GATEWAY COMMUNITY COLLEGE
A MARICOPA COMMUNITY COLLEGE

# 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.

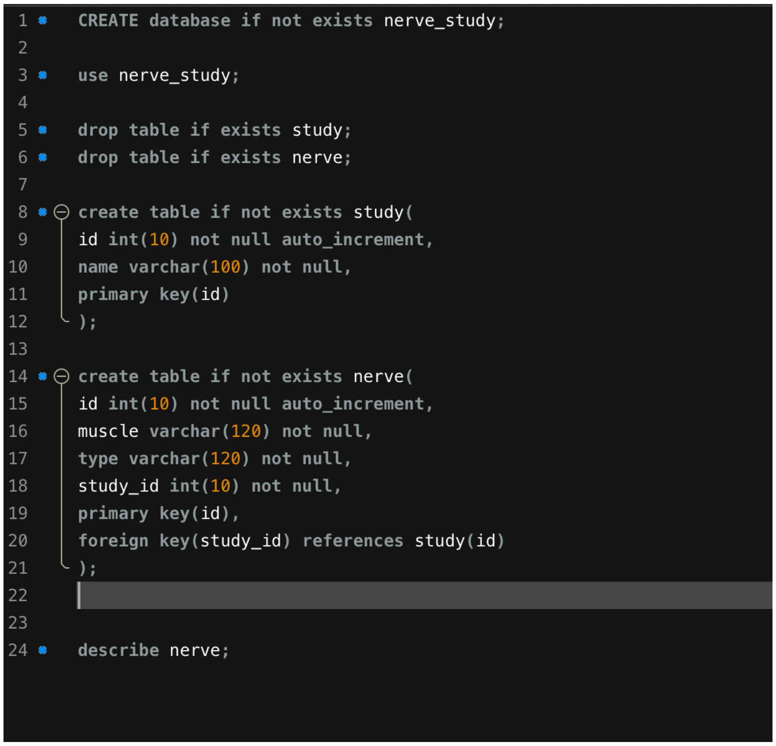
You need an option for each of the CRUD operations (Create, Read, Update, and Delete).

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.

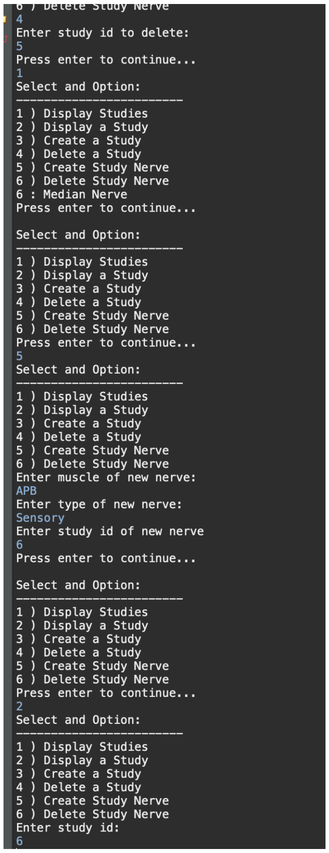
**GitHub URL:**

**SQL**



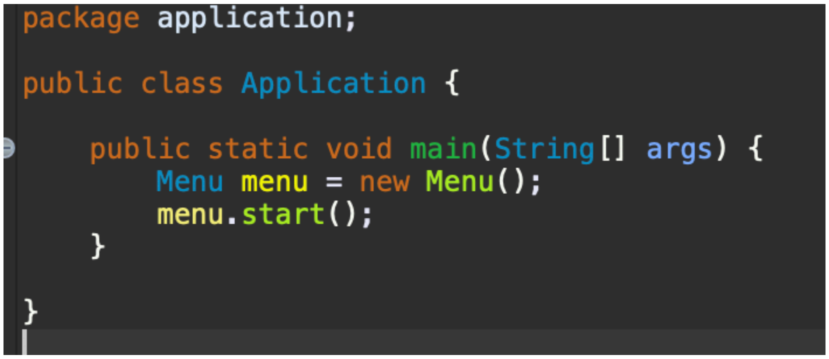
```
CREATE database if not exists nerve_study;

use nerve_study;

drop table if exists study;
drop table if exists nerve;

create table if not exists study(
id int(10) not null auto_increment,
name varchar(100) not null,
primary key(id)
);

create table if not exists nerve(
id int(10) not null auto_increment,
muscle varchar(120) not null,
type varchar(120) not null,
study_id int(10) not null,
primary key(id),
foreign key(study_id) references study(id)
);


describe nerve;
```

## SCREENSHOTS RUNNING APPLICATION



```
6 ) Delete Study Nerve
4
Enter study id to delete:
5
Press enter to continue...
1
Select and Option:
------------------------
1 ) Display Studies
2 ) Display a Study
3 ) Create a Study
4 ) Delete a Study
5 ) Create Study Nerve
6 ) Delete Study Nerve
6 : Median Nerve
Press enter to continue...

Select and Option:
------------------------
1 ) Display Studies
2 ) Display a Study
3 ) Create a Study
4 ) Delete a Study
5 ) Create Study Nerve
6 ) Delete Study Nerve
Press enter to continue...
5
Select and Option:
------------------------
1 ) Display Studies
2 ) Display a Study
3 ) Create a Study
4 ) Delete a Study
5 ) Create Study Nerve
6 ) Delete Study Nerve
Enter muscle of new nerve:
APB
Enter type of new nerve:
Sensory
Enter study id of new nerve
6
Press enter to continue...

Select and Option:
------------------------
1 ) Display Studies
2 ) Display a Study
3 ) Create a Study
4 ) Delete a Study
5 ) Create Study Nerve
6 ) Delete Study Nerve
Press enter to continue...
2
Select and Option:
------------------------
1 ) Display Studies
2 ) Display a Study
3 ) Create a Study
4 ) Delete a Study
5 ) Create Study Nerve
6 ) Delete Study Nerve
Enter study id:
6
```

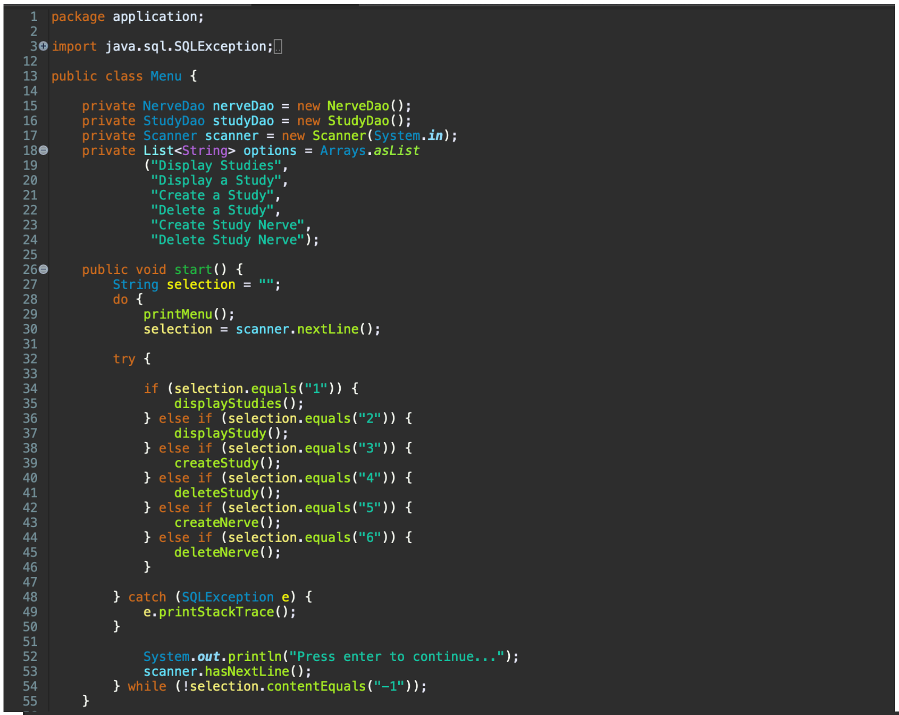
## APPLICTION



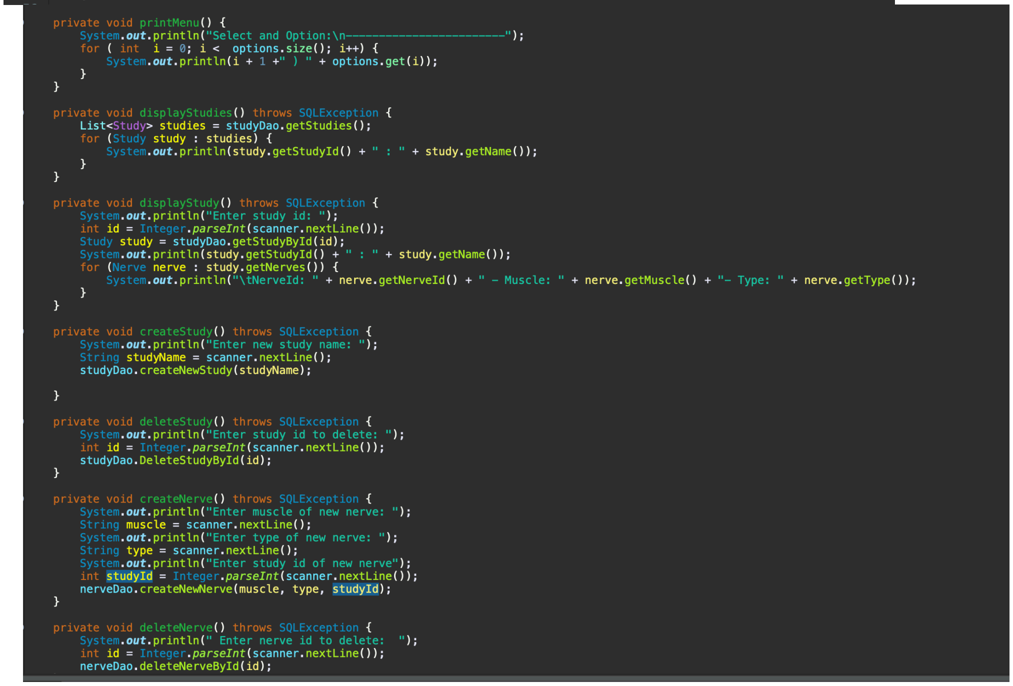
```java
package application;

public class Application {

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

}
```

GATEWAY COMMUNITY COLLEGE
A MARICOPA COMMUNITY COLLEGE

MENU

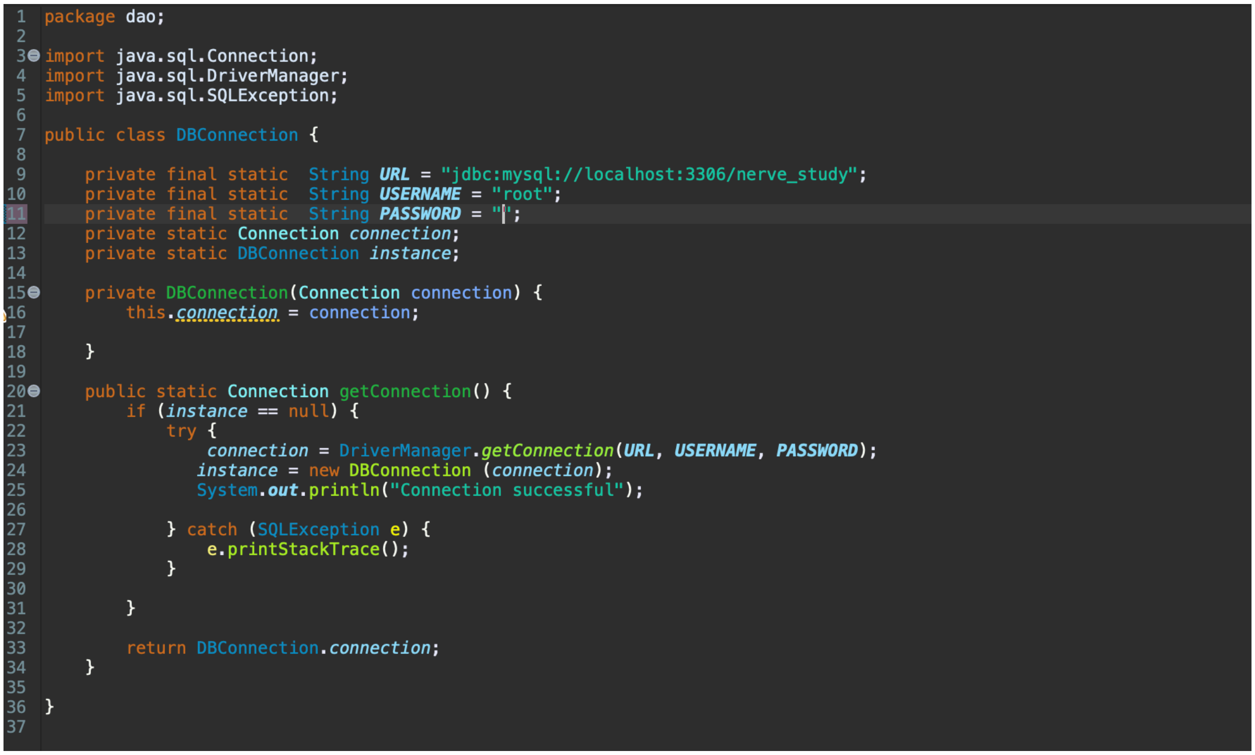


```
package application;

import java.sql.SQLException;

public class Menu {

    private NerveDao nerveDao = new NerveDao();
    private StudyDao studyDao = new StudyDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList
            ("Display Studies",
             "Display a Study",
             "Create a Study",
             "Delete a Study",
             "Create Study Nerve",
             "Delete Study Nerve");

    public void start() {
        String selection = "";
        do {
            printMenu();
            selection = scanner.nextLine();

        try {

            if (selection.equals("1")) {
                displayStudies();
            } else if (selection.equals("2")) {
                displayStudy();
            } else if (selection.equals("3")) {
                createStudy();
            } else if (selection.equals("4")) {
                deleteStudy();
            } else if (selection.equals("5")) {
                createNerve();
            } else if (selection.equals("6")) {
                deleteNerve();
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }

            System.out.println("Press enter to continue...");
            scanner.hasNextLine();
        } while (!selection.contentEquals("-1"));
    }
```



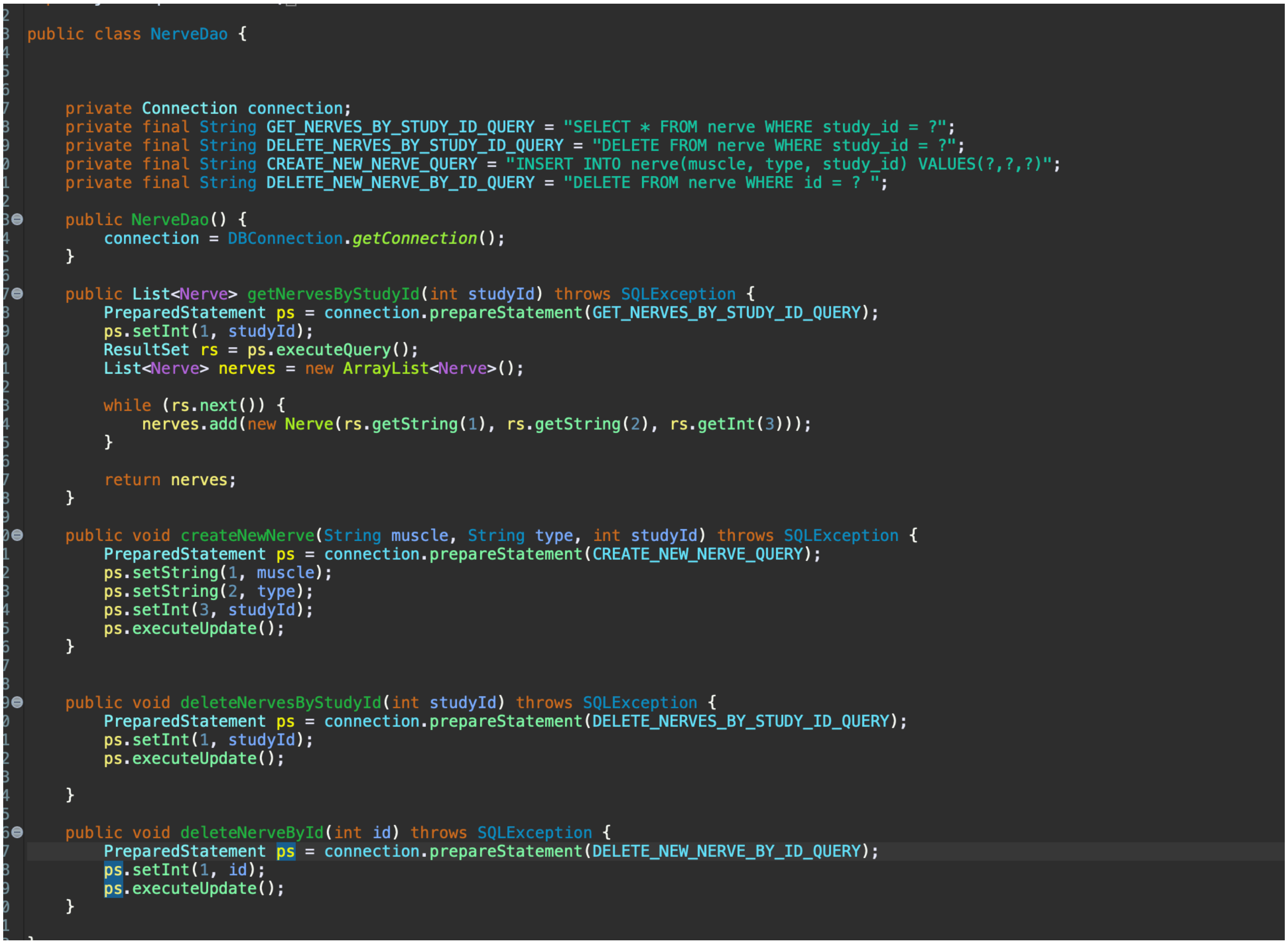
```
    private void printMenu() {
        System.out.println("Select and Option:\n-----------------------");
        for ( int  i = 0; i <  options.size(); i++) {
            System.out.println(i + 1 +" ) " + options.get(i));
        }
    }

    private void displayStudies() throws SQLException {
        List<Study> studies = studyDao.getStudies();
        for (Study study : studies) {
            System.out.println(study.getStudyId() + " : " + study.getName());
        }
    }

    private void displayStudy() throws SQLException {
        System.out.println("Enter study id: ");
        int id = Integer.parseInt(scanner.nextLine());
        Study study = studyDao.getStudyById(id);
        System.out.println(study.getStudyId() + " : " + study.getName());
        for (Nerve nerve : study.getNerves()) {
            System.out.println("\tNerveId: " + nerve.getNerveId() + " - Muscle: " + nerve.getMuscle() + "- Type: " + nerve.getType());
        }
    }

    private void createStudy() throws SQLException {
        System.out.println("Enter new study name: ");
        String studyName = scanner.nextLine();
        studyDao.createNewStudy(studyName);

    }

    private void deleteStudy() throws SQLException {
        System.out.println("Enter study id to delete: ");
        int id = Integer.parseInt(scanner.nextLine());
        studyDao.DeleteStudyById(id);
    }

    private void createNerve() throws SQLException {
        System.out.println("Enter muscle of new nerve: ");
        String muscle = scanner.nextLine();
        System.out.println("Enter type of new nerve: ");
        String type = scanner.nextLine();
        System.out.println("Enter study id of new nerve");
        int studyId = Integer.parseInt(scanner.nextLine());
        nerveDao.createNewNerve(muscle, type, studyId);
    }

    private void deleteNerve() throws SQLException {
        System.out.println(" Enter nerve id to delete:  ");
        int id = Integer.parseInt(scanner.nextLine());
        nerveDao.deleteNerveById(id);
```

DB CONNECTION



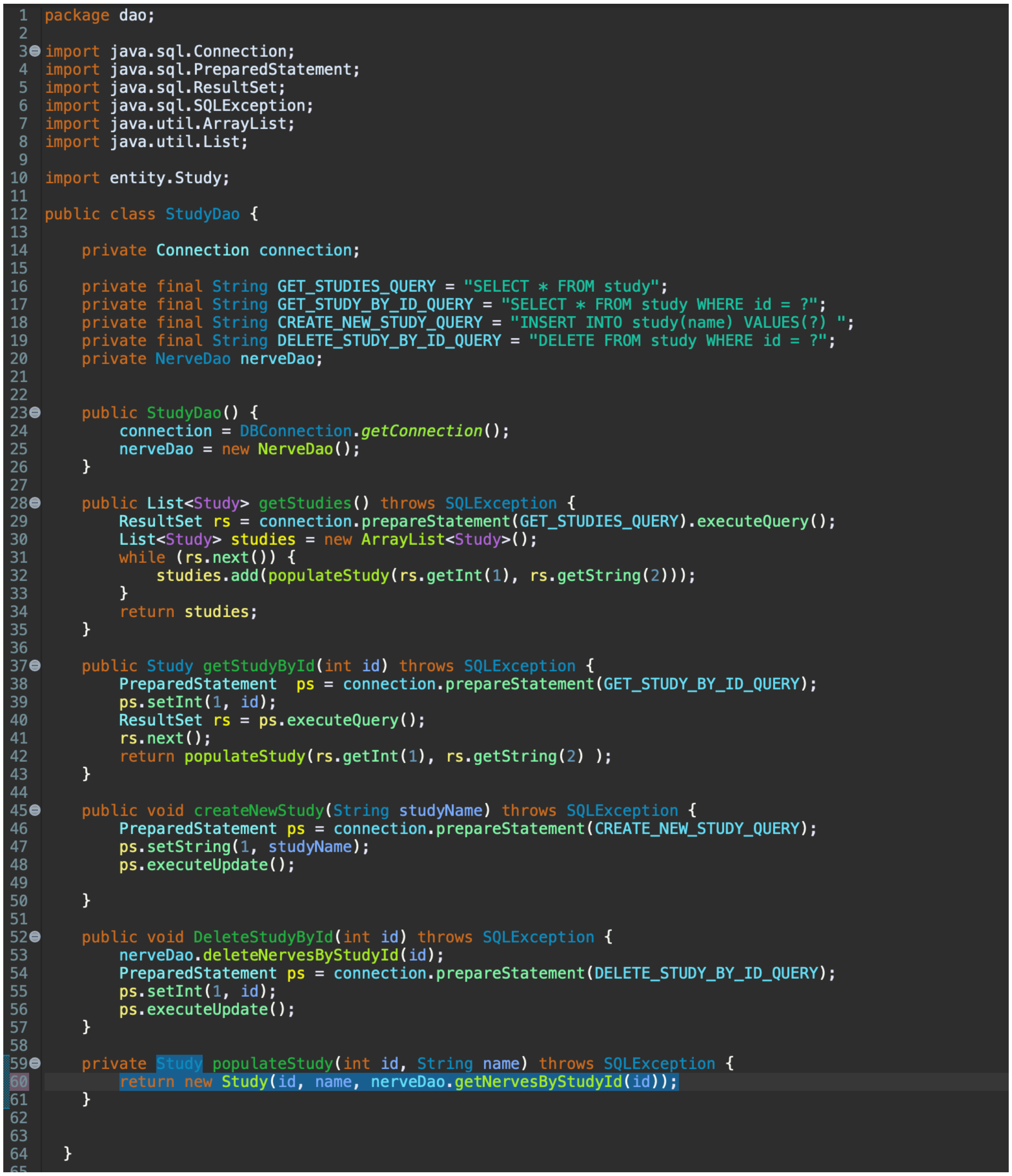
```java
package dao;

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

public class DBConnection {

    private final static  String URL = "jdbc:mysql://localhost:3306/nerve_study";
    private final static  String USERNAME = "root";
    private final static  String PASSWORD = "";
    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;
    }

}

```

NERVE DAO



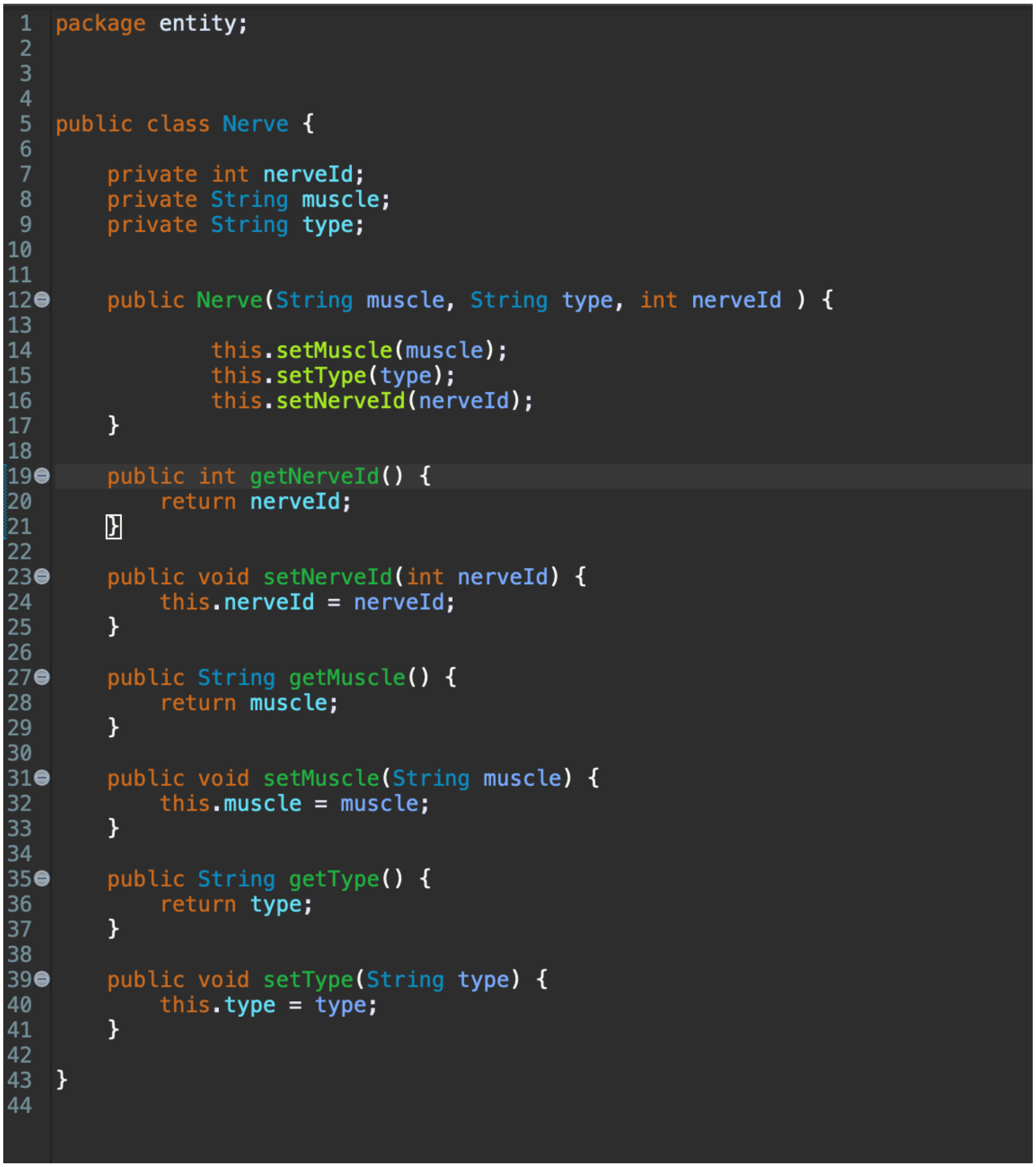
```java
public class NerveDao {



    private Connection connection;
    private final String GET_NERVES_BY_STUDY_ID_QUERY = "SELECT * FROM nerve WHERE study_id = ?";
    private final String DELETE_NERVES_BY_STUDY_ID_QUERY = "DELETE FROM nerve WHERE study_id = ?";
    private final String CREATE_NEW_NERVE_QUERY = "INSERT INTO nerve(muscle, type, study_id) VALUES(?,?,?)";
    private final String DELETE_NEW_NERVE_BY_ID_QUERY = "DELETE FROM nerve WHERE id = ? ";

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

    public List<Nerve> getNervesByStudyId(int studyId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_NERVES_BY_STUDY_ID_QUERY);
        ps.setInt(1, studyId);
        ResultSet rs = ps.executeQuery();
        List<Nerve> nerves = new ArrayList<Nerve>();

        while (rs.next()) {
            nerves.add(new Nerve(rs.getString(1), rs.getString(2), rs.getInt(3)));
        }

        return nerves;
    }

    public void createNewNerve(String muscle, String type, int studyId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_NERVE_QUERY);
        ps.setString(1, muscle);
        ps.setString(2, type);
        ps.setInt(3, studyId);
        ps.executeUpdate();
    }


    public void deleteNervesByStudyId(int studyId) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(DELETE_NERVES_BY_STUDY_ID_QUERY);
        ps.setInt(1, studyId);
        ps.executeUpdate();

    }

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

}
```

STUDY DAO



```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.Study;

public class StudyDao {

    private Connection connection;

    private final String GET_STUDIES_QUERY = "SELECT * FROM study";
    private final String GET_STUDY_BY_ID_QUERY = "SELECT * FROM study WHERE id = ?";
    private final String CREATE_NEW_STUDY_QUERY = "INSERT INTO study(name) VALUES(?) ";
    private final String DELETE_STUDY_BY_ID_QUERY = "DELETE FROM study WHERE id = ?";
    private NerveDao nerveDao;


    public StudyDao() {
        connection = DBConnection.getConnection();
        nerveDao = new NerveDao();
    }

    public List<Study> getStudies() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_STUDIES_QUERY).executeQuery();
        List<Study> studies = new ArrayList<Study>();
        while (rs.next()) {
            studies.add(populateStudy(rs.getInt(1), rs.getString(2)));
        }
        return studies;
    }

    public Study getStudyById(int id) throws SQLException {
        PreparedStatement  ps = connection.prepareStatement(GET_STUDY_BY_ID_QUERY);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populateStudy(rs.getInt(1), rs.getString(2) );
    }

    public void createNewStudy(String studyName) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_STUDY_QUERY);
        ps.setString(1, studyName);
        ps.executeUpdate();

    }

    public void DeleteStudyById(int id) throws SQLException {
        nerveDao.deleteNervesByStudyId(id);
        PreparedStatement ps = connection.prepareStatement(DELETE_STUDY_BY_ID_QUERY);
        ps.setInt(1, id);
        ps.executeUpdate();
    }

    private Study populateStudy(int id, String name) throws SQLException {
        return new Study(id, name, nerveDao.getNervesByStudyId(id));
    }


  }
```

NERVE

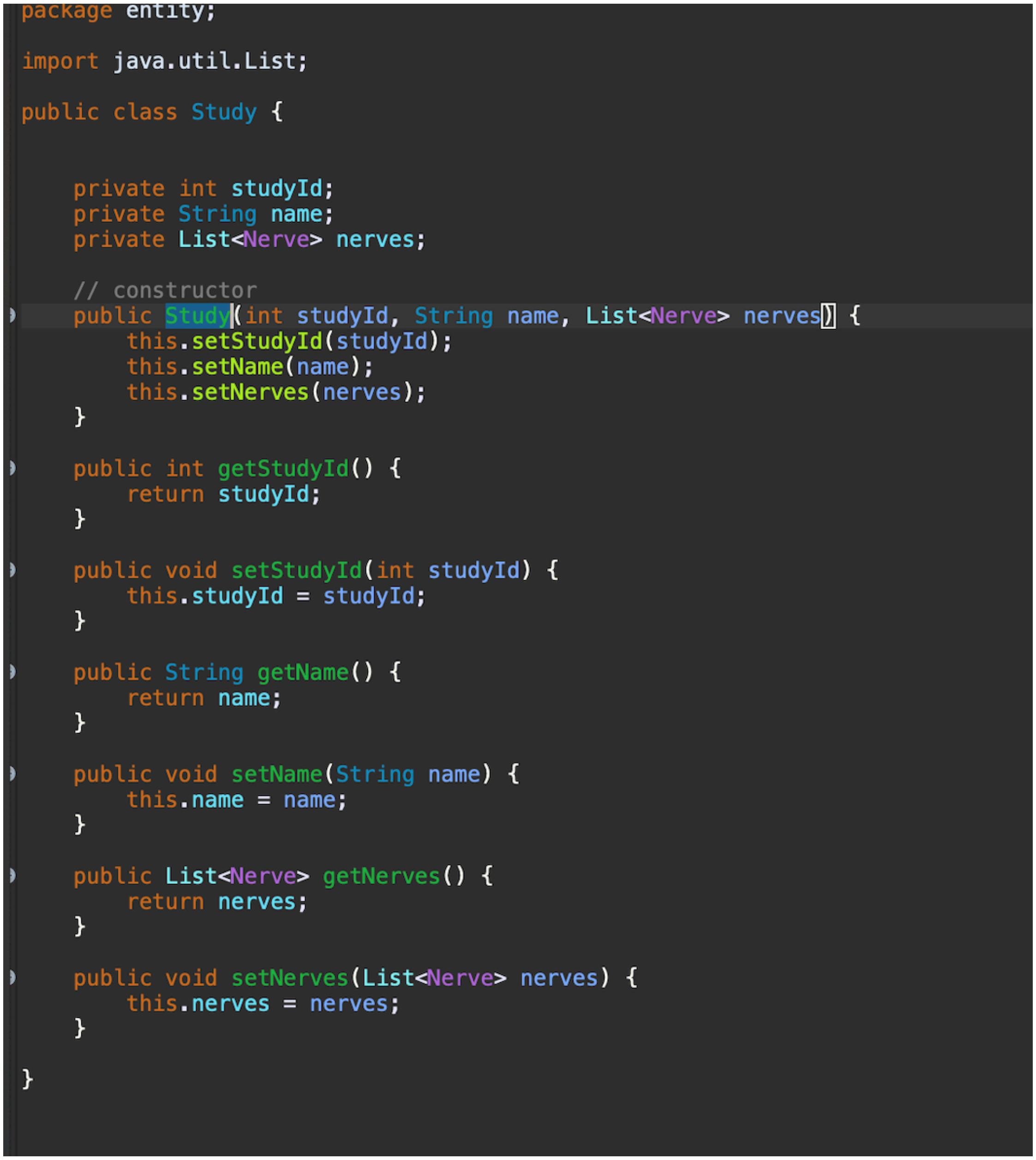


```java
package entity;



public class Nerve {

    private int nerveId;
    private String muscle;
    private String type;


    public Nerve(String muscle, String type, int nerveId ) {

            this.setMuscle(muscle);
            this.setType(type);
            this.setNerveId(nerveId);
    }

    public int getNerveId() {
        return nerveId;
    }

    public void setNerveId(int nerveId) {
        this.nerveId = nerveId;
    }

    public String getMuscle() {
        return muscle;
    }

    public void setMuscle(String muscle) {
        this.muscle = muscle;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

}
```

GATEWAY COMMUNITY COLLEGE

A MARICOPA COMMUNITY COLLEGE

## STUDY



```
package entity;

import java.util.List;

public class Study {


    private int studyId;
    private String name;
    private List<Nerve> nerves;

    // constructor
    public Study(int studyId, String name, List<Nerve> nerves) {
        this.setStudyId(studyId);
        this.setName(name);
        this.setNerves(nerves);
    }

    public int getStudyId() {
        return studyId;
    }

    public void setStudyId(int studyId) {
        this.studyId = studyId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Nerve> getNerves() {
        return nerves;
    }

    public void setNerves(List<Nerve> nerves) {
        this.nerves = nerves;
    }

}
```

GATEWAY
COMMUNITY COLLEGE

A MARICOPA COMMUNITY COLLEGE