Relational Databases with MySQL Week 6 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:**

This week you will be working together as a **team** to create a full CRUD application.

Your console CRUD application will need to use a database to store all the application data.

As a team, decide what you want your project to do. Get instructor approval early in the week before beginning development.

You need to have at least 3 entities.

Users should be able to interact via the console (i.e. Scanner(System.in)))

Use git to collaborate.

Everyone will be graded on their individual contributions.

**RENTAL CAR application to demonstrate CRUD using Java-MySQL-jdbc**

This simple Rental Car Application demonstrates the basic CRUD functionalities using Java, MySQL and the jdbc connector.

We use INSERT for the Create aspect of CRUD, specifically when adding a car and adding a customer.

We use SELECT for the Read aspect of CRUD, specifically when displaying the cars, the customers, and the rental transactions.

We use UPDATE for the Update aspect of CRUD, specifically while adding the rental car transactions to update the total miles on the car that was rented by a customer.

We use DELETE for the Delete aspect of CRUD, specifically while deleting a car.

For this simple Rental Car application our database is called “rentcar”. We use three(3) entities in this application demonstration:

* Cars (rentcars table)
* Customers (customers table)
* Transactions (timerent table)

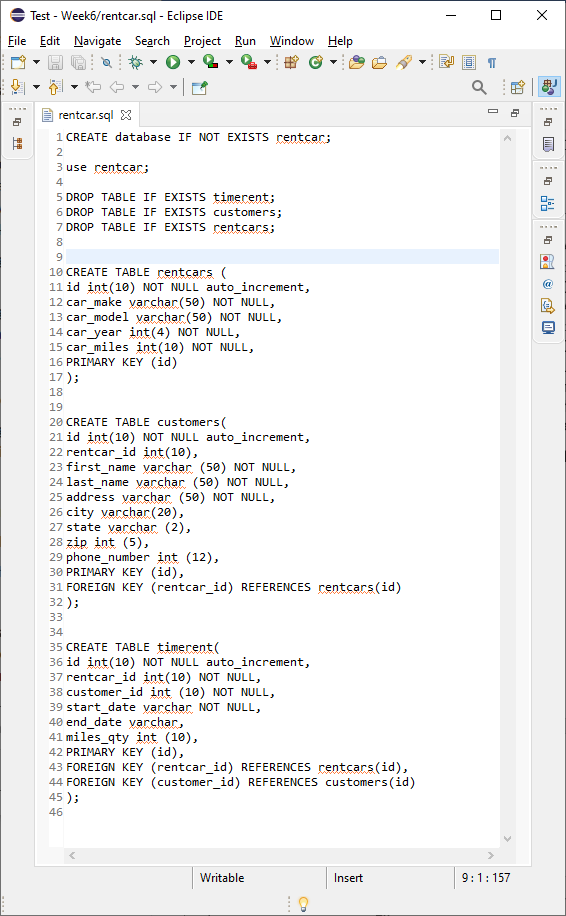
For the flow of the application testing –

1. Add a car option allows you to add the Car Make, Car Model, the YEAR of the car and the miles in the car. The table used is rentcars.
2. Display cars option outputs all the car rows in the table.
3. Add a customer option allows you to add a customer to the database in the customers table.
4. Display customers option allows you to output all the rows in the customers table.
5. Update Rental Transaction option allows you to create a transaction record while miles used during the transaction is used to update the total miles for that car in the rentcars table. This utilizes the car id and customer id fields to create the record in the timerent table.

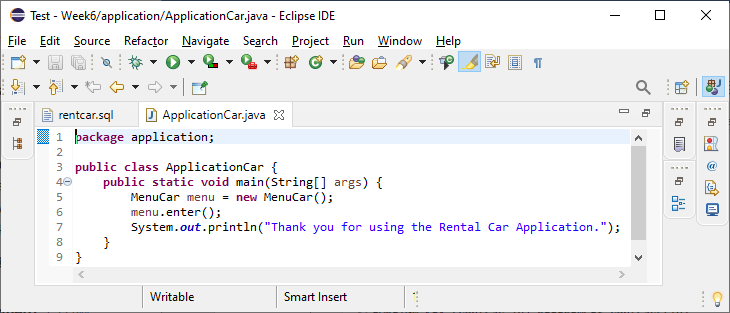
Please feel free to use this application code as you wish!

**Screenshots of Code:**

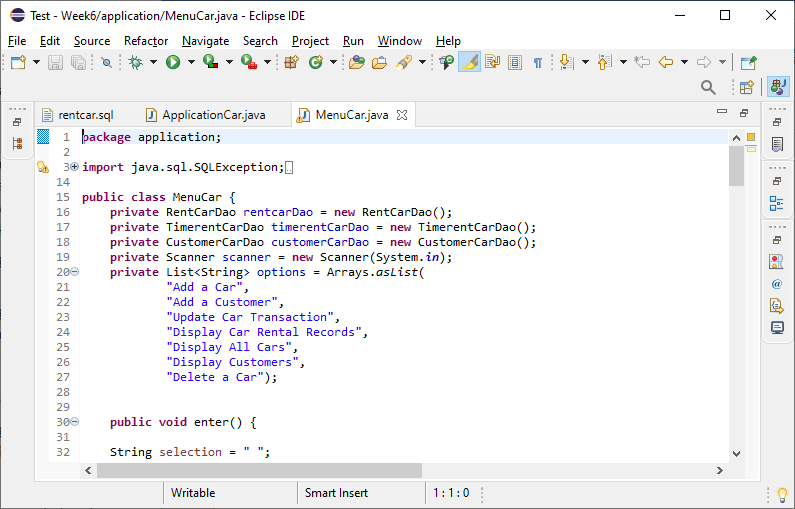
Rentcar.sql

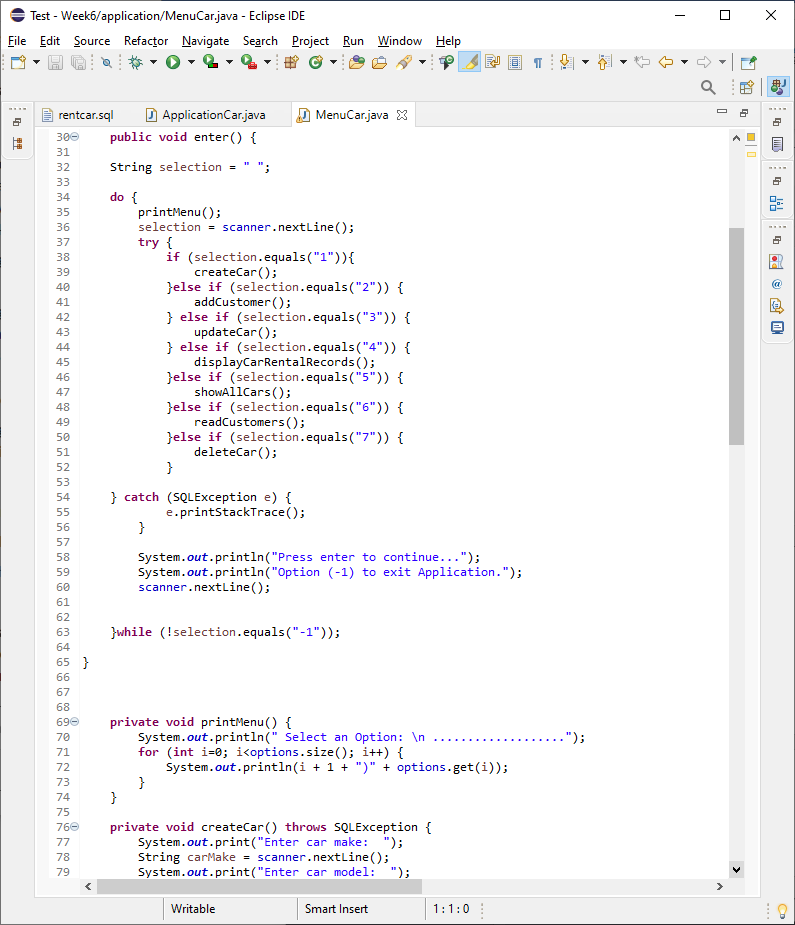


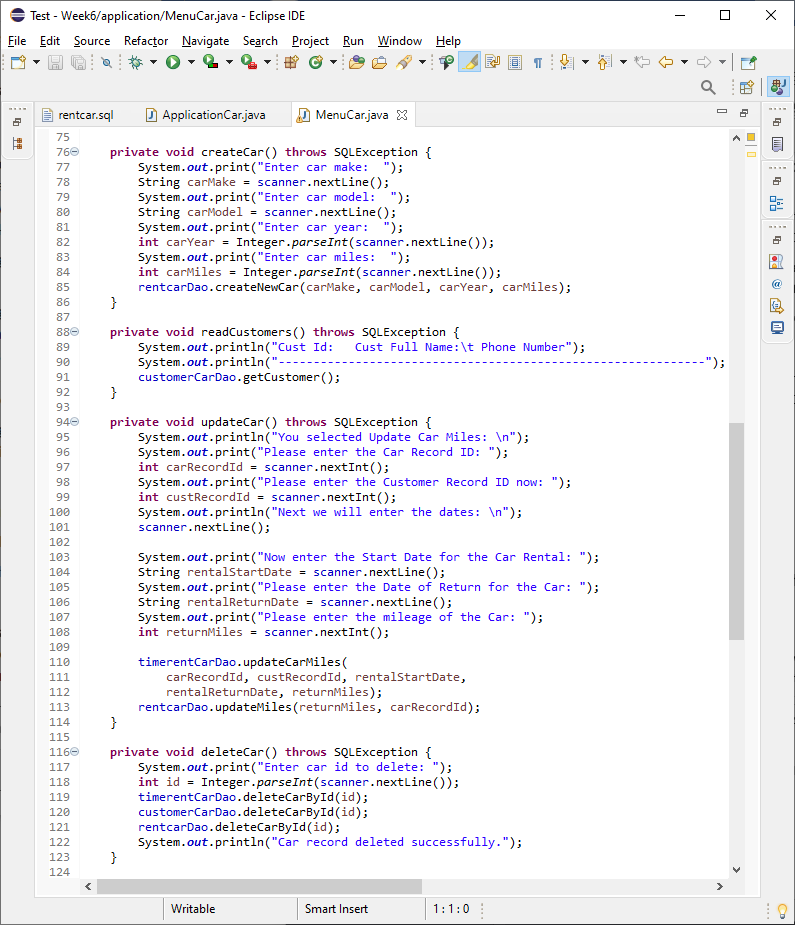
Application.java

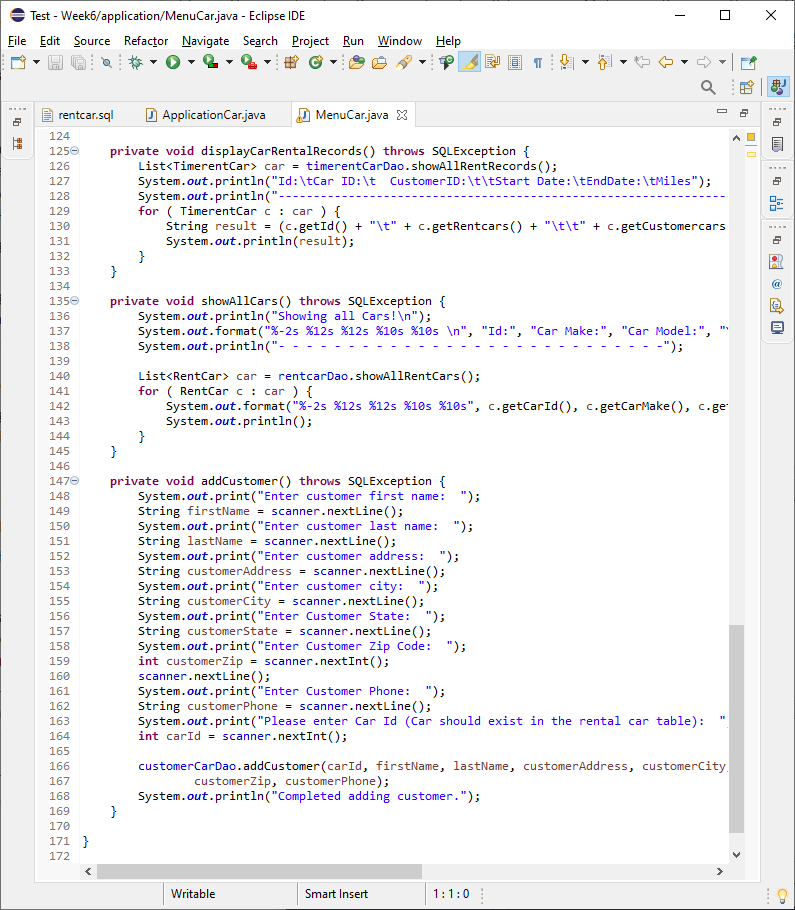


MenuCar.java

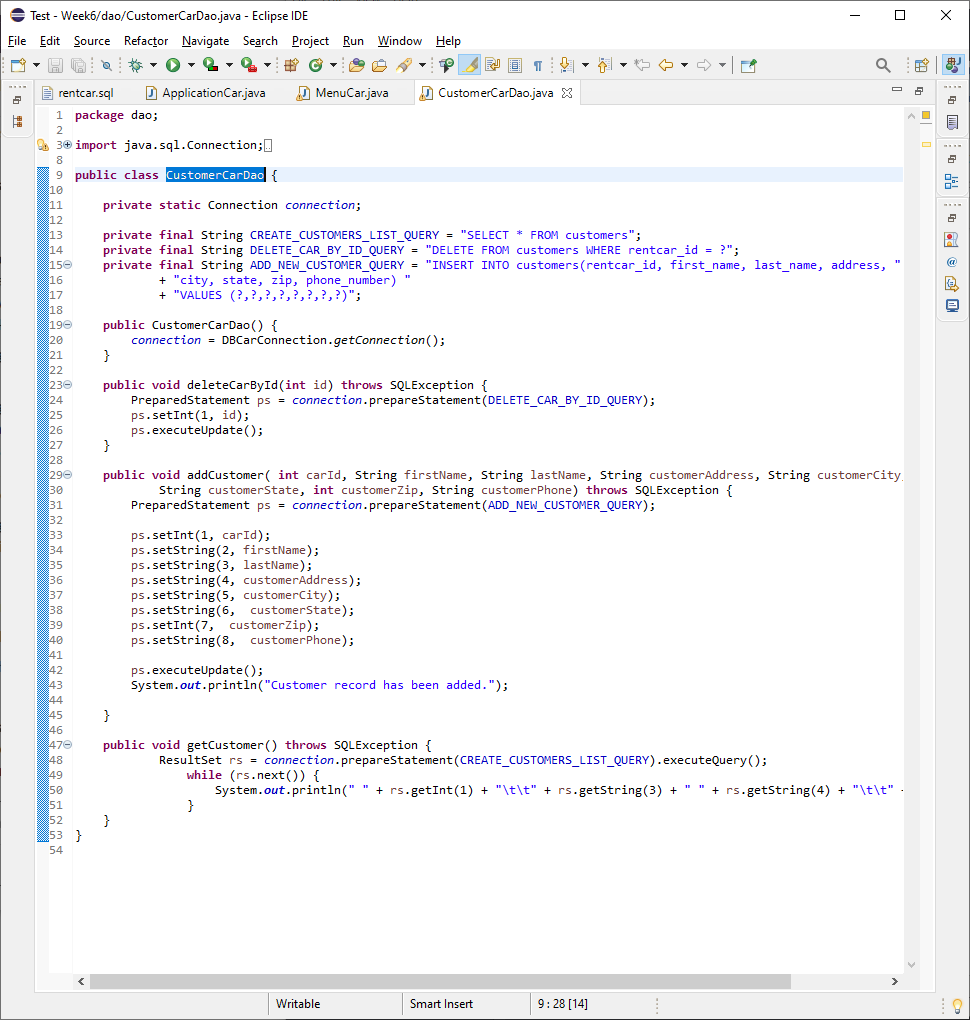




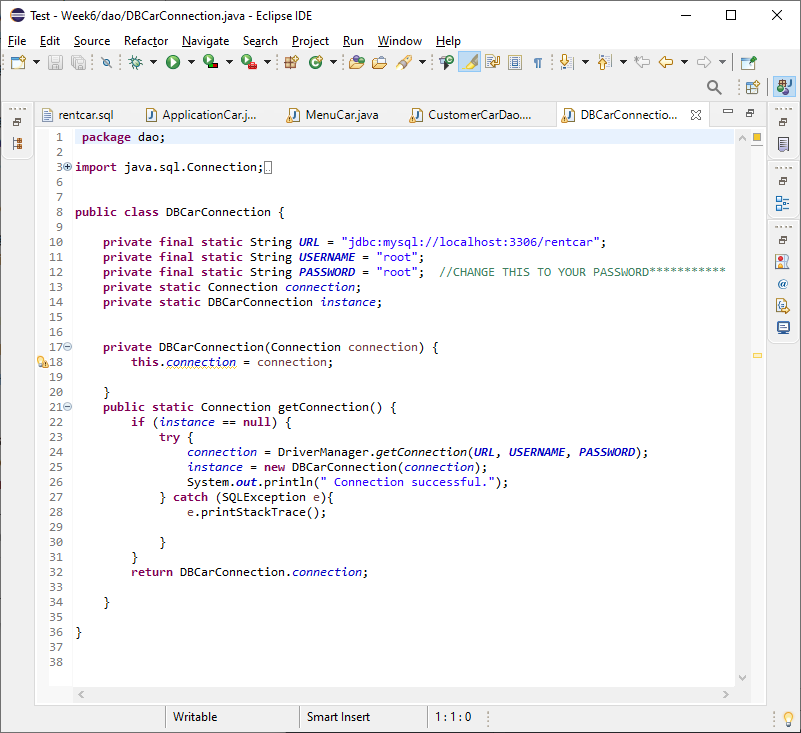




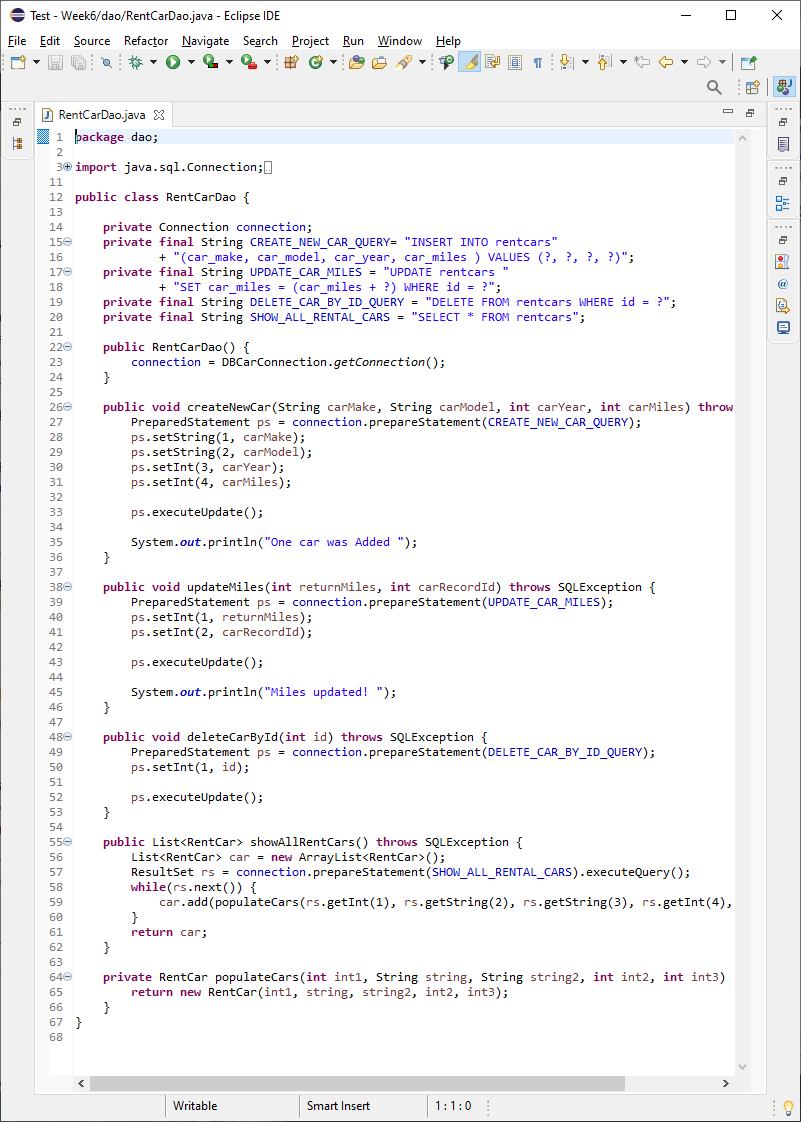
CustomerCarDao.java



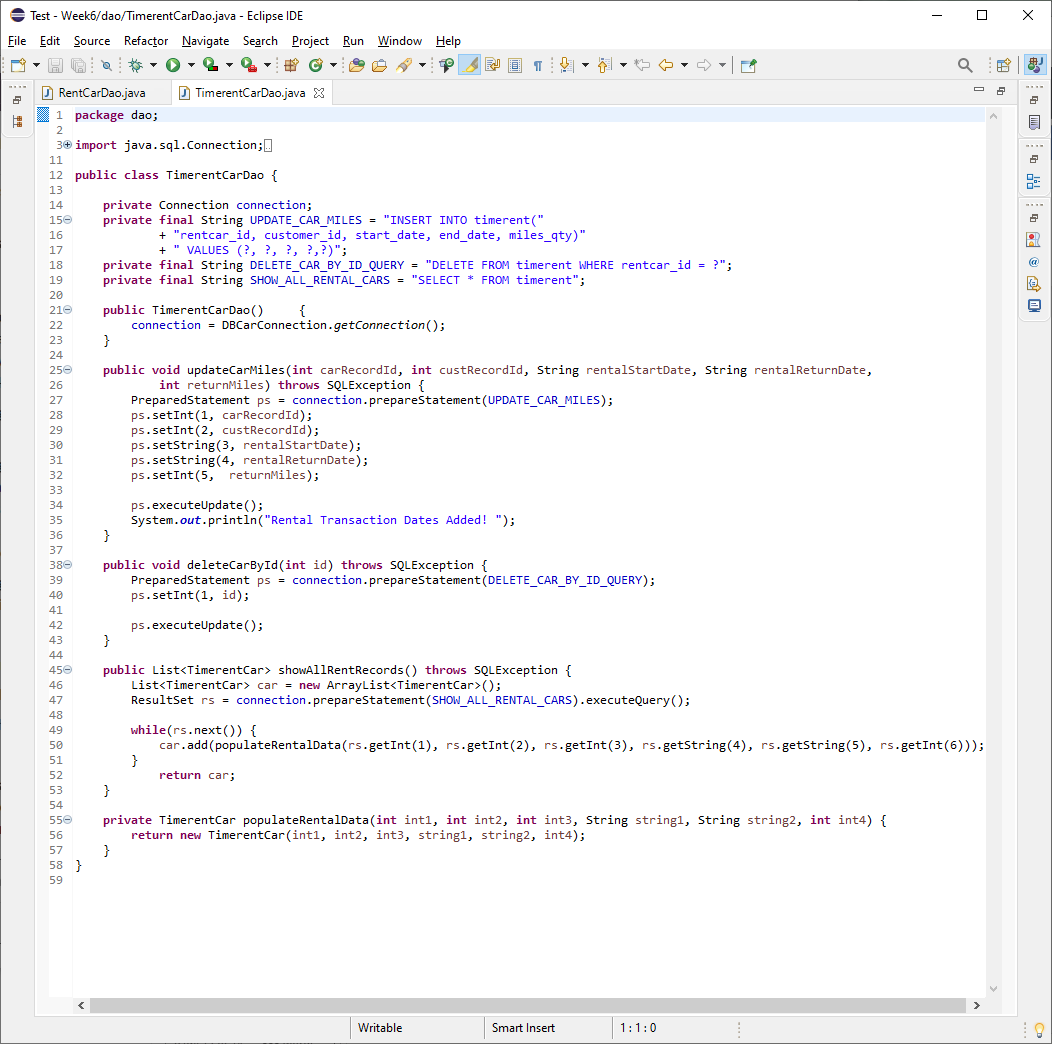
DBCarConnection.java



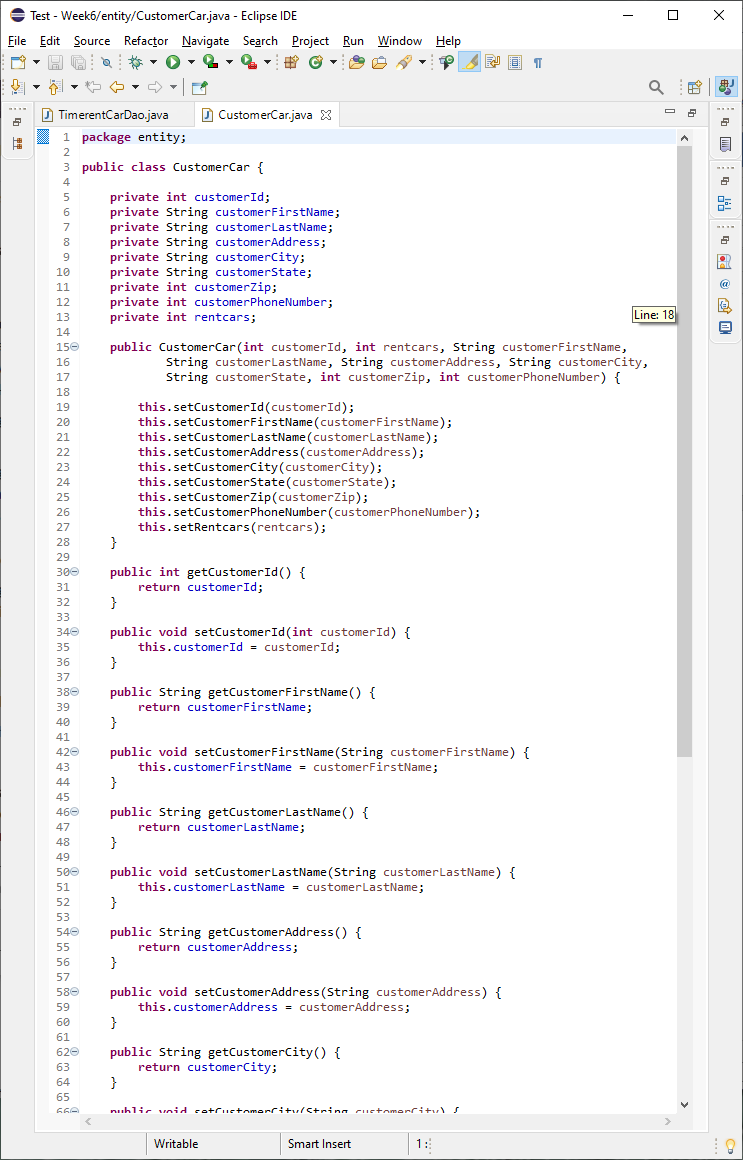
RentCarDao.java



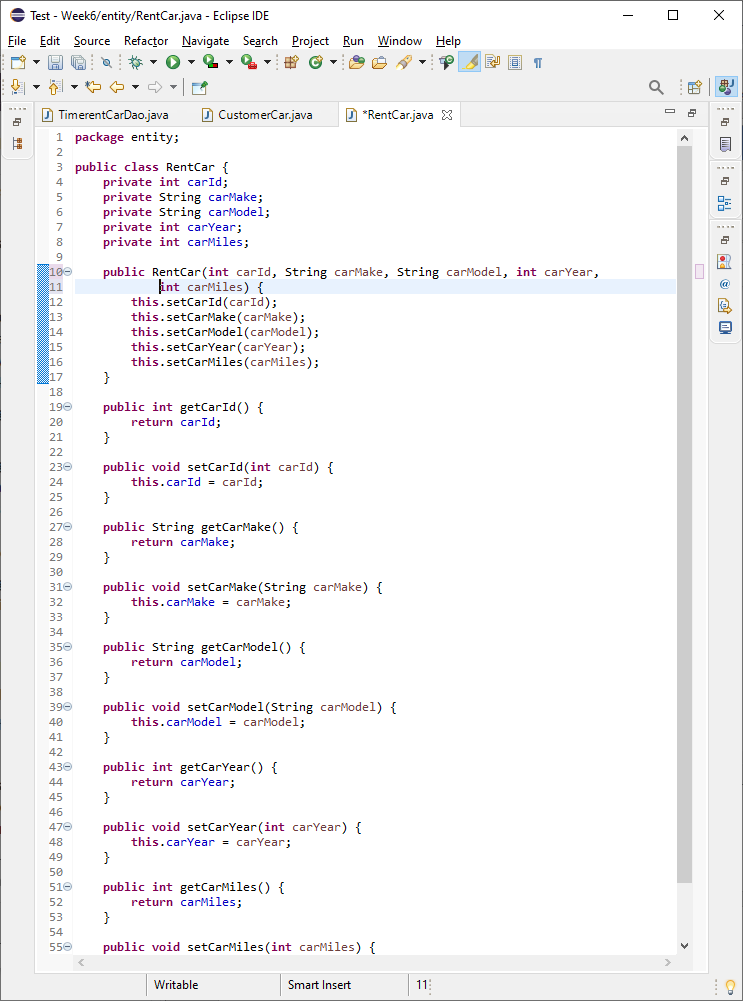
TimerentCarDao.java

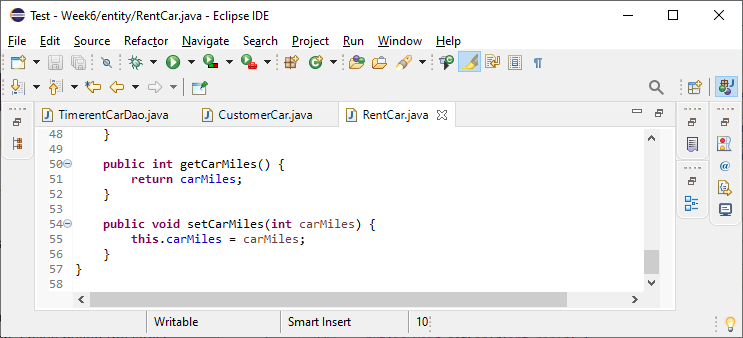


CustomerCar.java

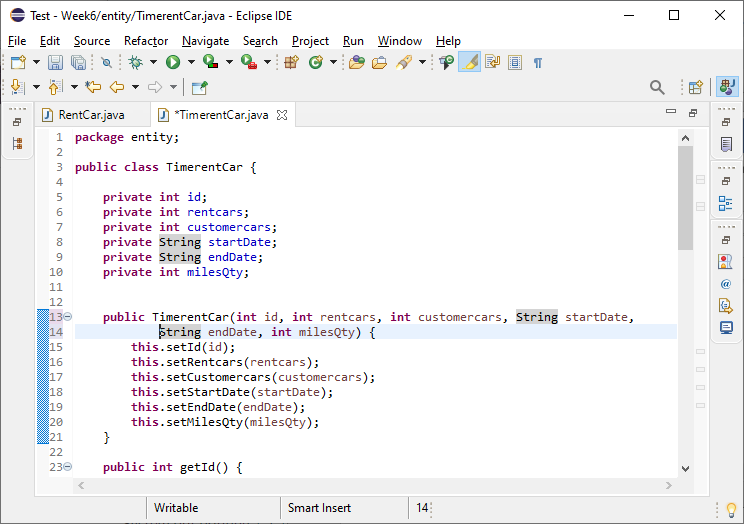


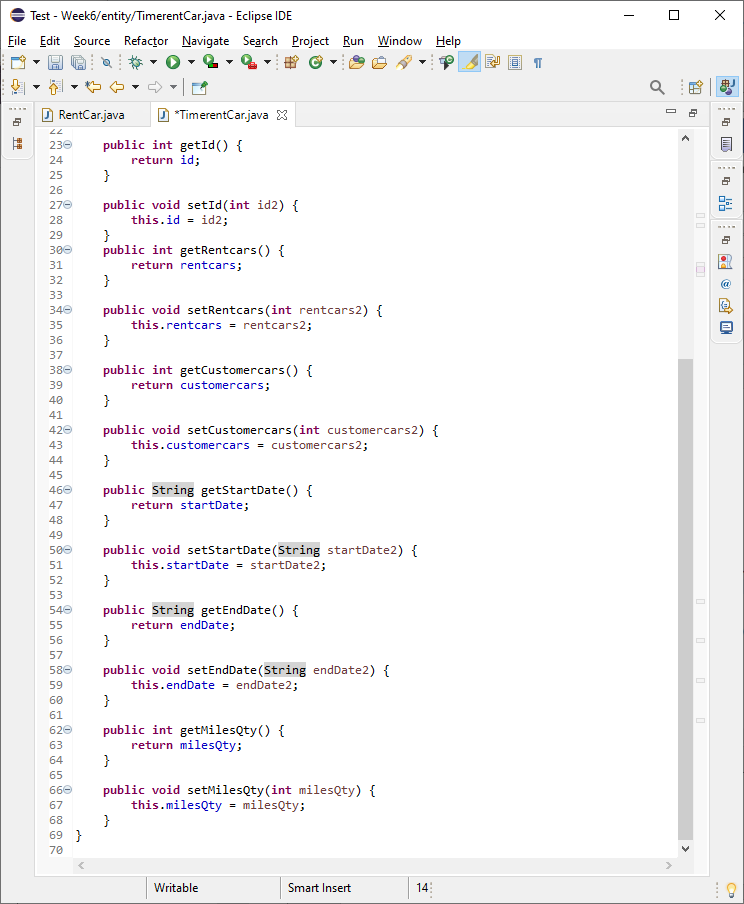
RentCar.java



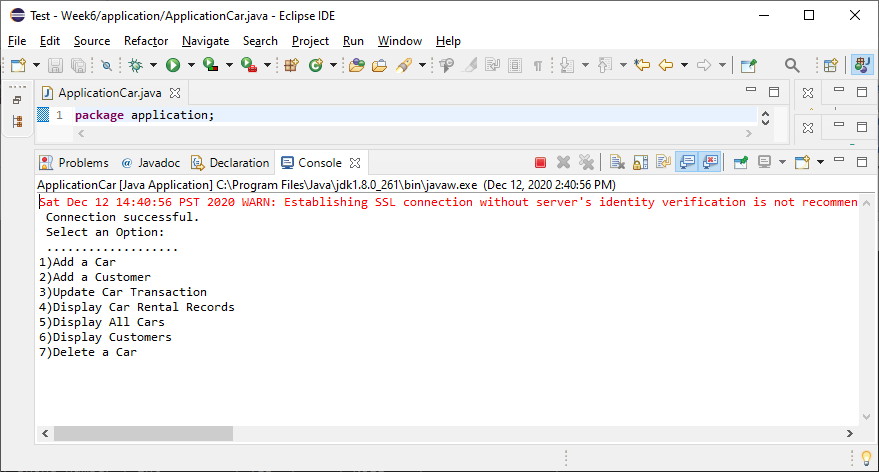


TimerentCar.java

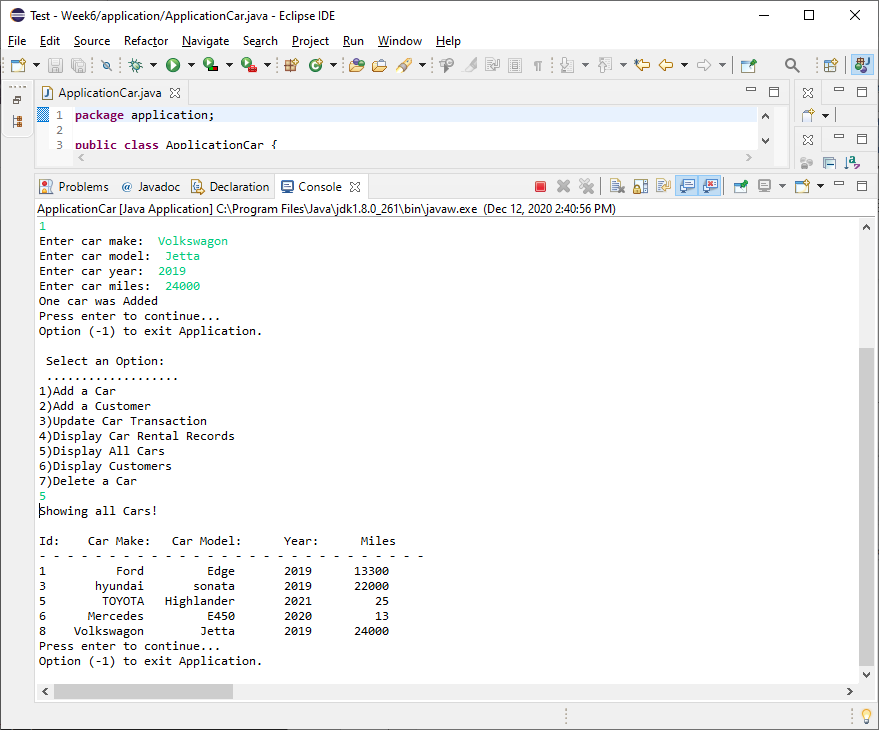




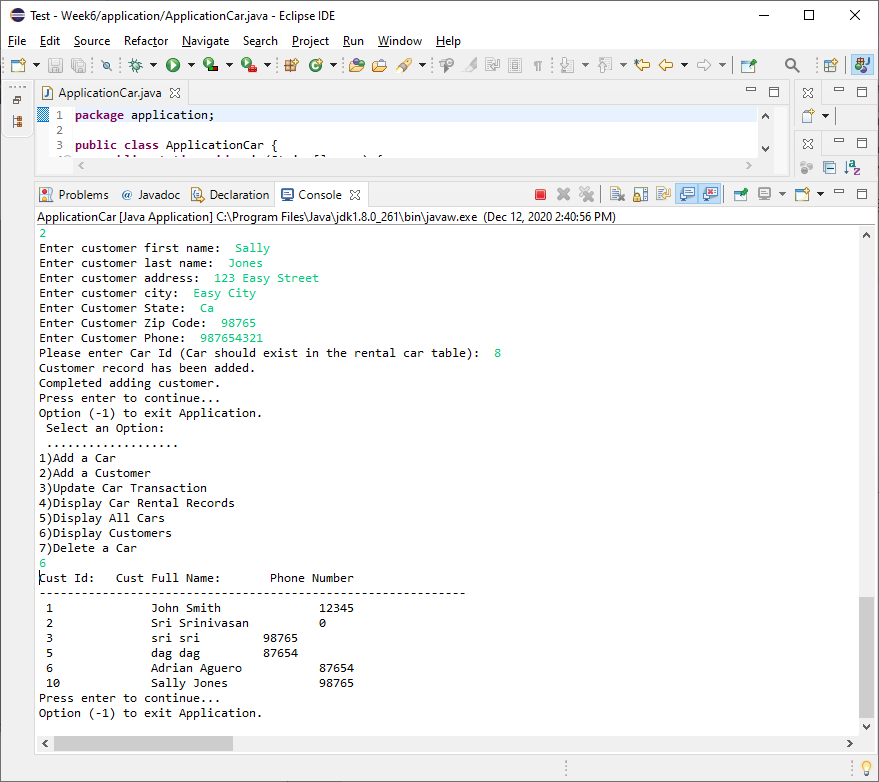
**Screenshots of Running Application:**



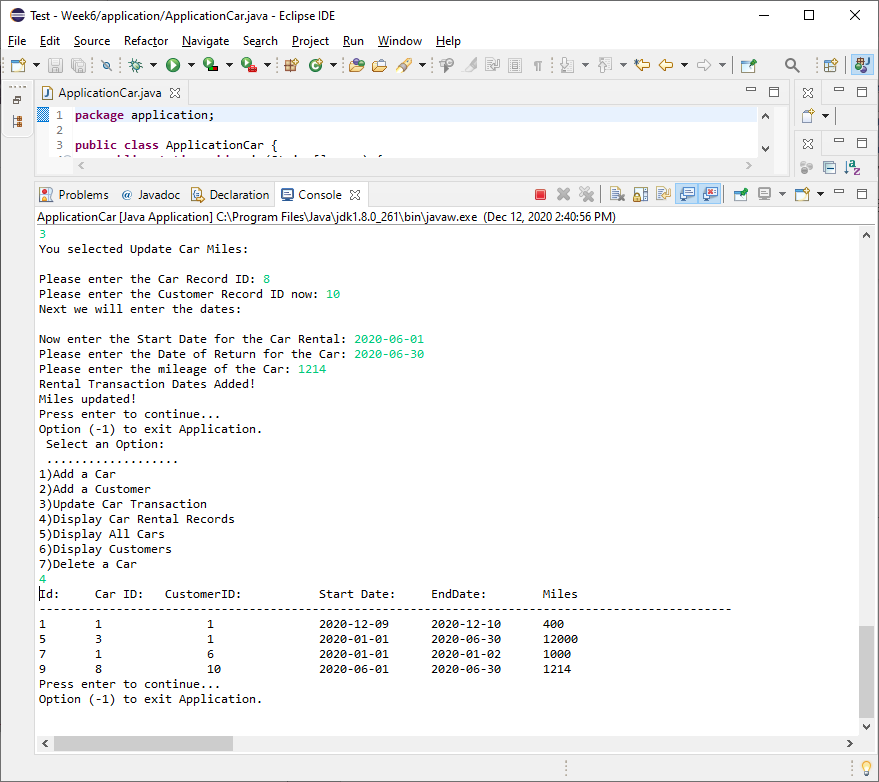
**Option1**. Add a Car & **Option 5:** Display All Cars



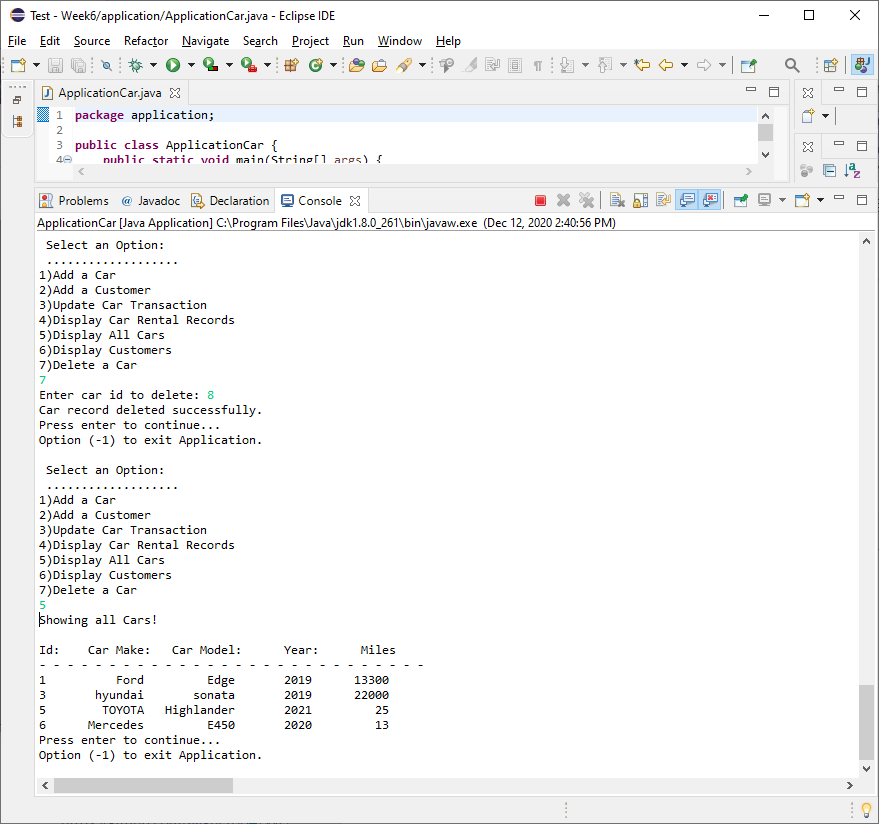
**Option2:** Add a Customer & **Option 6:** Display Customers



**Option 3:** Update Car Transaction & **Option 4:** Display Car Rental Records



**Option 7:** Delete a Car



**URL to GitHub Repository:**

**My repository:** [**https://github.com/srikripa/MSQL-Repos-W6.git**](https://github.com/srikripa/MSQL-Repos-W6.git)

**Project Team repository:** [**https://github.com/aaguero64/Week6.git**](https://github.com/aaguero64/Week6.git)