# 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:**

*package* com.promineo.week11.Dao;  
  
*import* com.promineo.week11.Models.Car;  
  
*import* java.sql.CallableStatement;  
*import* java.sql.Connection;  
*import* java.sql.ResultSet;  
*import* java.sql.SQLException;  
*import* java.util.ArrayList;  
  
*public class* CarDao *extends* MySqlConnection *implements* ICarDao {  
 *private* Connection connection;  
 *public* CarDao() {  
 connection = *super*.getConnection("cars");  
 }  
  
 @Override  
 *public* ArrayList<Car> getCars() {  
 ArrayList<Car> cars = *new* ArrayList<Car>();  
  
 *try* {  
 CallableStatement st = connection.prepareCall("{call getCars()}");  
 *boolean* hasResult = st.execute();  
 *if*(hasResult) {  
 ResultSet rs = st.getResultSet();  
  
 *while*(rs.next()) {  
 cars.add(*new* Car(rs.getInt(1), rs.getString(2), rs.getString(3)));  
 }  
 }  
 } *catch* (SQLException ex) {  
 System.out.println(ex.getMessage());  
 *return* cars;  
 }  
 *return* cars;  
 }  
  
 @Override  
 *public void* createCar(Car car) {  
 *try* {  
 CallableStatement st = connection.prepareCall("{call addCar(?, ?)}");  
 st.setString(1, car.getMake());  
 st.setString(2, car.getModel());  
 st.execute();  
 }  
 *catch*(SQLException ex)  
 {  
 System.out.println(ex.getMessage());  
 }  
 }  
  
 @Override  
 *public void* deleteCar(Car car) {  
 *try* {  
 CallableStatement st = connection.prepareCall("{call deleteCar(?)}");  
 st.setInt(1, car.getCarId());  
 st.execute();  
 } *catch* (SQLException ex) {  
 System.out.println(ex.getMessage());  
 }  
 }  
  
 @Override  
 *public void* updateCar(Car car) {  
 *try* {  
 CallableStatement st = connection.prepareCall("{call updateCar(?,?,?)}");  
 st.setInt(1, car.getCarId());  
 st.setString(2, car.getMake());  
 st.setString(3, car.getModel());  
 st.execute();  
 } *catch*(SQLException ex) {  
 System.out.println(ex.getMessage());  
 }  
 }  
}

*package* com.promineo.week11.Dao;  
  
*import* com.promineo.week11.Models.Car;  
  
*import* java.util.ArrayList;  
  
*public interface* ICarDao {  
 ArrayList<Car> getCars();  
 *void* createCar(Car car);  
 *void* deleteCar(Car car);  
 *void* updateCar(Car car);  
}

*package* com.promineo.week11.Dao;  
  
*import* java.sql.Connection;  
*import* java.sql.DriverManager;  
*import* java.sql.SQLException;  
  
*public class* MySqlConnection {  
 *private* Connection conn;  
 *private final* String Username = "root";  
 *private final* String Password = "31Ack1ag00n";  
  
 *public* Connection getConnection(String dbName) {  
 *try* {  
 String connString = "jdbc:mysql://localhost:3306/" + dbName;  
 *if* (conn == *null* || conn.isClosed()) {  
 conn = DriverManager.*getConnection*(connString, Username, Password);  
 System.out.println("dbConnected");  
 }  
  
 *return* conn;  
 }  
 *catch*(SQLException ex)  
 {  
 System.out.println(ex.getMessage());  
 *return* conn;  
 }  
 }  
}

*package* com.promineo.week11.Models;  
  
*public class* Car {  
  
 *private* String make;  
 *private* String model;  
  
 *public* Car(*int* carId, String make, String model) {  
 *this*.carId = carId;  
 *this*.make = make;  
 *this*.model = model;  
 }  
  
 *public* Car(String make, String model) {  
 *this*.make = make;  
 *this*.model = model;  
 }  
  
 *public* Car(*int* carId) {  
 *this*.carId = carId;  
 }  
  
 *public int* getCarId() {  
 *return* carId;  
 }  
  
 *public void* setCarId(*int* carId) {  
 *this*.carId = carId;  
 }  
  
 *private int* 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;  
 }  
  
}

*package* com.promineo.week11.Service;  
  
*import* com.promineo.week11.Dao.CarDao;  
*import* com.promineo.week11.Models.Car;  
  
*import* java.util.ArrayList;  
  
*public class* CarService *implements* ICarService{  
 *private static* CarDao *carDao*;  
  
 *public* CarService() {  
 *carDao* = *new* CarDao();  
 }  
  
  
 @Override  
 *public* ArrayList<Car> getCars() {  
 *return carDao*.getCars();  
 }  
  
 @Override  
 *public void* createCar(Car car) {  
 *carDao*.createCar(car);  
 }  
  
 @Override  
 *public void* deleteCar(Car car) {  
 *carDao*.deleteCar(car);  
 }  
  
 @Override  
 *public void* updateCar(Car car) {  
 *carDao*.updateCar(car);  
 }  
  
  
}

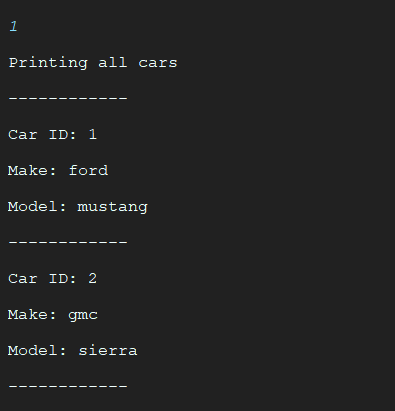
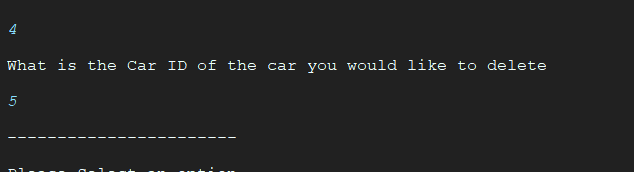
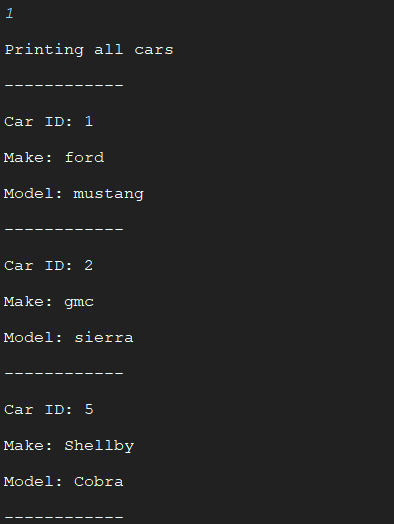
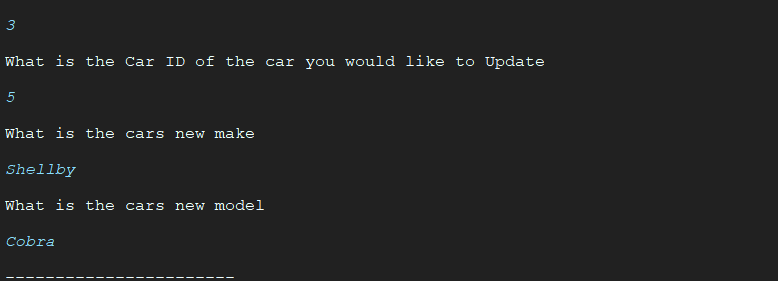
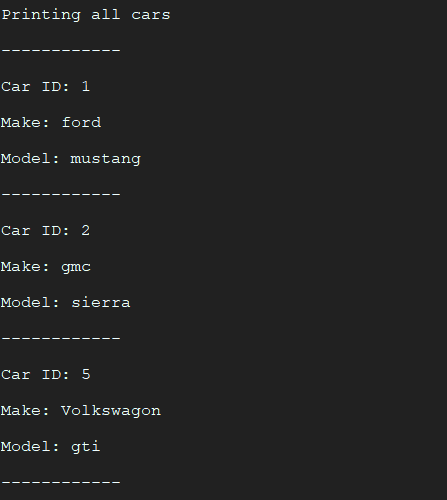
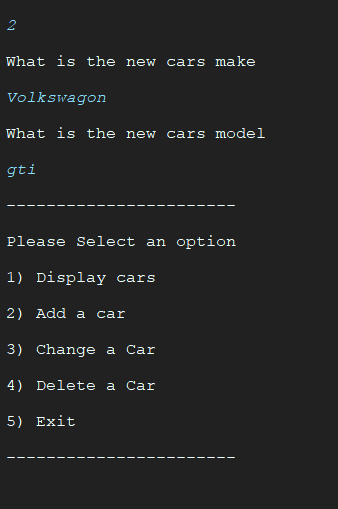
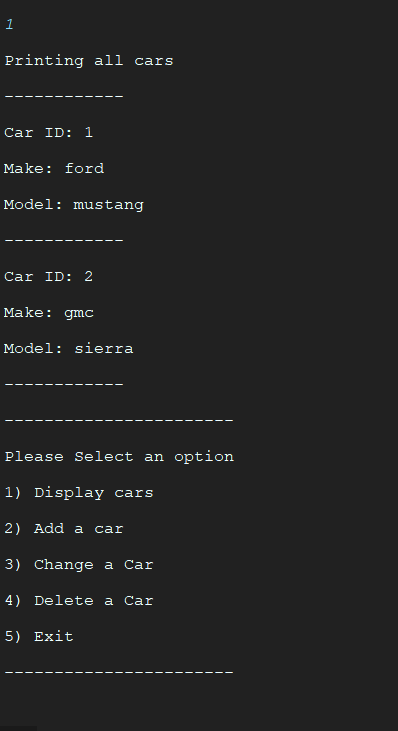
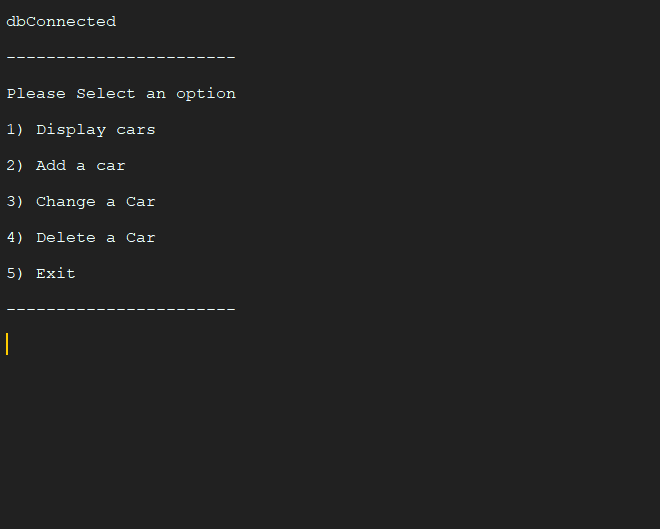
*package* com.promineo.week11.Service;  
  
*import* com.promineo.week11.Models.Car;  
  
*import* java.util.ArrayList;  
  
*public interface* ICarService {  
 ArrayList<Car> getCars();  
 *void* createCar(Car car);  
 *void* deleteCar(Car car);  
 *void* updateCar(Car car);  
}

*package* com.promineo.week11.Views;  
  
*import* com.promineo.week11.Models.Car;  
*import* com.promineo.week11.Service.CarService;  
  
*import* java.util.ArrayList;  
*import* java.util.Scanner;  
  
*public class* Menu {  
 CarService carService = *new* CarService();  
 *private* Scanner scanner = *new* Scanner(System.in);  
 *public void* start() {  
 String selection = "";  
  
 *do* {  
 printMenu();  
 selection = scanner.nextLine();  
 *if* (selection.equals("1")) {  
 displayCars();  
 } *else if* (selection.equals("2")) {  
 addCar();  
 } *else if* (selection.equals("3")) {  
 updateCar();  
 } *else if* (selection.equals("4")) {  
 deleteCar();  
 } *else if* (!selection.equals("5")) {  
 System.out.println("Please select a number 1 - 5");  
 }  
 } *while* (!selection.equals("5"));  
 }  
  
 *private void* deleteCar() {  
 System.out.println("What is the Car ID of the car you would like to delete");  
 *int* id = scanner.nextInt();  
 carService.deleteCar(*new* Car(id));  
 }  
  
 *private void* updateCar() {  
 System.out.println("What is the Car ID of the car you would like to Update");  
 *int* id = scanner.nextInt();  
 scanner.nextLine();  
 System.out.println("What is the cars new make");  
 String make = scanner.nextLine();  
 System.out.println("What is the cars new model");  
 String model = scanner.nextLine();  
 carService.updateCar(*new* Car(id, make, model));  
 }  
  
 *private void* addCar() {  
 System.out.println("What is the new cars make");  
 String make = scanner.nextLine();  
 System.out.println("What is the new cars model");  
 String model = scanner.nextLine();  
 carService.createCar(*new* Car(make, model));  
 }  
  
 *private void* displayCars() {  
 System.out.println("Printing all cars");  
 System.out.println("------------");  
 ArrayList<Car> cars = carService.getCars();  
 *for* (Car car : cars) {  
 System.out.println("Car ID: " + car.getCarId());  
 System.out.println("Make: " + car.getMake());  
 System.out.println("Model: " + car.getModel());  
 System.out.println("------------");  
 }  
 }  
  
 *private void* printMenu() {  
 System.out.println("-----------------------");  
 System.out.println("Please Select an option");  
 System.out.println("1) Display cars");  
 System.out.println("2) Add a car");  
 System.out.println("3) Change a Car");  
 System.out.println("4) Delete a Car");  
 System.out.println("5) Exit");  
 System.out.println("-----------------------");  
 }  
}

*package* com.promineo.week11.Views;  
  
*public class* Application {  
  
 *public static void* main(String[] args) {  
 Menu menu = *new* Menu();  
 menu.start();  
 }  
}

*create* database cars;  
  
use cars;  
  
*create table* car (  
 carId *int not null* auto\_increment *primary key*,  
 make *varchar*(50),  
 model *varchar*(50)  
 );  
   
delimiter //  
*-- Select all cars in the car table  
create procedure* getCars()  
*begin  
 select* \* *from* car;  
*end*//  
delimiter ;  
  
delimiter //  
*-- Select all cars in the car table  
create procedure* addCar(*in* make *varchar*(50), *in* model *varchar*(50))  
*begin  
 insert into* car (make, model)  
 *values*(make, model);  
*end*//  
delimiter ;  
  
delimiter //  
*-- Select all cars in the car table  
create procedure* updateCar(*in* id *int*, *in* make *varchar*(50), *in* model *varchar*(50))  
*begin  
 update* car  
 *set* make = make,  
 model = model  
 *where* carId = id;  
*end*//  
delimiter ;  
  
delimiter //  
*-- Select all cars in the car table  
create procedure* deleteCar(*in* id *int*)  
*begin  
 delete from* car  
 *where* carId = id;  
*end*//  
delimiter ;

**Screenshots of Running Application:**

****

**URL to GitHub Repository:**

[**https://github.com/jvgreen/Promineo-Boot-Camp/tree/master/mysql/week5/coding05/src**](https://github.com/jvgreen/Promineo-Boot-Camp/tree/master/mysql/week5/coding05/src)