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.

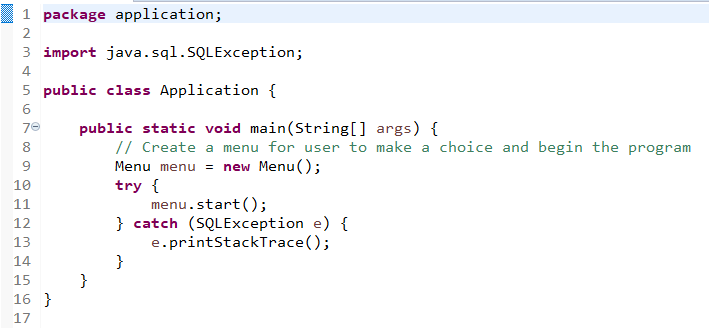
Remember that both parameters on PreparedStatements and the ResultSet columns are based on indexes that start with 1, not 0.

**Screenshots of Code:**

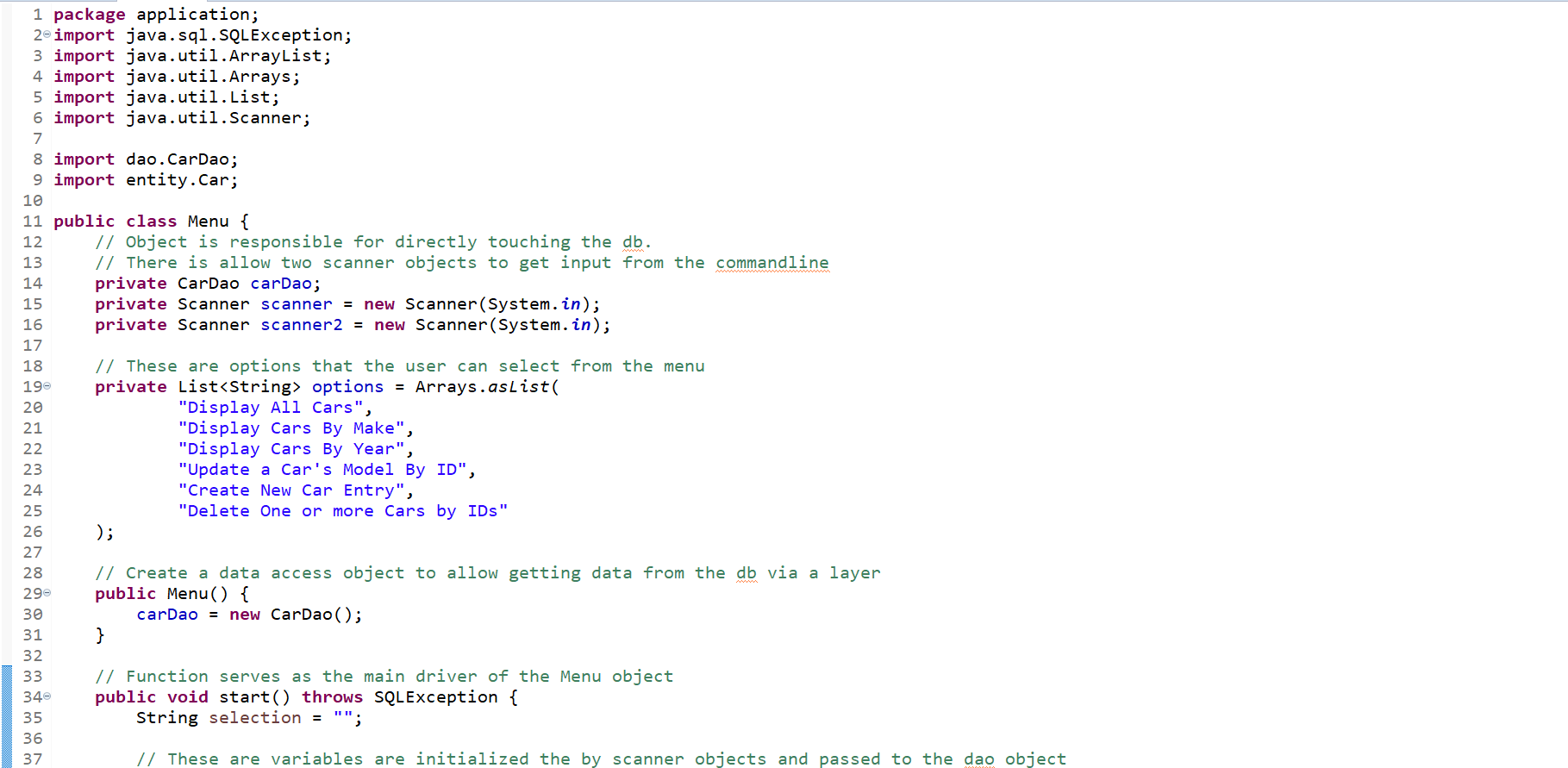
**Create\_DB.sql**

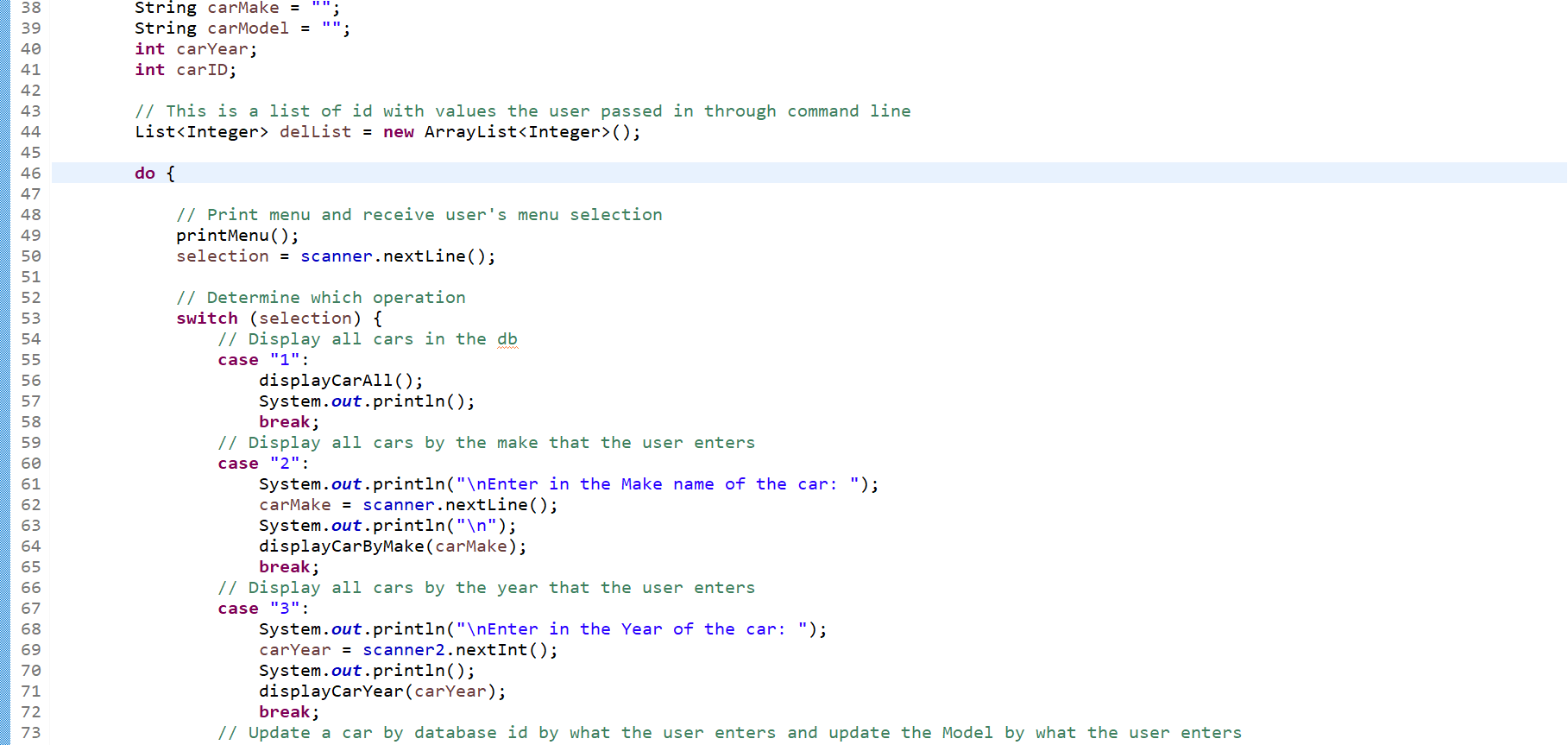
****

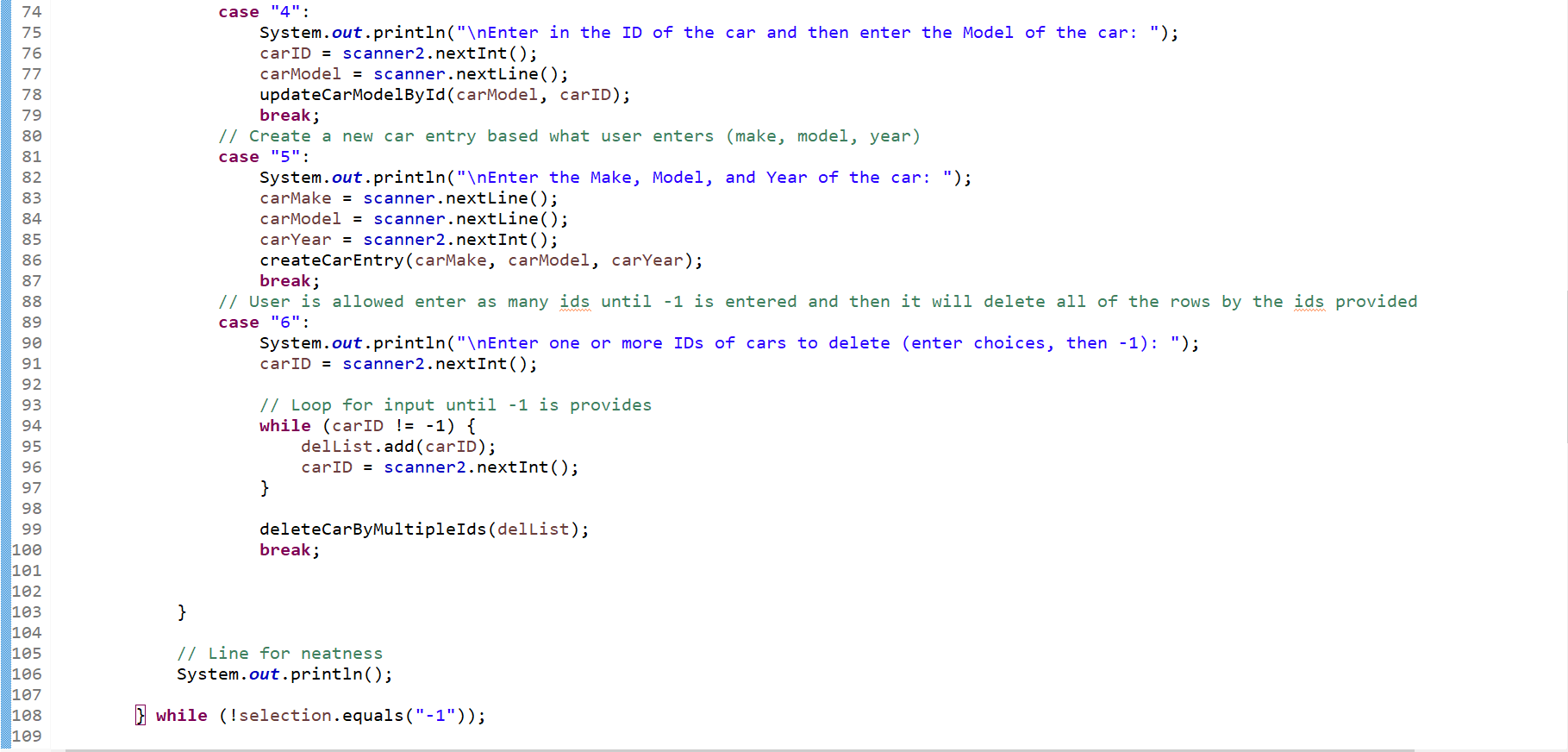
**Application.java**

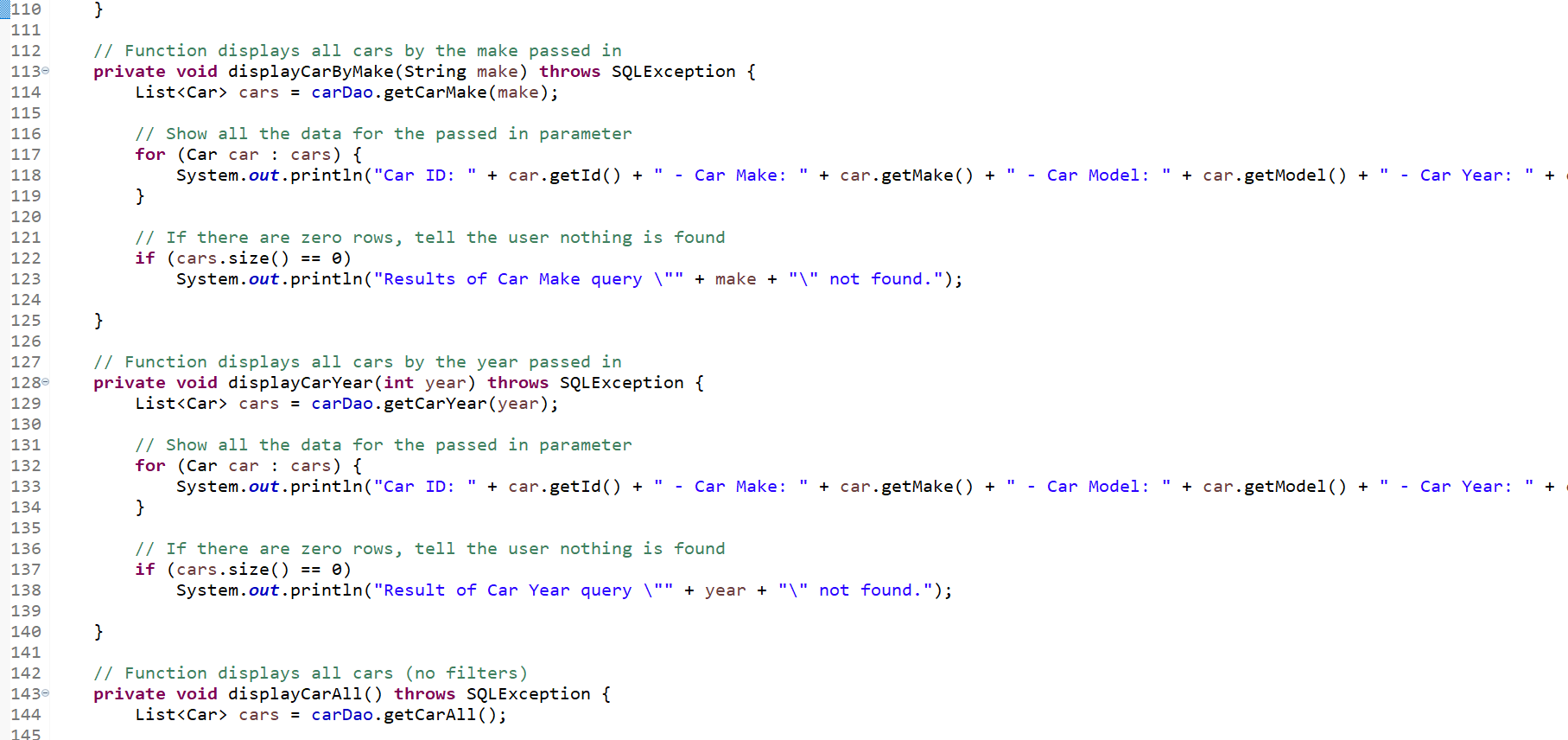


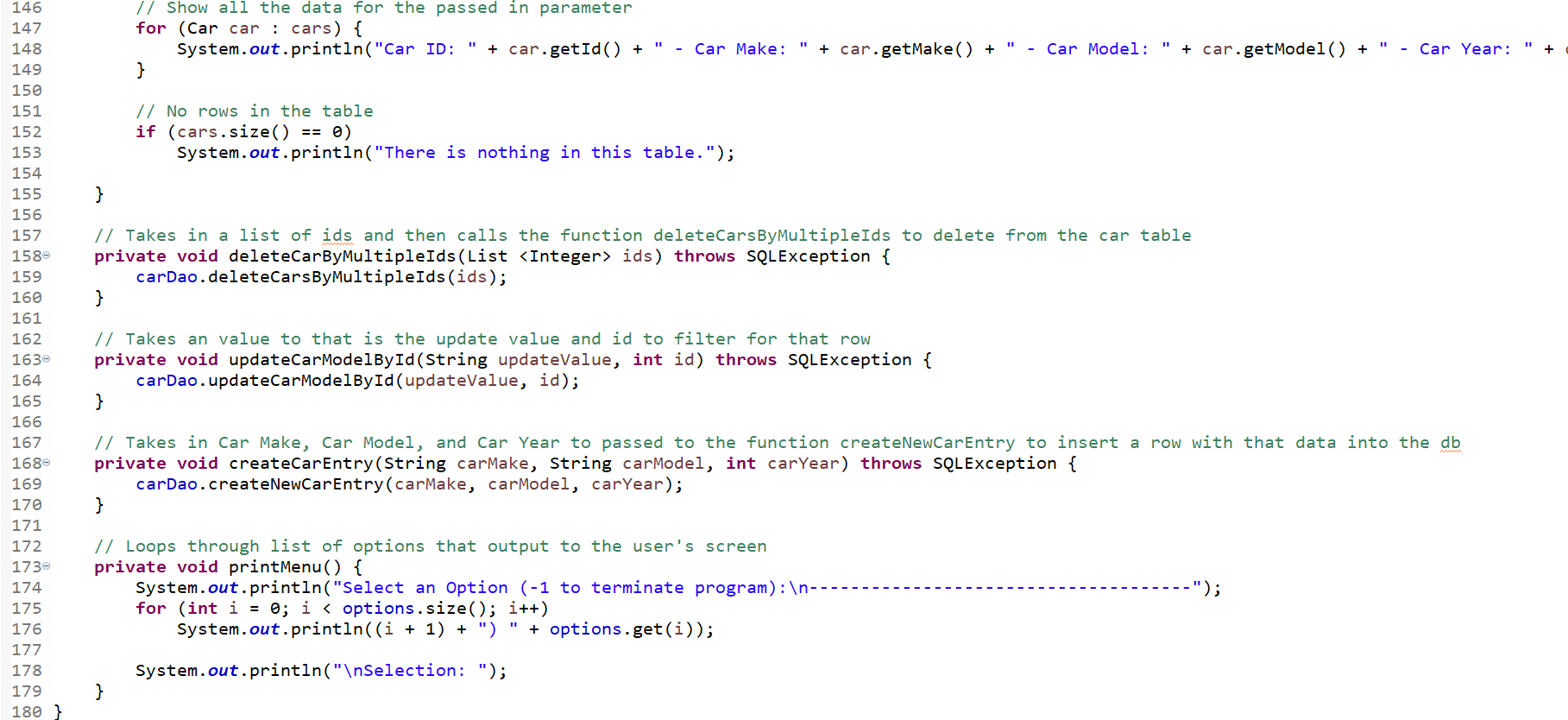
**Menu.java**



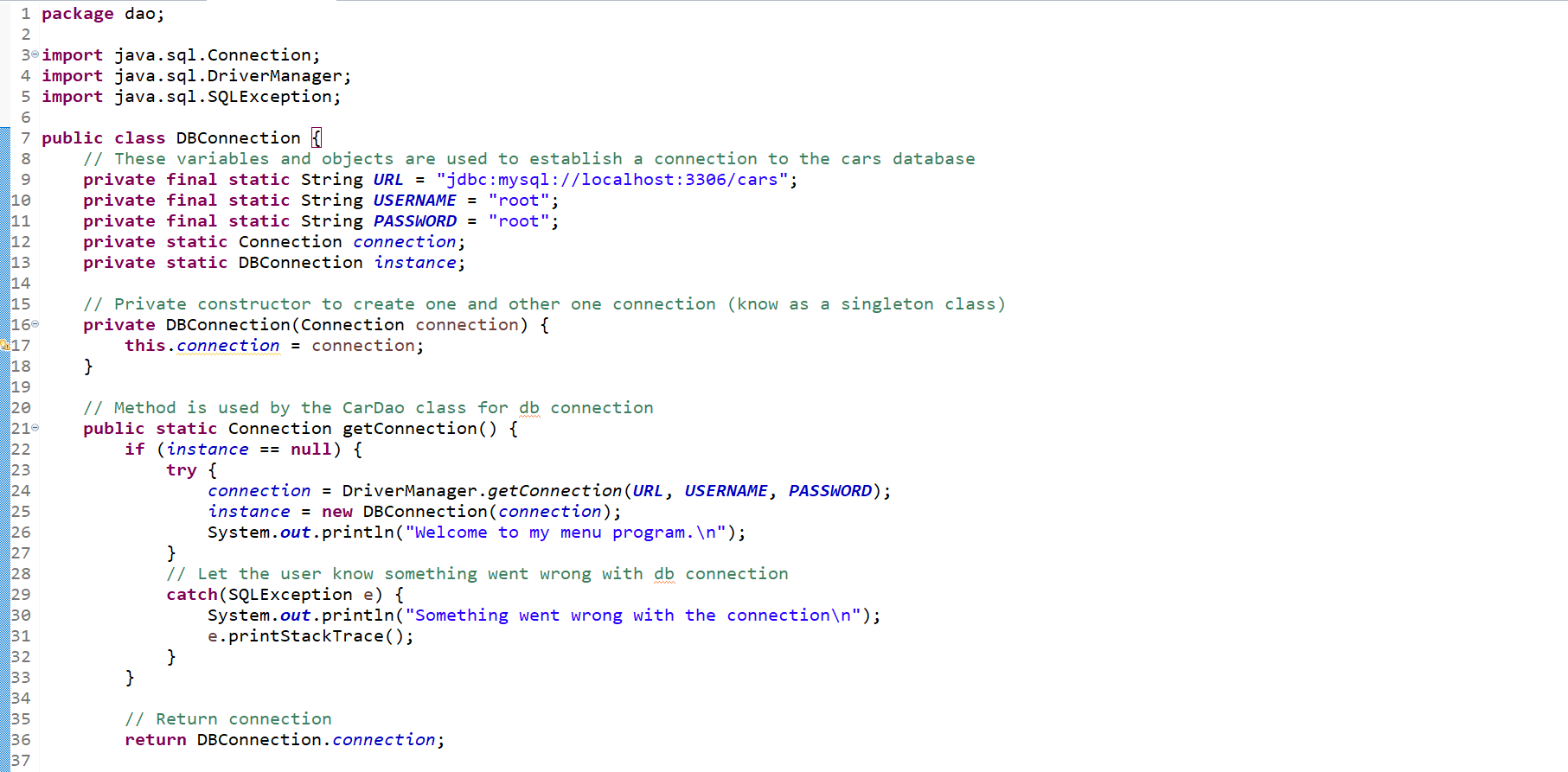




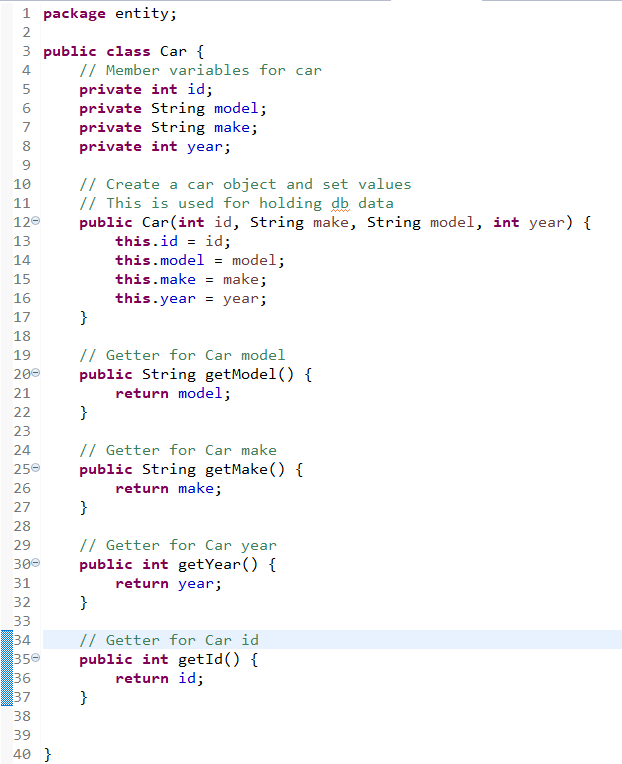




**DBConnection.java**

****

**Car.java**

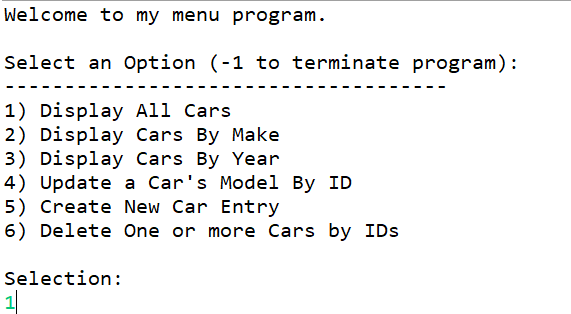
****

**CarDao.java**

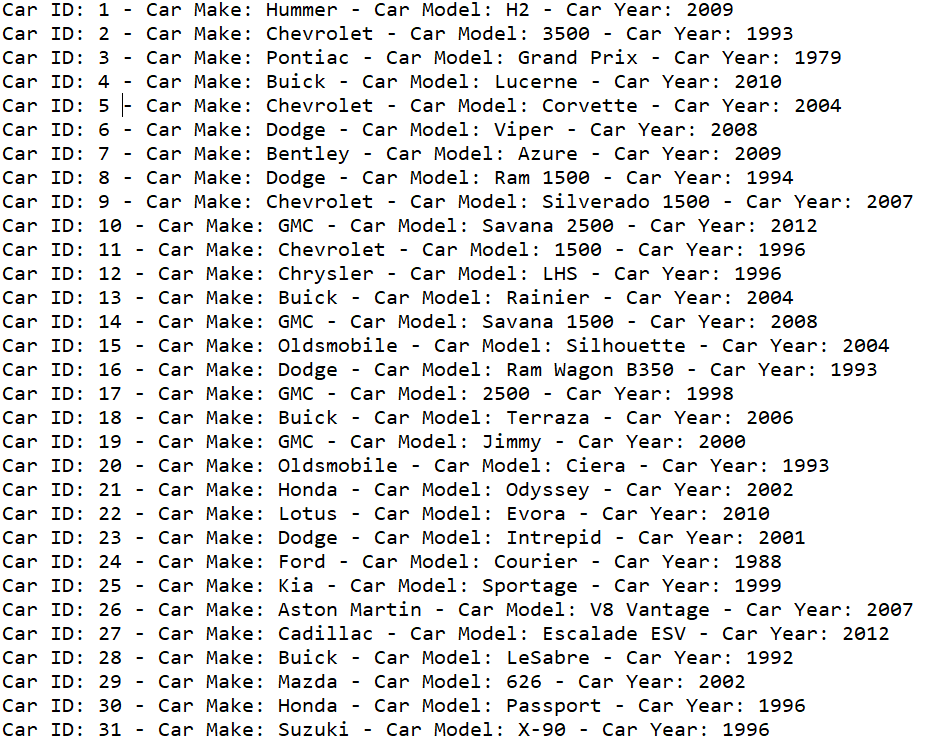


**Screenshots of Running Application:**

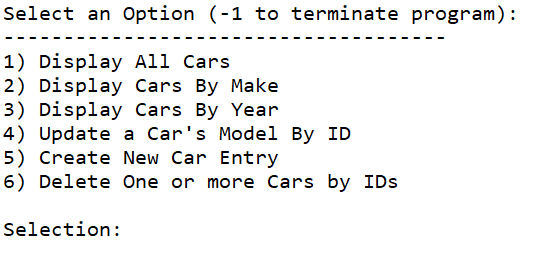
**Menu**



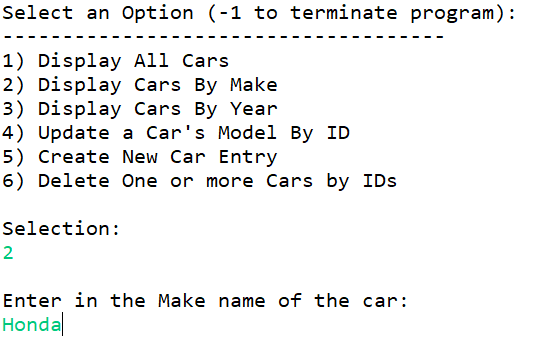
**After Selection Choice 1**

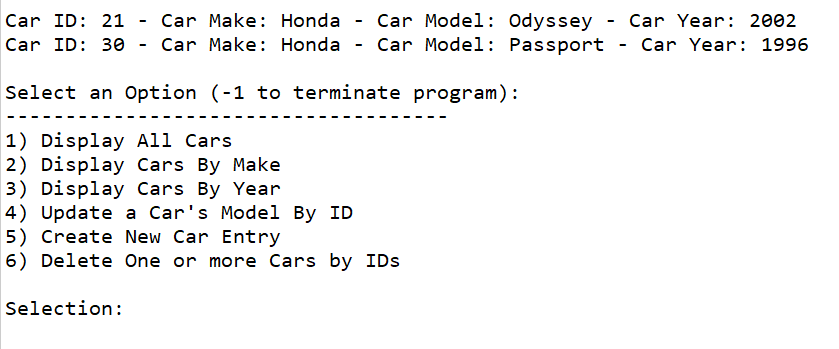


**Menu Displayed Again**

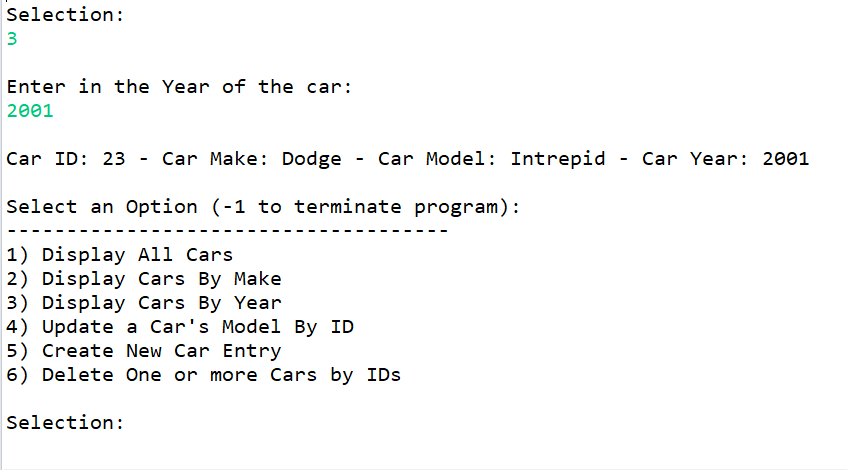


**Choice 2 selected and Car Make filter is Honda**

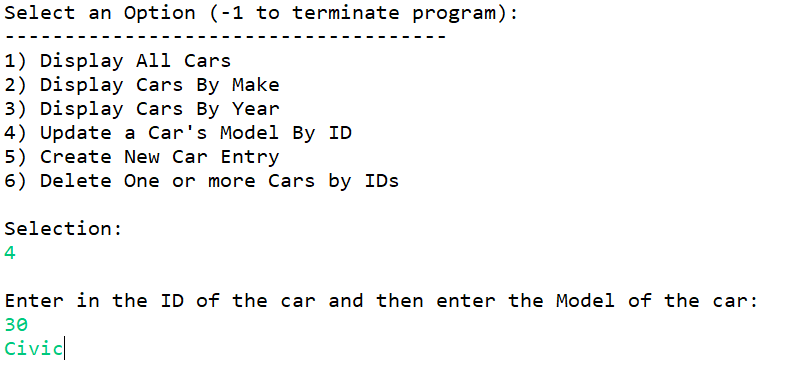




**Choice 3 and rows are selected by Car Year**



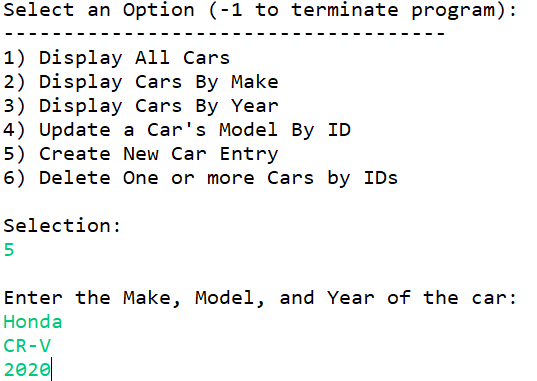
**Choice 4 selected and Car ID = 30, Car Model is being updated to Civic**



**Selected option 1 to show results of update**



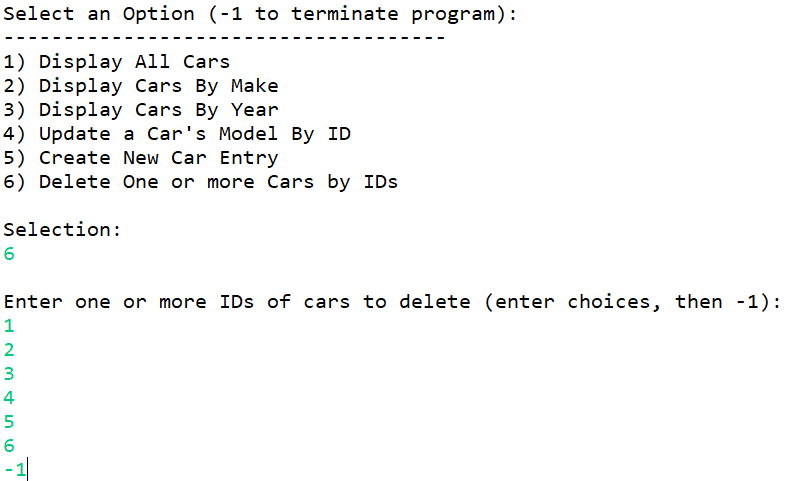
**(Option 5) Creating a new row with input given via command line**

****

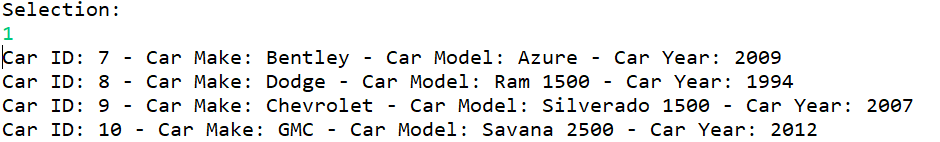
**Selected option 1 to show results of create**

****

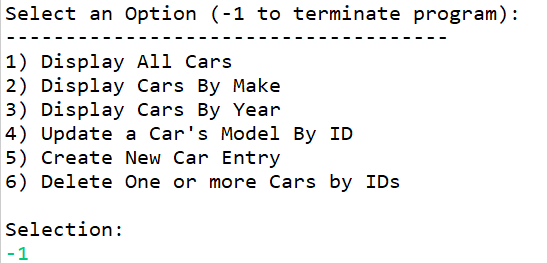
**Selected Option 6**

****

**Ran Option 1 to show rows are gone**

****

**Exited Program**

****

**URL to GitHub Repository:** <https://github.com/jamesaiello42/Intro-to-MySQL-Week-5-Coding-Assignment>