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:

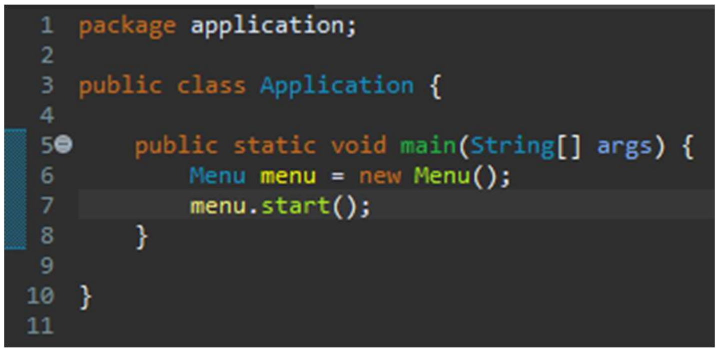
The application does not need to be as complex as the example in the video curriculum.

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

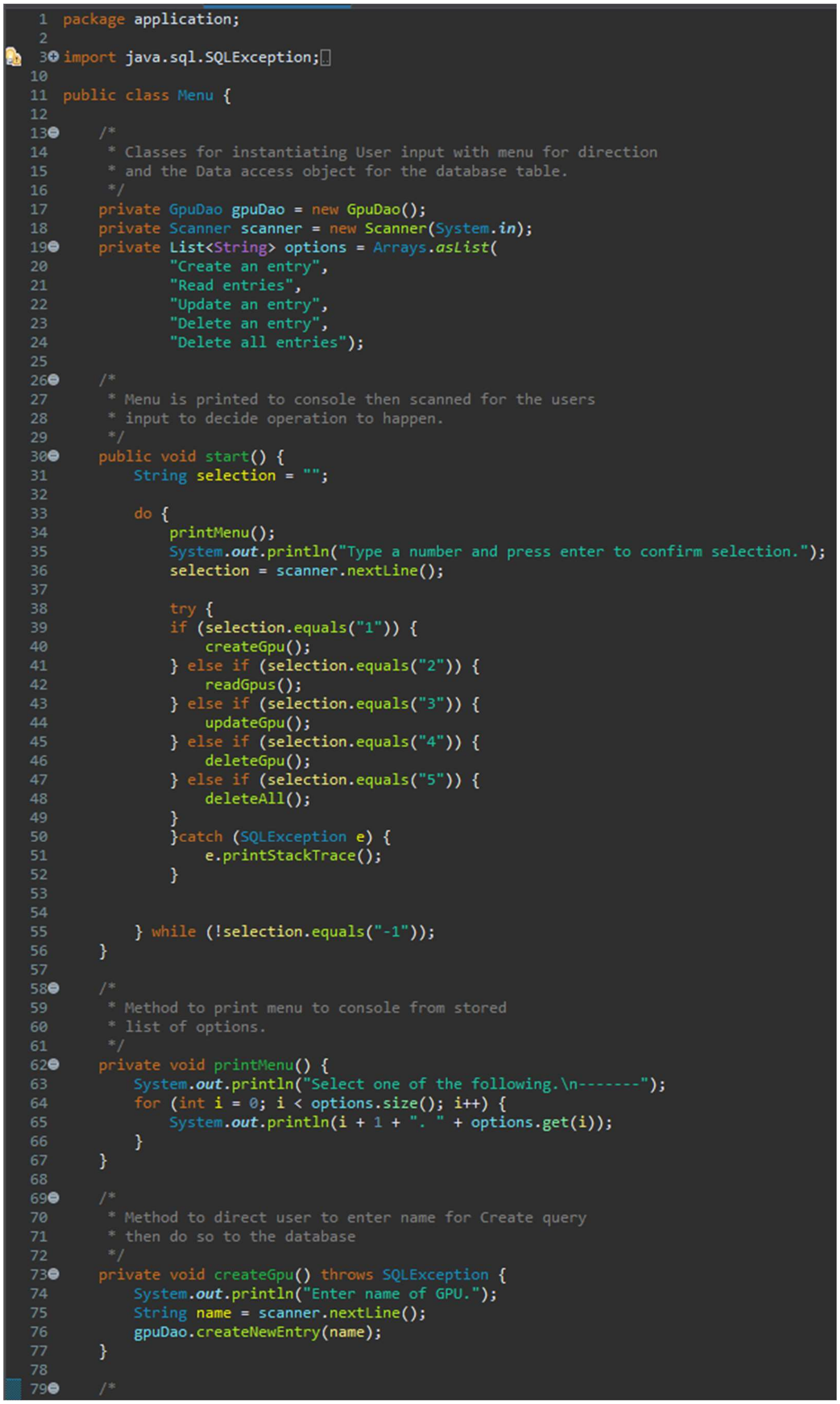
Remember that PreparedStatment.executeQuery() is only for Reading data and .executeUpdate() is used for Creating, Updating, and Deleting data.

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

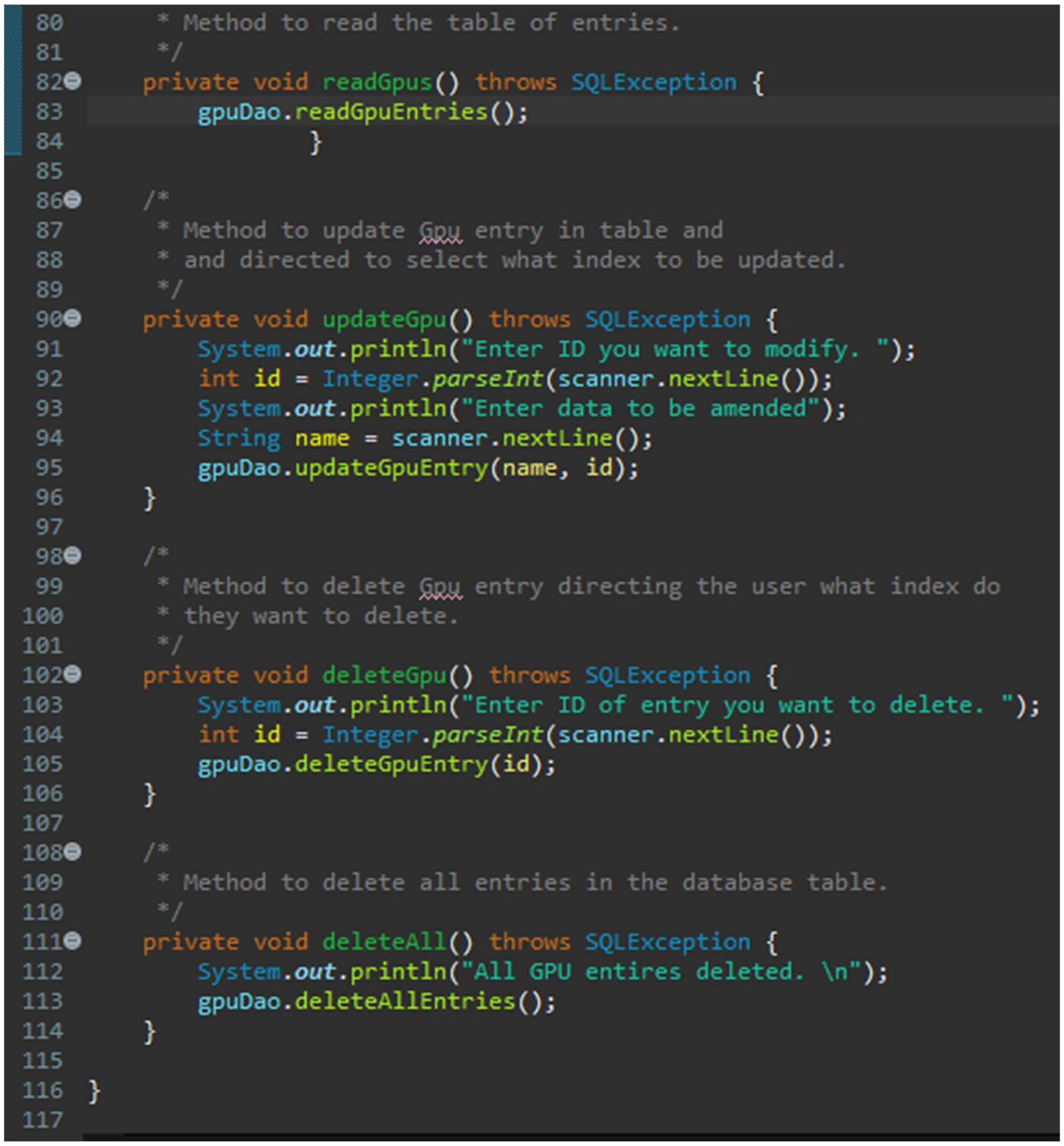
Screenshots of Code:



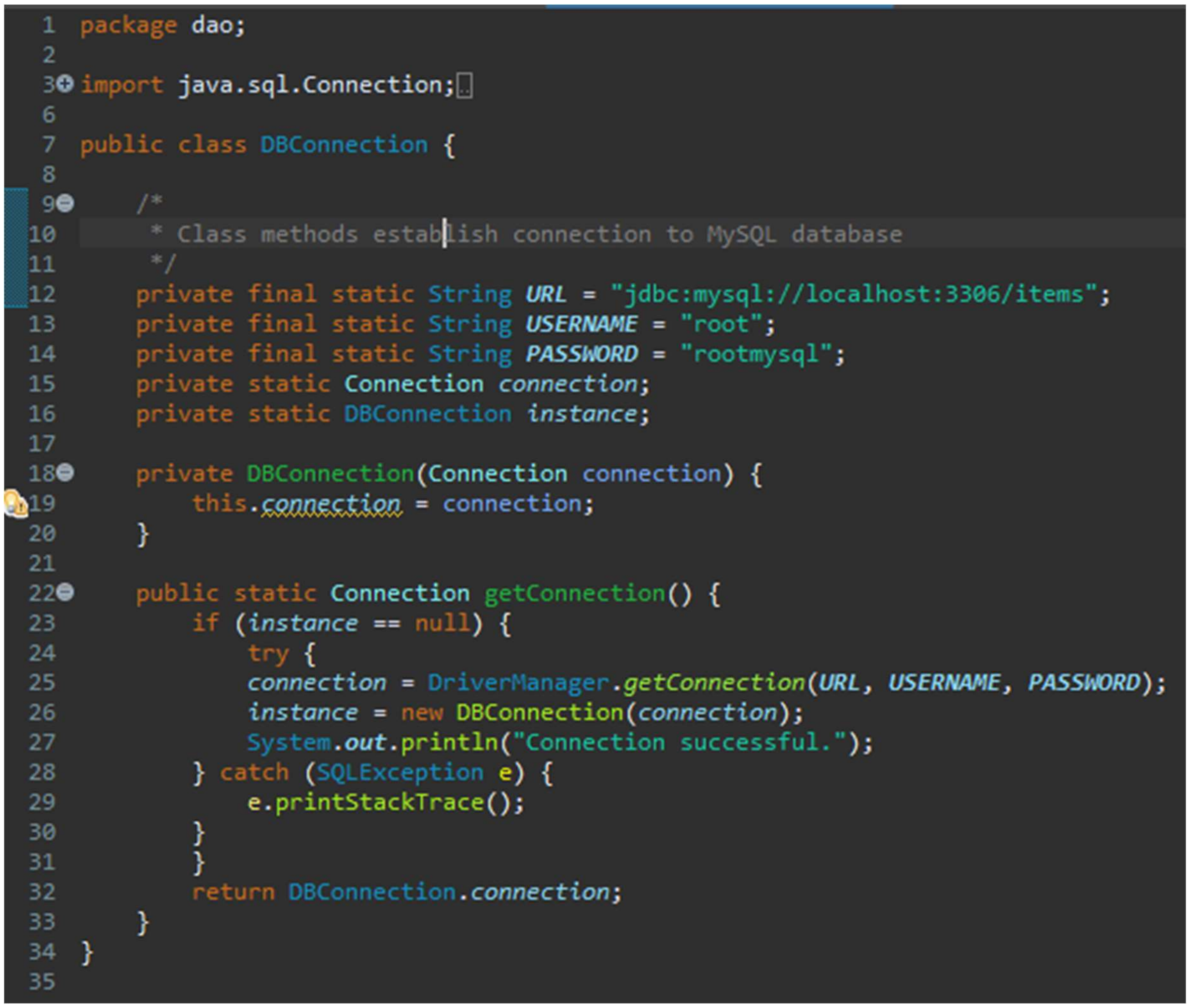
```java
package application;

public class Application {

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

}

```



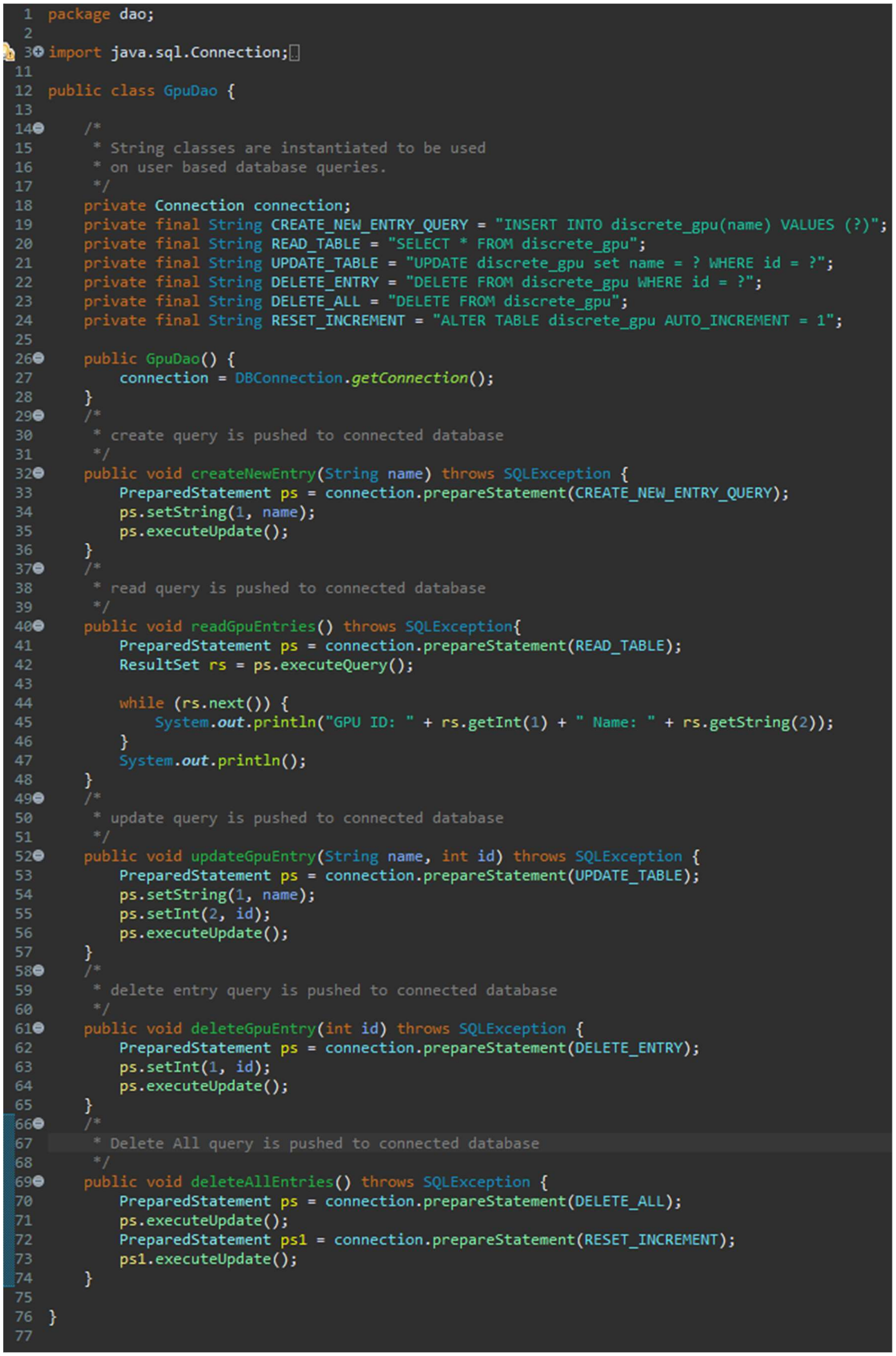
```java
package application;

import java.sql.SQLException;

public class Menu {

    /*
     * Classes for instantiating User input with menu for direction
     * and the Data access object for the database table.
     */
    private GpuDao gpuDao = new GpuDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
            "Create an entry",
            "Read entries",
            "Update an entry",
            "Delete an entry",
            "Delete all entries");

    /*
     * Menu is printed to console then scanned for the users
     * input to decide operation to happen.
     */
    public void start() {
        String selection = "";

        do {
            printMenu();
            System.out.println("Type a number and press enter to confirm selection.");
            selection = scanner.nextLine();

            try {
            if (selection.equals("1")) {
                createGpu();
            } else if (selection.equals("2")) {
                readGpus();
            } else if (selection.equals("3")) {
                updateGpu();
            } else if (selection.equals("4")) {
                deleteGpu();
            } else if (selection.equals("5")) {
                deleteAll();
            }
            }catch (SQLException e) {
                e.printStackTrace();
            }


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

    /*
     * Method to print menu to console from stored
     * list of options.
     */
    private void printMenu() {
        System.out.println("Select one of the following.\n-------");
        for (int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + ". " + options.get(i));
        }
    }

    /*
     * Method to direct user to enter name for Create query
     * then do so to the database
     */
    private void createGpu() throws SQLException {
        System.out.println("Enter name of GPU.");
        String name = scanner.nextLine();
        gpuDao.createNewEntry(name);
    }

    /*
```



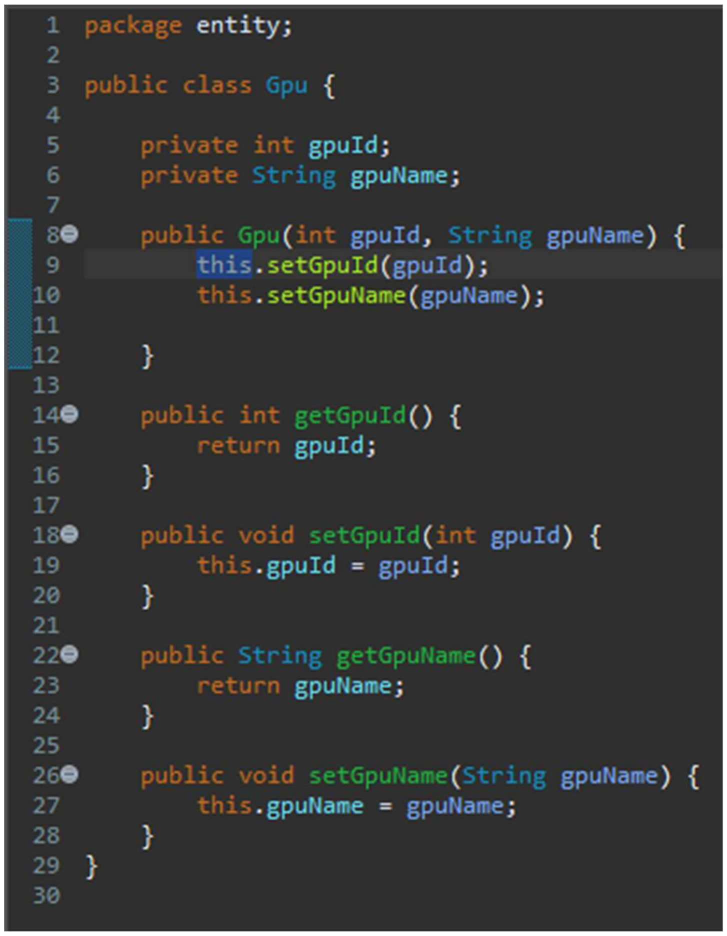
```java
     * Method to read the table of entries.
     */
    private void readGpus() throws SQLException {
        gpuDao.readGpuEntries();
            }

    /*
     * Method to update Gpu entry in table and
     * and directed to select what index to be updated.
     */
    private void updateGpu() throws SQLException {
        System.out.println("Enter ID you want to modify. ");
        int id = Integer.parseInt(scanner.nextLine());
        System.out.println("Enter data to be amended");
        String name = scanner.nextLine();
        gpuDao.updateGpuEntry(name, id);
    }

    /*
     * Method to delete Gpu entry directing the user what index do
     * they want to delete.
     */
    private void deleteGpu() throws SQLException {
        System.out.println("Enter ID of entry you want to delete. ");
        int id = Integer.parseInt(scanner.nextLine());
        gpuDao.deleteGpuEntry(id);
    }

    /*
     * Method to delete all entries in the database table.
     */
    private void deleteAll() throws SQLException {
        System.out.println("All GPU entires deleted. \n");
        gpuDao.deleteAllEntries();
    }

}
```



```java
package dao;

import java.sql.Connection;

public class DBConnection {

    /*
     * Class methods establish connection to MySQL database
     */
    private final static String URL = "jdbc:mysql://localhost:3306/items";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "rootmysql";
    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;
    }
}

```



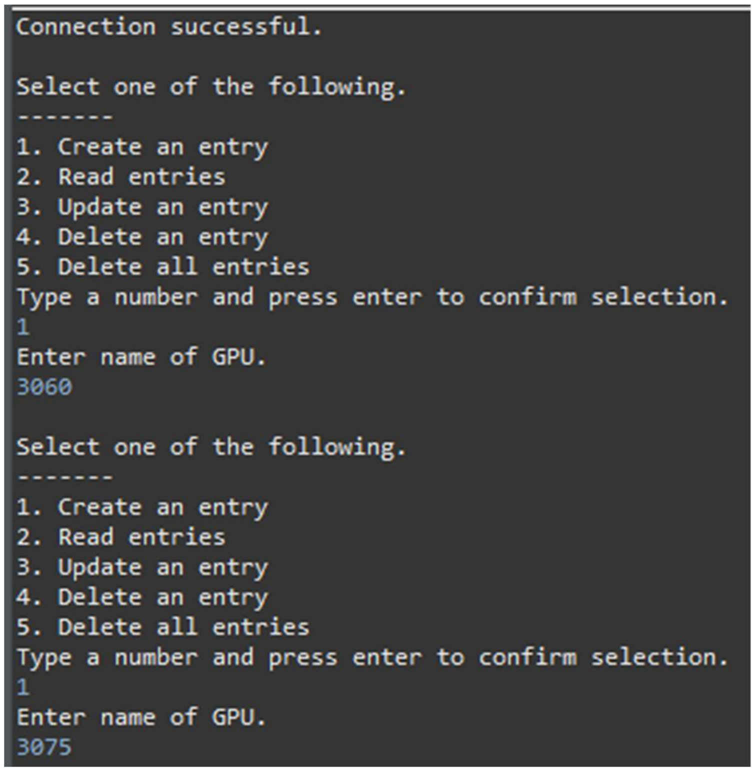
```java
package dao;

import java.sql.Connection;

public class GpuDao {

    /*
     * String classes are instantiated to be used
     * on user based database queries.
     */
    private Connection connection;
    private final String CREATE_NEW_ENTRY_QUERY = "INSERT INTO discrete_gpu(name) VALUES (?)";
    private final String READ_TABLE = "SELECT * FROM discrete_gpu";
    private final String UPDATE_TABLE = "UPDATE discrete_gpu set name = ? WHERE id = ?";
    private final String DELETE_ENTRY = "DELETE FROM discrete_gpu WHERE id = ?";
    private final String DELETE_ALL = "DELETE FROM discrete_gpu";
    private final String RESET_INCREMENT = "ALTER TABLE discrete_gpu AUTO_INCREMENT = 1";

    public GpuDao() {
        connection = DBConnection.getConnection();
    }
    /*
     * create query is pushed to connected database
     */
    public void createNewEntry(String name) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_ENTRY_QUERY);
        ps.setString(1, name);
        ps.executeUpdate();
    }
    /*
     * read query is pushed to connected database
     */
    public void readGpuEntries() throws SQLException{
        PreparedStatement ps = connection.prepareStatement(READ_TABLE);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            System.out.println("GPU ID: " + rs.getInt(1) + " Name: " + rs.getString(2));
        }
        System.out.println();
    }
    /*
     * update query is pushed to connected database
     */
    public void updateGpuEntry(String name, int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_TABLE);
        ps.setString(1, name);
        ps.setInt(2, id);
        ps.executeUpdate();
    }
    /*
     * delete entry query is pushed to connected database
     */
    public void deleteGpuEntry(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_ENTRY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }
    /*
     * Delete All query is pushed to connected database
     */
    public void deleteAllEntries() throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_ALL);
        ps.executeUpdate();
        PreparedStatement ps1 = connection.prepareStatement(RESET_INCREMENT);
        ps1.executeUpdate();
    }

}
```



```
package entity;

public class Gpu {

    private int gpuId;
    private String gpuName;

    public Gpu(int gpuId, String gpuName) {
        this.setGpuId(gpuId);
        this.setGpuName(gpuName);

    }

    public int getGpuId() {
        return gpuId;
    }

    public void setGpuId(int gpuId) {
        this.gpuId = gpuId;
    }

    public String getGpuName() {
        return gpuName;
    }

    public void setGpuName(String gpuName) {
        this.gpuName = gpuName;
    }
}

```

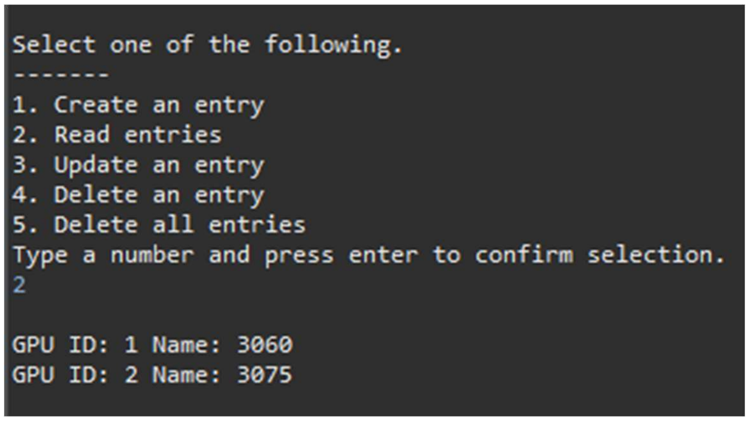
Screenshots of Running Application:

Create



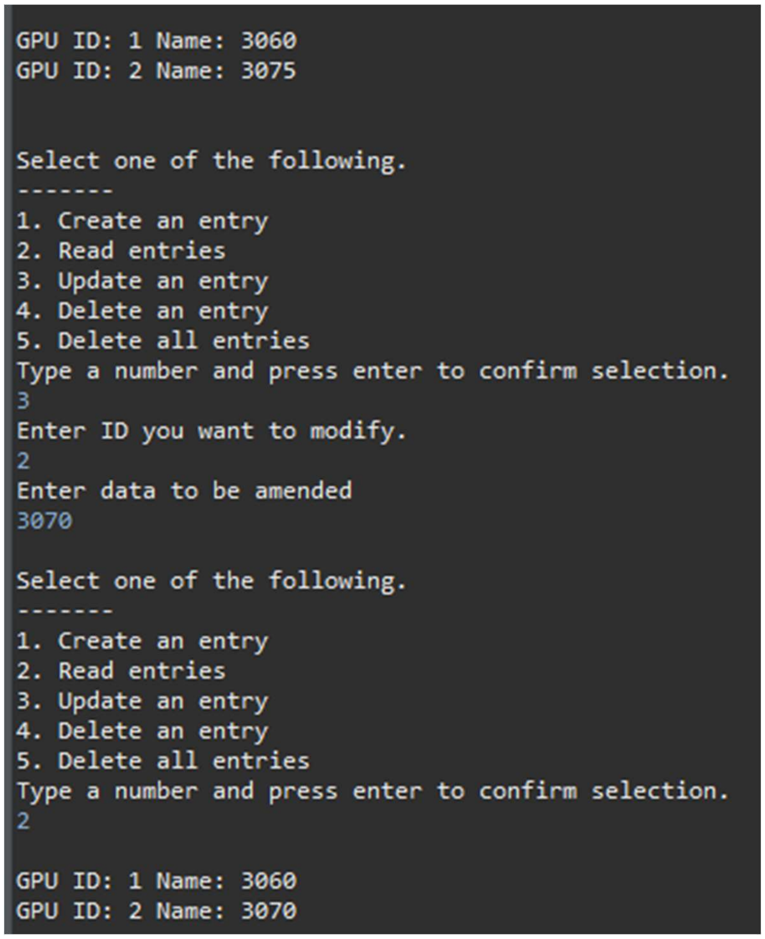
```
Connection successful.

Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
1
Enter name of GPU.
3060

Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
1
Enter name of GPU.
3075
```

Read



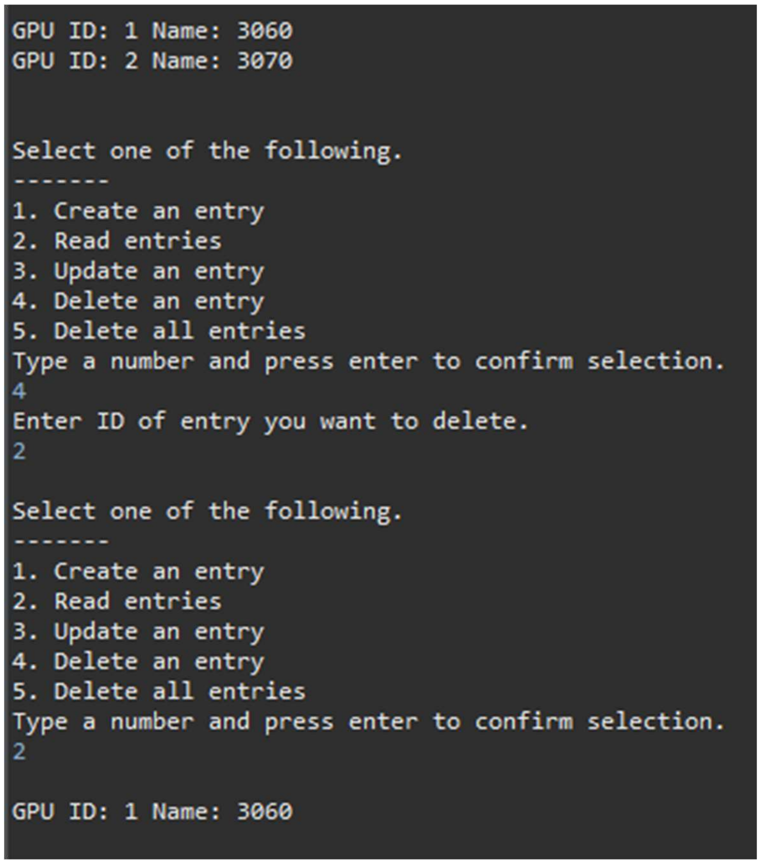
```

Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
2

GPU ID: 1 Name: 3060
GPU ID: 2 Name: 3075
```

Update



```
GPU ID: 1 Name: 3060
GPU ID: 2 Name: 3075


Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
3
Enter ID you want to modify.
2
Enter data to be amended
3070

Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
2

GPU ID: 1 Name: 3060
GPU ID: 2 Name: 3070
```

Delete



```
GPU ID: 1 Name: 3060
GPU ID: 2 Name: 3070


Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
4
Enter ID of entry you want to delete.
2

Select one of the following.
-------
1. Create an entry
2. Read entries
3. Update an entry
4. Delete an entry
5. Delete all entries
Type a number and press enter to confirm selection.
2

GPU ID: 1 Name: 3060
```

URL to GitHub Repository:

https://github.com/PierceIsaacson/week-10-MySQL-week4