    this.setMake(make);
    this.setModel(model);
    this.setYear(year);
    }

public int getCarId() {
    return carId;
}

public void setCarId(int carId) {
    this.carId = carId;
}

public String getMake() {
    return make;
}

public void setMake(String make) {
    this.make = make;
}

```


```
public String getModel() {
    return model;
}

public void setModel(String model) {
    this.model = model;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

public void addCar(String make, String model, int year) {
    this.make = make;
    this.model = model;
    this.year = year;
}

}
```

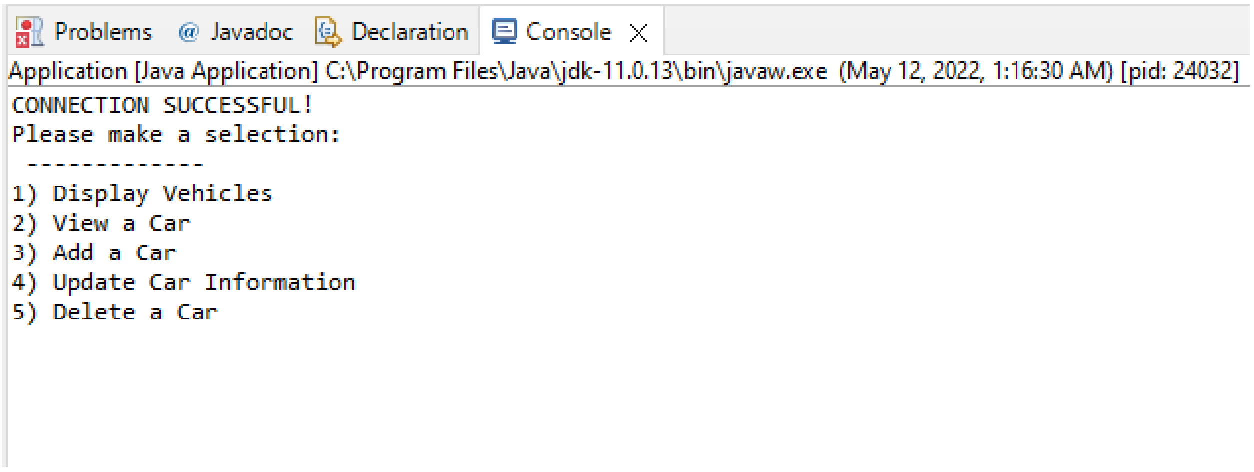
dbConnection class:



```
package dao;

import java.sql.Connection;

public class dbConnection {

    private final static String URL = "jdbc:mysql://localhost:3306/vehicles";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "128ShermanAvenue";
    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;
    }
}
```

Screenshots of Running Application:

Application launching:

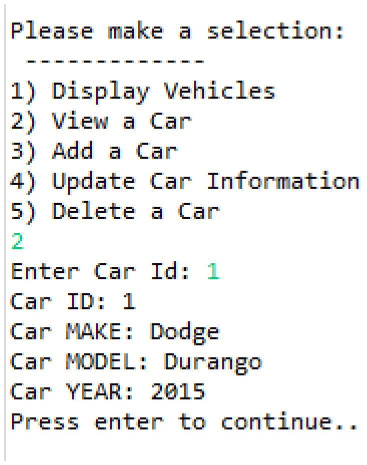


```
Application [Java Application] C:\Program Files\Java\jdk-11.0.13\bin\javaw.exe  (May 12, 2022, 1:16:30 AM) [pid: 24032]
CONNECTION SUCCESSFUL!
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
```

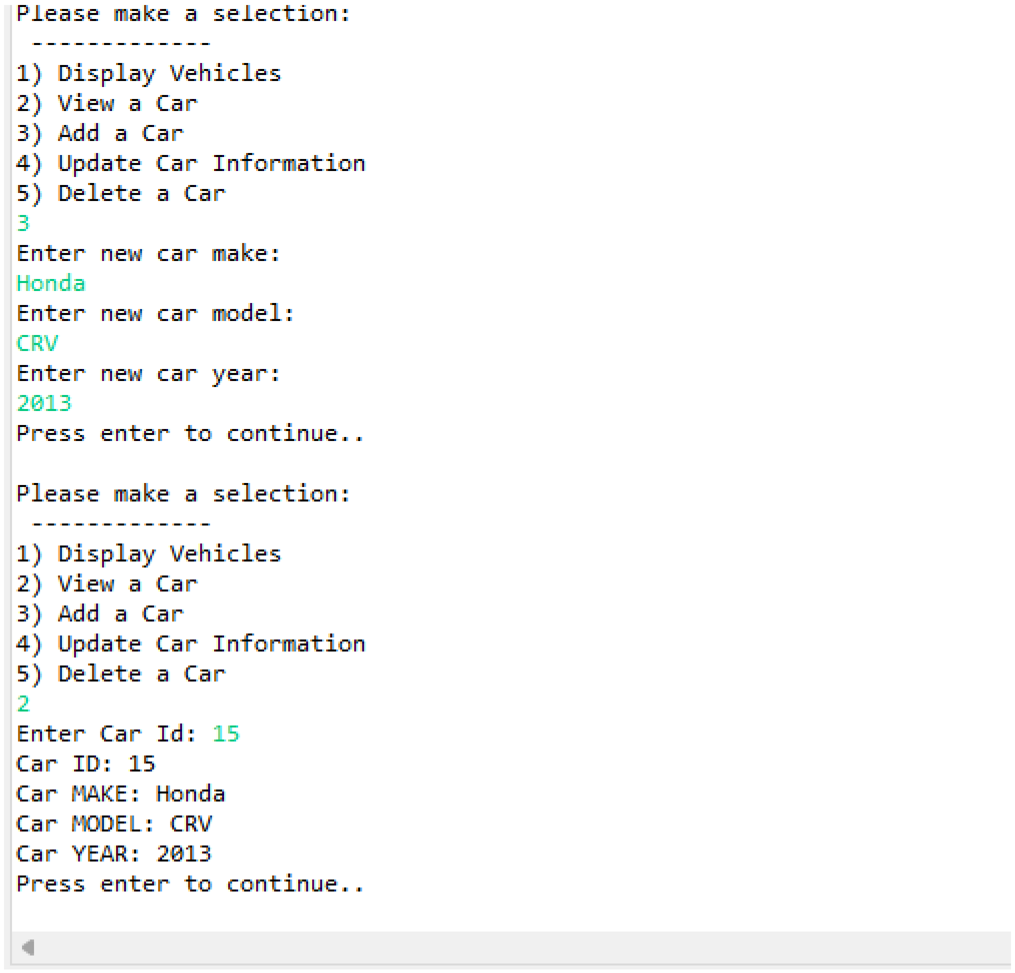
Option 1 – Display Vehicles:

```
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
1
Car ID: 1
Car MAKE: Dodge
Car MODEL: Durango
Car YEAR: 2015
---------------------------
Car ID: 4
Car MAKE: Hyundai
Car MODEL: Accent
Car YEAR: 2006
---------------------------
Car ID: 10
Car MAKE: Dodge
Car MODEL: Avenger
Car YEAR: 2013
---------------------------
Car ID: 11
Car MAKE: Ford
Car MODEL: Bronco
Car YEAR: 2022
---------------------------
Press enter to continue..
```

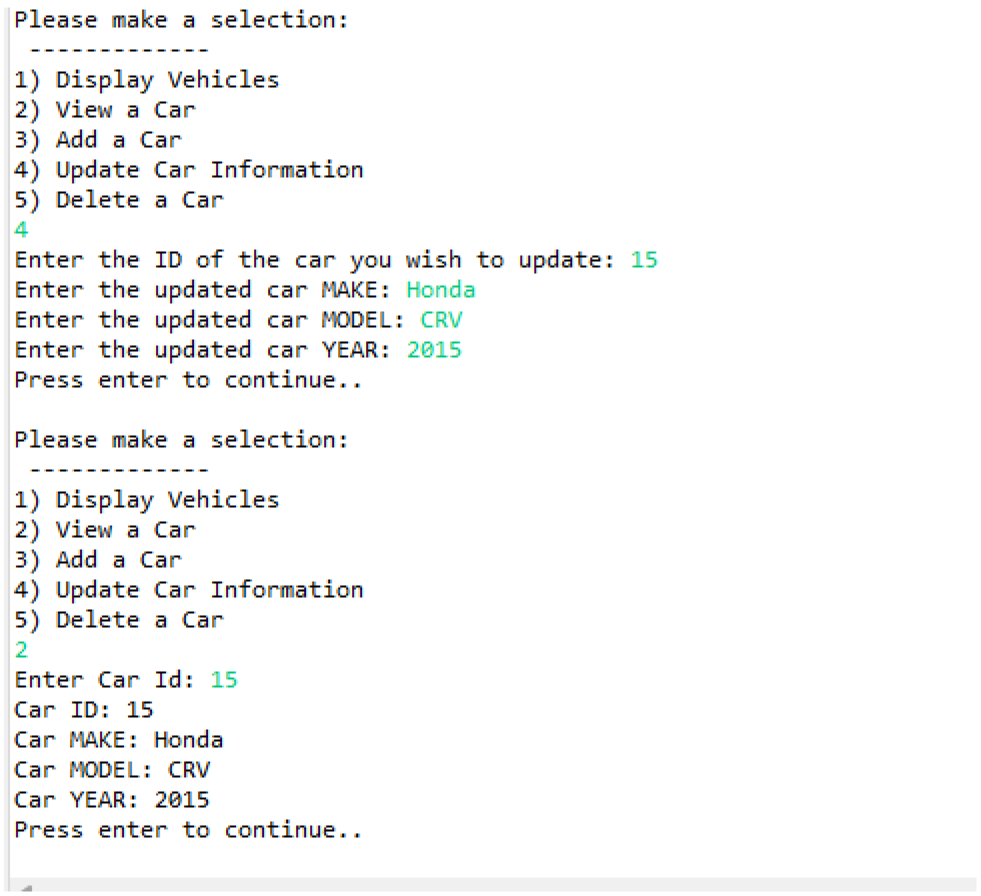
Option 2 – View a Specific Vehicle:


```
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
2
Enter Car Id: 1
Car ID: 1
Car MAKE: Dodge
Car MODEL: Durango
Car YEAR: 2015
Press enter to continue..
```

Option 3 – Add a Car:

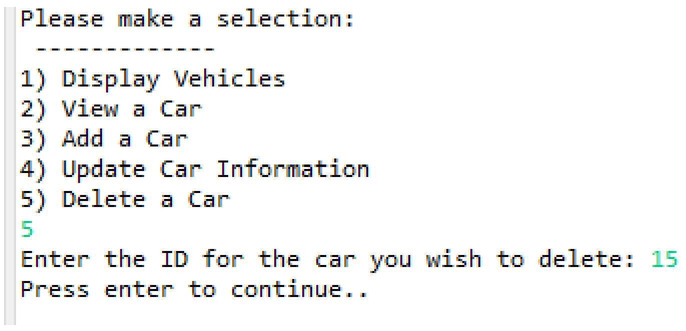

```
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
3
Enter new car make:
Honda
Enter new car model:
CRV
Enter new car year:
2013
Press enter to continue..

Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
2
Enter Car Id: 15
Car ID: 15
Car MAKE: Honda
Car MODEL: CRV
Car YEAR: 2013
Press enter to continue..
```

Option 4 – Update Car Information:

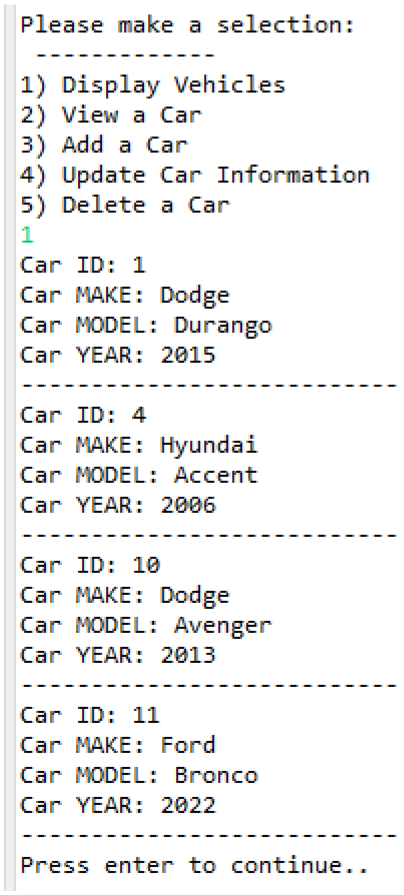


```
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
4
Enter the ID of the car you wish to update: 15
Enter the updated car MAKE: Honda
Enter the updated car MODEL: CRV
Enter the updated car YEAR: 2015
Press enter to continue..

Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
2
Enter Car Id: 15
Car ID: 15
Car MAKE: Honda
Car MODEL: CRV
Car YEAR: 2015
Press enter to continue..
```

Option 5 – Delete a Car:



```
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
5
Enter the ID for the car you wish to delete: 15
Press enter to continue..
```



```
Please make a selection:
 -------------
1) Display Vehicles
2) View a Car
3) Add a Car
4) Update Car Information
5) Delete a Car
1
Car ID: 1
Car MAKE: Dodge
Car MODEL: Durango
Car YEAR: 2015
---------------------------
Car ID: 4
Car MAKE: Hyundai
Car MODEL: Accent
Car YEAR: 2006
---------------------------
Car ID: 10
Car MAKE: Dodge
Car MODEL: Avenger
Car YEAR: 2013
---------------------------
Car ID: 11
Car MAKE: Ford
Car MODEL: Bronco
Car YEAR: 2022
---------------------------
Press enter to continue..
```

URL to GitHub Repository:

https://github.com/harvisd/week10Assignment