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 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:

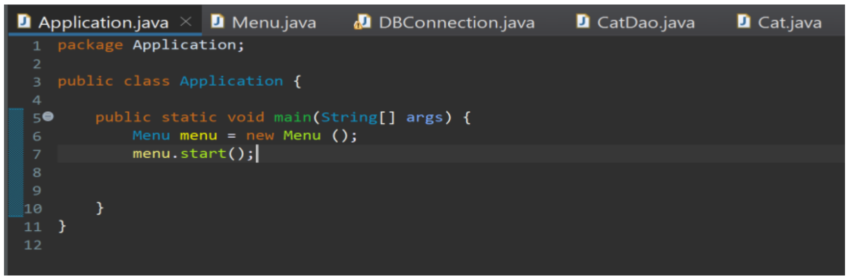
The application does not need to be as complex as the example in the video curriculum.

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

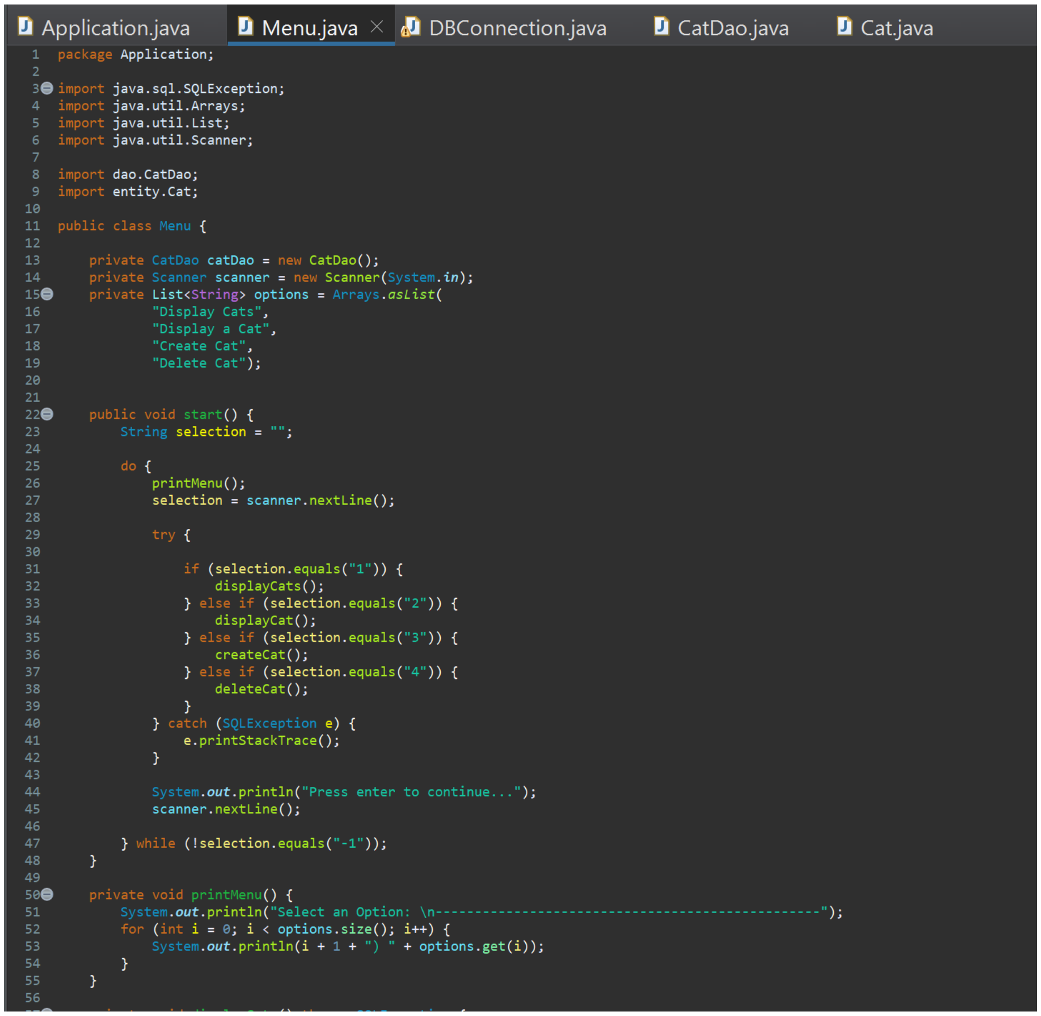
Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

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

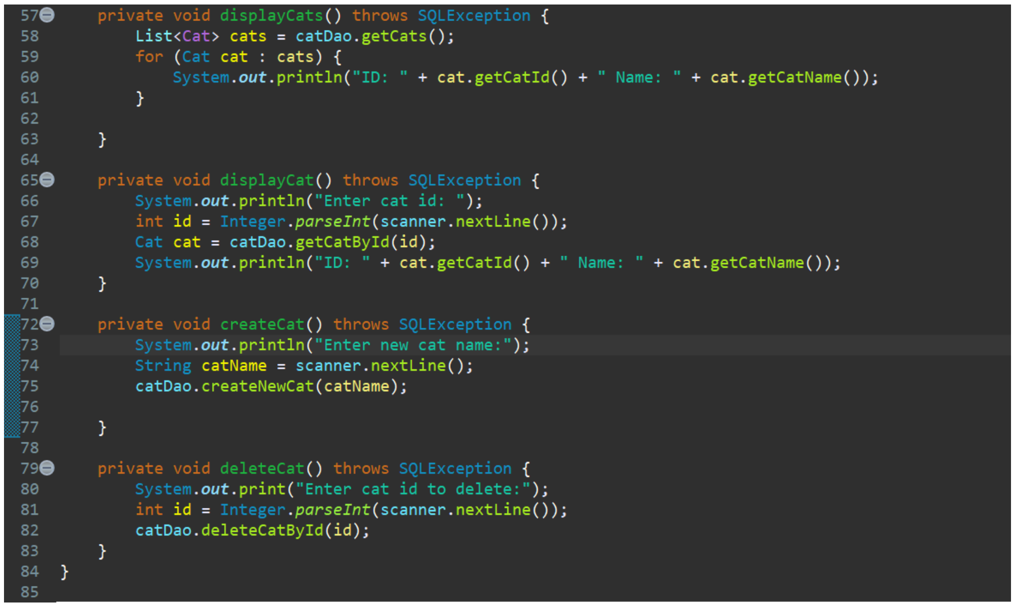
Screenshots of Code:



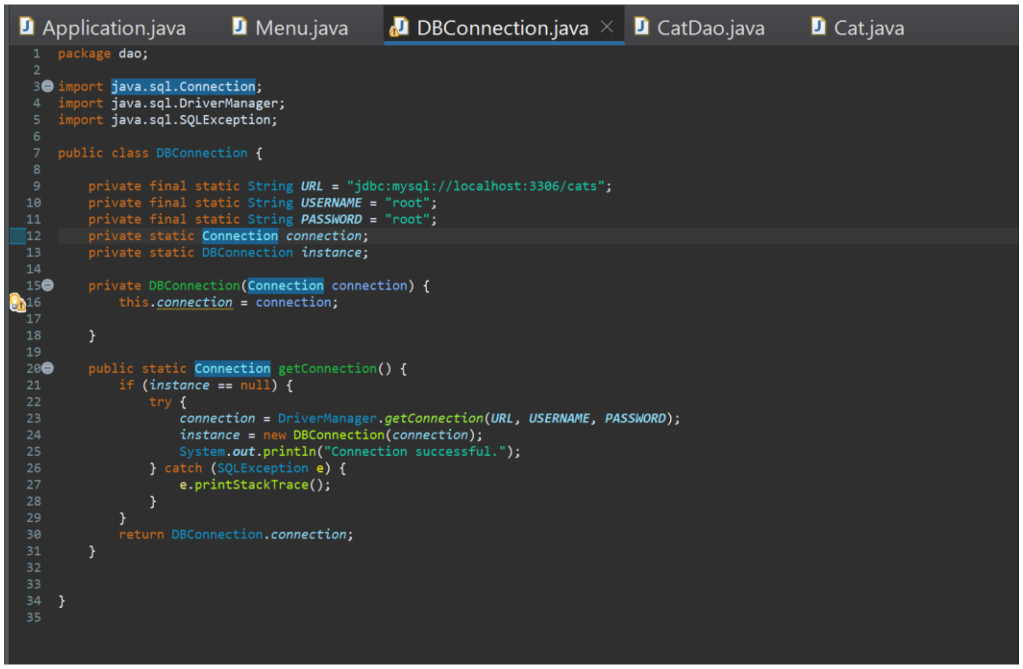
```java
package Application;

public class Application {

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


    }
}

```



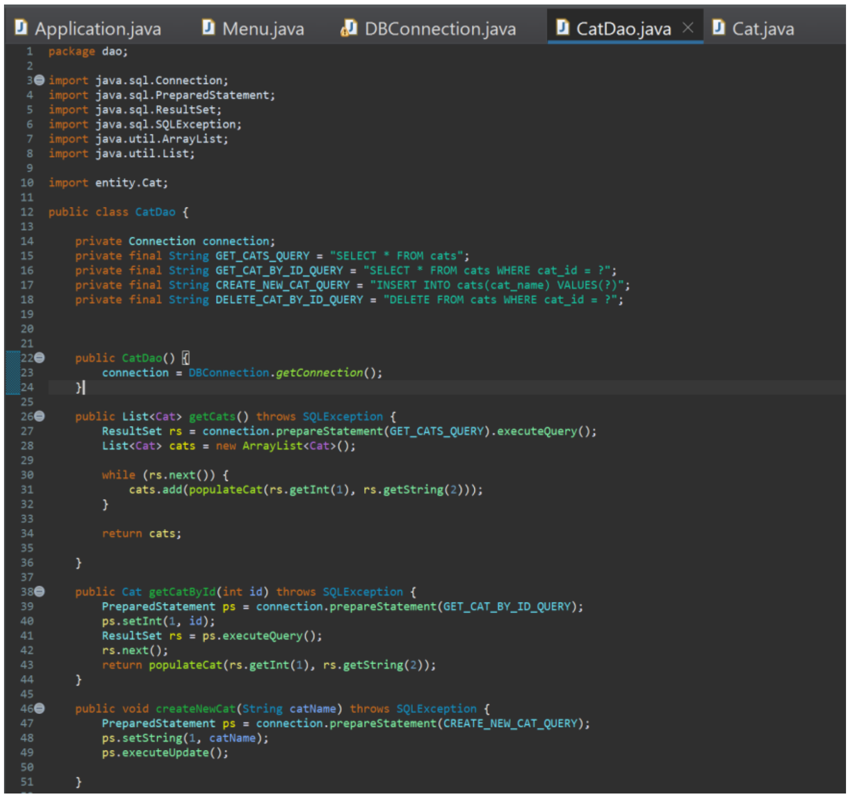
```java
package Application;

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

import dao.CatDao;
import entity.Cat;

public class Menu {

    private CatDao catDao = new CatDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
            "Display Cats",
            "Display a Cat",
            "Create Cat",
            "Delete Cat");


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

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

            try {

                if (selection.equals("1")) {
                    displayCats();
                } else if (selection.equals("2")) {
                    displayCat();
                } else if (selection.equals("3")) {
                    createCat();
                } else if (selection.equals("4")) {
                    deleteCat();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }

            System.out.println("Press enter to continue...");
            scanner.nextLine();

        } while (!selection.equals("-1"));
    }

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

```



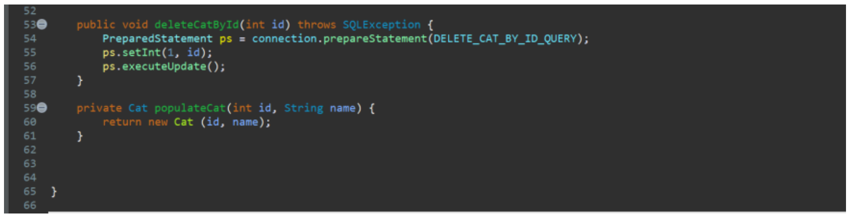
```java
    private void displayCats() throws SQLException {
        List<Cat> cats = catDao.getCats();
        for (Cat cat : cats) {
            System.out.println("ID: " + cat.getCatId() + " Name: " + cat.getCatName());
        }

    }

    private void displayCat() throws SQLException {
        System.out.println("Enter cat id: ");
        int id = Integer.parseInt(scanner.nextLine());
        Cat cat = catDao.getCatById(id);
        System.out.println("ID: " + cat.getCatId() + " Name: " + cat.getCatName());
    }

    private void createCat() throws SQLException {
        System.out.println("Enter new cat name:");
        String catName = scanner.nextLine();
        catDao.createNewCat(catName);

    }

    private void deleteCat() throws SQLException {
        System.out.print("Enter cat id to delete:");
        int id = Integer.parseInt(scanner.nextLine());
        catDao.deleteCatById(id);
    }
}
```



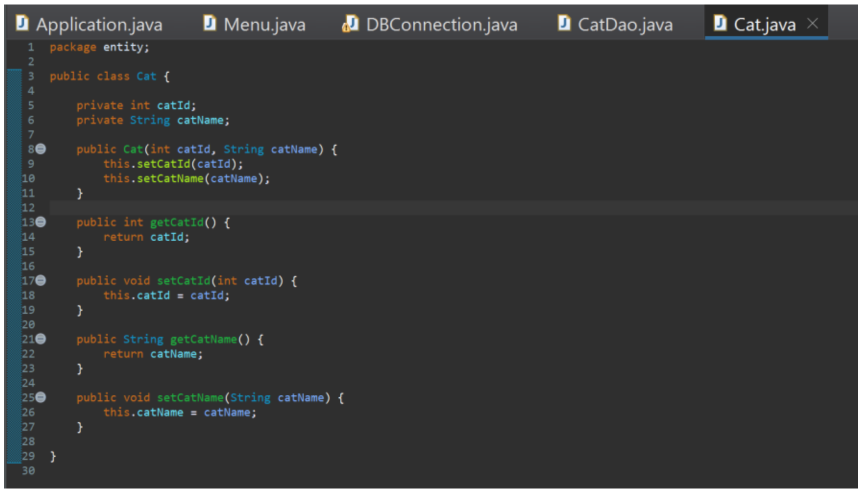
```java
package dao;

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

public class DBConnection {

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

    private DBConnection(Connection connection) {
        this.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;
    }


}

```



```java
package dao;

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

import entity.Cat;

public class CatDao {

    private Connection connection;
    private final String GET_CATS_QUERY = "SELECT * FROM cats";
    private final String GET_CAT_BY_ID_QUERY = "SELECT * FROM cats WHERE cat_id = ?";
    private final String CREATE_NEW_CAT_QUERY = "INSERT INTO cats(cat_name) VALUES(?)";
    private final String DELETE_CAT_BY_ID_QUERY = "DELETE FROM cats WHERE cat_id = ?";



    public CatDao() {
        connection = DBConnection.getConnection();
    }

    public List<Cat> getCats() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_CATS_QUERY).executeQuery();
        List<Cat> cats = new ArrayList<Cat>();

        while (rs.next()) {
            cats.add(populateCat(rs.getInt(1), rs.getString(2)));
        }

        return cats;

    }

    public Cat getCatById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_CAT_BY_ID_QUERY);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populateCat(rs.getInt(1), rs.getString(2));
    }

    public void createNewCat(String catName) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_CAT_QUERY);
        ps.setString(1, catName);
        ps.executeUpdate();

    }
```

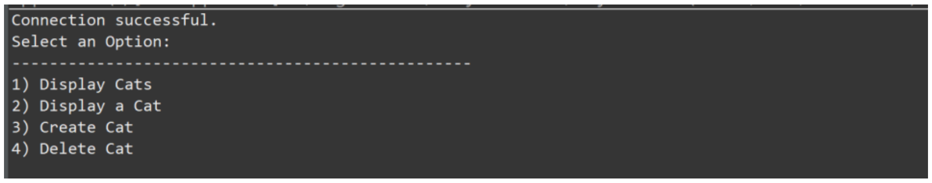


```java

    public void deleteCatById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_CAT_BY_ID_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }

    private Cat populateCat(int id, String name) {
        return new Cat (id, name);
    }



}

```

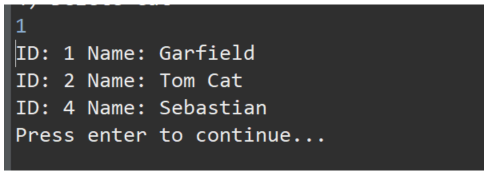


```java
package entity;

public class Cat {

    private int catId;
    private String catName;

    public Cat(int catId, String catName) {
        this.setCatId(catId);
        this.setCatName(catName);
    }

    public int getCatId() {
        return catId;
    }

    public void setCatId(int catId) {
        this.catId = catId;
    }

    public String getCatName() {
        return catName;
    }

    public void setCatName(String catName) {
        this.catName = catName;
    }

}
```

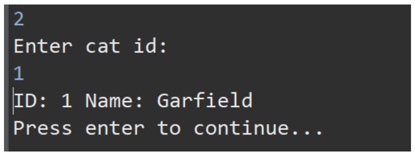
Screenshots of Running Application:



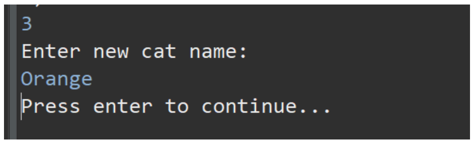
```
Connection successful.
Select an Option:
-------------------------------------------------
1) Display Cats
2) Display a Cat
3) Create Cat
4) Delete Cat
```



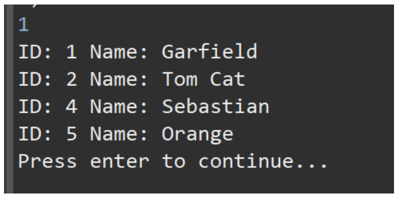
```
1
ID: 1 Name: Garfield
ID: 2 Name: Tom Cat
ID: 4 Name: Sebastian
Press enter to continue...
```



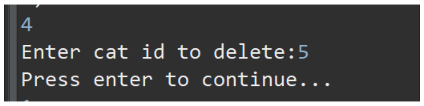
```
2
Enter cat id:
1
ID: 1 Name: Garfield
Press enter to continue...
```



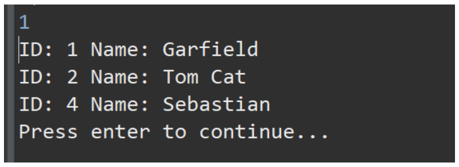
```
3
Enter new cat name:
Orange
Press enter to continue...
```



```
1
ID: 1 Name: Garfield
ID: 2 Name: Tom Cat
ID: 4 Name: Sebastian
ID: 5 Name: Orange
Press enter to continue...
```



```
4
Enter cat id to delete:5
Press enter to continue...
```



```
1
ID: 1 Name: Garfield
ID: 2 Name: Tom Cat
ID: 4 Name: Sebastian
Press enter to continue...
```

URL to GitHub Repository:

https://github.com/JoanaBarao7/MySQL_week4