# COMPUTER PROGRAMMING

## PROG8580

Section 1


Alpha Team

SAEED MOGHADAM – 8799319
AMANJOT SIGNH – 8807626
DELMI ENRIQUE ALARCON GUERRA – 8787512


PROJECT 2

## Problem Description

Write a Java console program that views, inserts, and updates staff information stored in a MySQL database, as shown in the following figure. Your console application should take arguments through the command line for view the entries, insert an entry, and update staff information of an entry.

## Group Participation

| Name | Student # | Participation |
|---|---|---|
| Saeed Moghadam | 8799319 | 33% |
| DELMI ENRIQUE ALARCON GUERRA | 8787512 | 33% |
| AMANJOT SIGNH | 8807626 | 33% |

## Analysis

The program connects to mySQL database management system to manipulate (execute DML queries) on created table Staff in database my_db. Mentioned table and database was created according to project question description. This document only focuses on works in java side and skips explaining workbench installation and process taken in workbench editor to create table and database.

The program displays four options and ask user to choose one of them:

1- Select
2- Insert
3- Update
4- Exit

According to user 's choice below programs are executed:

Select:

- select query is executed and data in table staff is displayed for the user in an appropriate format (data is represented in a nice table format).
- System displays above mentioned options for the user to continue with other choice

Insert:

- the system prompts the user to give a value for each table column one by one. (For example, the user inserts the last name and enters).
- If the user enters a value longer than the predefined column's length, the system displays an error: Error: Data truncation: Data too long for column Name.
- If the user doesn't enter the value for a column, the system considers a null value.
- The ID column is primary and Identity. The system gives the number starting one to each row and increments the number for the subsequent row insertion. Itis the usual procedure in databases.
- At the end of inserting the row, the system executes a select query to display the inserted row to the user.

Update:

- The system displays the table content to the user and asks for entering an ID because the update query is executed based on ID in databases.
- If the entered ID does not match any IDs of the table staff, the system again asks the user to enter an ID.
- System displays message: Enter a new value or press enter if you want the value in parentheses not to be updated if you want to delete them add a space.
- If the user enters a value longer than the predefined column's length, the system displays an error: Error: Data truncation: Data too long for column Name.
- System displays above mentioned options for the user to continue with other choice.

## Design:

The program includes four class:

### Database Connection:

The StaffDB class contains the data to make the connection to the database, as well as to close the connection.

### Staff Class:

The staff class contains the fields just like the staff table in the database, which contains an accessor and a mutator.

### Database transactions:

The StaffSQL class contains the insert, update, and select queries to the database:

> **Insert Method:** This method establishes a connection with the database, then generates the script with the parameters to insert into the table and finally executes the insert statement.
>
> **Update method:** This method establishes a connection with the database, then generates the script with the parameters to update in the table and finally executes the insert statement.
>
> **get(id) method:** This method establishes a connection with the database, then generates a query script on the staff table by sending the Id parameter, finally it executes the query, and the obtained data is set in the Staff class.
>
> **list() method:** This method establishes a connection with the database, then generates a query script on the staff table, finally executes the query and inserts the obtained data into an arrayList.

### Display Information

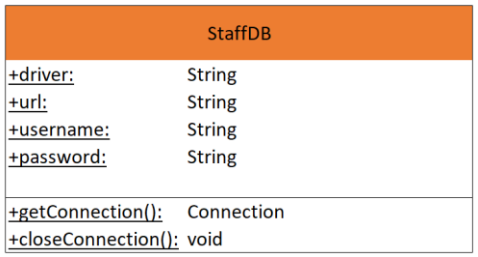
This class contains the main class of the program, in which it shows a menu to select the type of transaction that you want to perform on the Staff table:

> **optionSelect:** this method validates that there is data in the table to display it in table format, obtaining it from the StaffDB class, if it does not find data it displays a message.

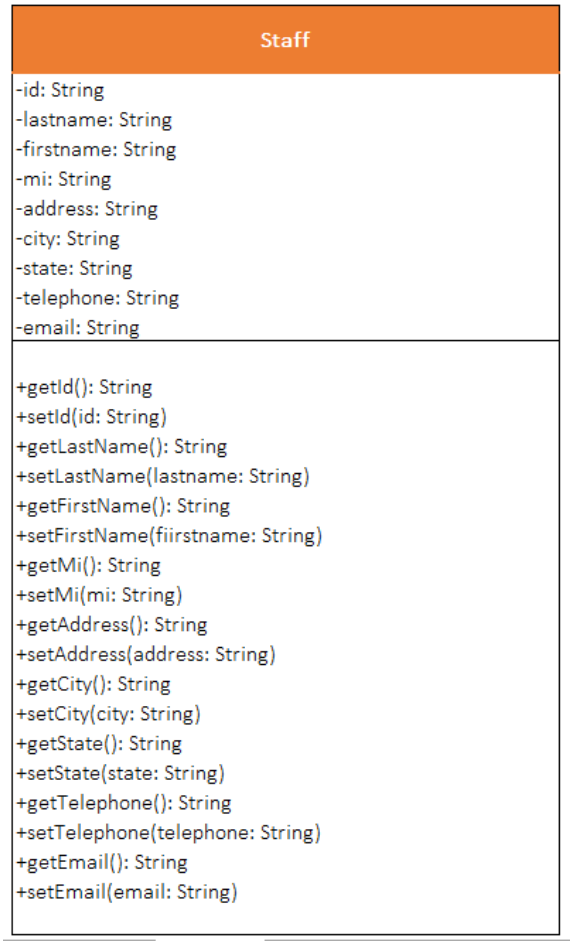
optionInsert: This method first checks if there is data in the table, to automatically generate the new Id to insert, then prompts the user to enter all the fields and finally displays the table data.

optionUpdate: this method verifies that there is data in the table to update, if there is data, it is displayed on the screen and requests that a valid Id be entered to be modified, then it requests the data to be modified, showing the existing ones, if the user wishes to modify a field, it has to enter it, if the user does not want to modify it, he can simply press Enter, finally the updated data is updated and displayed on the screens.

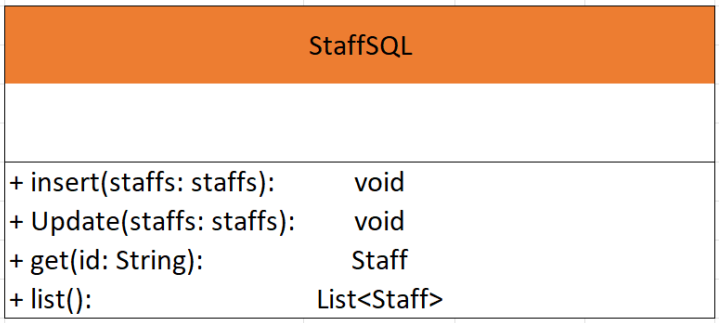
This is the UML class diagram for the StaffDB class.



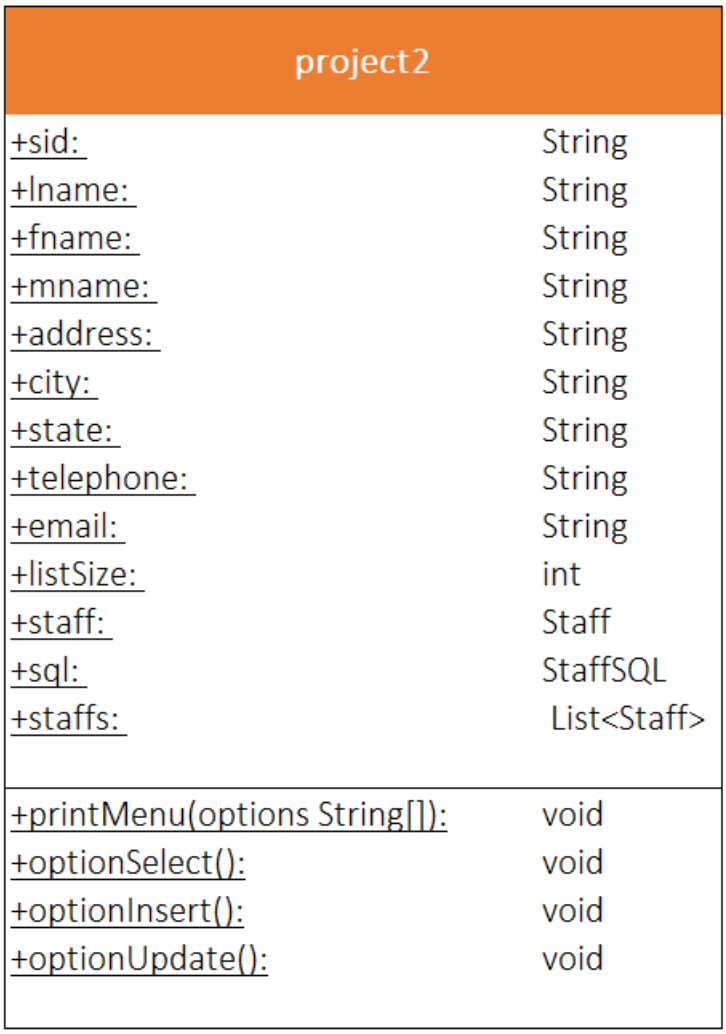

This is the UML class diagram for the Staff class.



This is the UML class diagram for the StaffSQL class.




This is the UML class diagram for the project2 class.



## Coding

```
/*
 * AMANJOT SIGNH (8807626)
 * DELMI ALARCON (8787512)
 * SAEED MOGHADAM (8799319)
 * PROG8580 - Computer Programming - Section 1
 * Project 2
 */
```

----------Class Project2 (including main method)----------

```
import java.util.*;
import static java.lang.System.exit;

public class Project2 {
    // Declare variables to store data to display
    static String sid;
    static String lname;
    static String fname;
    static String mname;
    static String address;
    static String city;
    static String state;
    static String telephone;
    static String email;

    static int listSize = 0;
    static Staff staff = new Staff();
    static StaffSQL sql = new StaffSQL();
    static List<Staff> staffs;

    public static void main(String[] args) throws Exception {
        // Declare options with menu list
       String[] options = {"1- Select",
                           "2- Insert",
                           "3- Update",
                           "4- Exit",
        };

        Scanner input = new Scanner(System.in);

        int option = 1;
        while (option != 4){
             // Call printMenu method sending options
            printMenu(options);
            try {
                option = input.nextInt();
                switch (option){
                    case 1: optionSelect(); break; // Call optionSelect() method
                    case 2: optionInsert(); break; // Call optionInsert() method
                    case 3: optionUpdate(); break; // Call optionUpdate() method
                    case 4: exit(0);
                }
                StaffDB.closeConnection();
            }
            catch (Exception ex){
                System.out.println("Please enter an integer value between 1 and " +
options.length);
                input.next();
```

```
            }
        }
    }

    public static void printMenu(String[] options){
        for (String option : options){
            System.out.println(option);
        }
        System.out.print("Choose your option : ");
    }
```

---------- Option Select ----------

```
    private static void optionSelect() {
        staffs = sql.list();
        listSize = staffs.size();

        if (listSize > 0) {
              System.out.println("-".repeat(140));
              System.out.printf("%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-
20s", "ID", "Last Name", "First Name", "Mid", "Address", "City", "State", "Telephone", "Email");
              System.out.println();
              System.out.println("-".repeat(140));
              for(Staff staff: staffs) {
                  sid = staff.getId();
                   lname = staff.getLastName();
                  fname = staff.getFirstName();
                  mname = staff.getMi();
                  address = staff.getAddress();
                  city = staff.getCity();
                  state = staff.getState();
                  telephone = staff.getTelephone();
                  email = staff.getEmail();

                  String output = "%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-
20s";
                  System.out.println(String.format(output, sid, lname, fname, mname, address,
city, state, telephone, email));
              }
              System.out.println("-".repeat(140));
        }
          else {
             System.out.println("\nThere is no data in the Staff table.\n");
        }
    }
```

---------- Option Insert ----------

```
    private static void optionInsert() {
        Scanner input = new Scanner(System.in);
        staffs = sql.list();

        int lastIdx = staffs.size() - 1;
        int newId;

        if (lastIdx >= 0) {
                    Staff lastElement = staffs.get(lastIdx);
                    newId = Integer. parseInt(lastElement.getId()) + 1;
        }
        else {
             newId = 1;
        }
```

```
        System.out.print("Enter Last Name: ");
              lname = input.nextLine();
              System.out.print("Enter First Name: ");
              fname = input.nextLine();
              System.out.print("Enter Middle Name: ");
              mname = input.nextLine();
              System.out.print("Enter Address: ");
              address = input.nextLine();
              System.out.print("Enter City: ");
              city = input.nextLine();
              System.out.print("Enter State: ");
              state = input.nextLine();
              System.out.print("Enter Telephone: ");
              telephone = input.nextLine();
              System.out.print("Enter Email: ");
              email = input.nextLine();

              staff.setId(Integer.toString(newId));
              staff.setLastName(lname);
              staff.setFirstName(fname);
              staff.setMi(mname);
              staff.setAddress(address);
              staff.setCity(city);
              staff.setState(state);
              staff.setTelephone(telephone);
              staff.setEmail(email);


        sql.insert(staff);
        optionSelect();
    }
        ---------- Option Update ----------

    private static void optionUpdate() {
       System.out.println();
       optionSelect();

       if (listSize > 0) {
              Scanner input = new Scanner(System.in);

              System.out.print("Enter an ID of the list to Update: ");
                     String id = input.nextLine();
                     Staff staff = sql.get(id);

                     while(staff.getId() == null) {
                            System.out.print("Enter an ID of the list to Update: ");
                            id = input.nextLine();

                            staff = sql.get(id);
                     };
                     System.out.println("-".repeat(140));
                     System.out.println("Enter a new value or press enter if you want the value in
parentheses not to be updated, if you want to delete them add a space.");
                     System.out.println("-".repeat(140));

                     System.out.print("Last Name (" + staff.getLastName() + "): ");
                     lname = input.nextLine();
                     System.out.print("First Name (" + staff.getFirstName() + "): ");
                     fname = input.nextLine();
                     System.out.print("Middle Name (" + staff.getMi() + "): ");
```

```
                    mname = input.nextLine();
                    System.out.print("Address (" + staff.getAddress() + "): ");
                    address = input.nextLine();
                    System.out.print("City (" + staff.getCity() + "): ");
                    city = input.nextLine();
                    System.out.print("State (" + staff.getState() + "): ");
                    state = input.nextLine();
                    System.out.print("Telephone (" + staff.getTelephone() + "): ");
                    telephone = input.nextLine();
                    System.out.print("Email (" + staff.getEmail() + "): ");
                    email = input.nextLine();

                staff.setLastName((lname=="")?staff.getLastName():lname);
                staff.setFirstName((fname=="")?staff.getFirstName():fname);
                staff.setMi((mname=="")?staff.getMi():mname);
                staff.setAddress((address=="")?staff.getAddress():address);
                staff.setCity((city=="")?staff.getCity():city);
                staff.setState((state=="")?staff.getState():state);
                staff.setTelephone((telephone=="")?staff.getTelephone():telephone);
                staff.setEmail((email=="")?staff.getEmail():email);

                sql.update(staff);
                optionSelect();
        }
    }
}
```

----------Class Staff ----------

```
public class Staff {
	private String id;
	private String lastname;
	private String firstname;
	private String mi;
	private String address;
	private String city;
	private String state;
	private String telephone;
	private String email;

	//generating getter and setter methods
	public String getId() {
		return id;}

	public void setId(String id) {
		this.id = id;}

	public String getLastName() {
		return lastname;}

	public void setLastName(String lastname) {
		this.lastname = lastname;}

	public String getFirstName() {
		return firstname;}

	public void setFirstName(String firstname) {
		this.firstname = firstname;}

	public String getMi() {
```

```
		return mi;}

	public void setMi(String mi) {
		this.mi = mi;}

	public String getAddress() {
		return address;}

	public void setAddress(String address) {
		this.address = address;}

	public String getCity() {
		return city;}

	public void setCity(String city) {
		this.city = city;}

	public String getState() {
		return state;}

	public void setState(String state) {
		this.state = state;}

	public String getTelephone() {
		return telephone;}

	public void setTelephone(String telephone) {
		this.telephone = telephone;}

	public String getEmail() {
		return email;}

	public void setEmail(String email) {
		this.email = email;}
}
```

----------Class StaffDB----------

```
import static java.lang.System.exit;
import java.sql.*;

public class StaffDB {
	// Declare variables to get a connection
	static Connection connection;
	static String driver = "com.mysql.cj.jdbc.Driver";
	static String url = "jdbc:mysql://localhost:3306/my_db";
	static String username = "root";
	static String password = "root";

	public static Connection getConnection() throws Exception {
		try {
			// Get Connection with paremeters inserted
			Class.forName(driver);
			connection = DriverManager.getConnection(url, username, password);
		} catch (SQLException e) {
			System.out.println("\nError: " + e.getMessage());
			exit(0);
		}
		return connection;
	}
```

```
	public static void closeConnection() throws Exception {
		try {
			// Close connection
			connection.close();
		} catch (SQLException e) {
			System.out.println("\nError: " + e.getMessage());
		}
	}
}
```

## ----------Class StaffSQL ----------

```
import java.sql.*;
import java.util.*;
import static java.lang.System.exit;

public class StaffSQL {
```

### ---------- method Insert ----------

```
	public void insert(Staff staffs) {
		try {
			// Establish connection to the database
			Connection con = StaffDB.getConnection();
			// Query to insert data
			String sql = "INSERT INTO
staff(id,lastname,firstname,mi,address,city,state,telephone,email) VALUES (?,?,?,?,?,?,?,?,?)";
			// Creates a PreparedStatement object for sending parameterized SQL statements
			// to the database
			PreparedStatement ps = con.prepareStatement(sql);
			// Sets the designated parameter
			ps.setString(1, staffs.getId());
			ps.setString(2, staffs.getLastName());
			ps.setString(3, staffs.getFirstName());
			ps.setString(4, staffs.getMi());
			ps.setString(5, staffs.getAddress());
			ps.setString(6, staffs.getCity());
			ps.setString(7, staffs.getState());
			ps.setString(8, staffs.getTelephone());
			ps.setString(9, staffs.getEmail());
			// Execute the insert statement
			ps.executeUpdate();
			System.out.println("-".repeat(140));
			System.out.println("Saved!");
		} catch (Exception e) {
			System.out.println("\nError: " + e.getMessage());
			exit(0);
		}
	}
```

### ---------- method Update ----------

```
	public void update(Staff staffs) {
		try {
			// Establish connection to the database
			Connection con = StaffDB.getConnection();
			// Query to update data
			String sql = "UPDATE staff SET lastname = ?, firstname = ?, mi = ?, address
= ?, city = ?, state = ?, telephone = ?, email = ? WHERE id = ?";
			// Creates a PreparedStatement object for sending parameterized SQL statements
```

```
                // to the database
                PreparedStatement ps = con.prepareStatement(sql);
                ps.setString(1, staffs.getLastName());
                ps.setString(2, staffs.getFirstName());
                ps.setString(3, staffs.getMi());
                ps.setString(4, staffs.getAddress());
                ps.setString(5, staffs.getCity());
                ps.setString(6, staffs.getState());
                ps.setString(7, staffs.getTelephone());
                ps.setString(8, staffs.getEmail());
                ps.setString(9, staffs.getId());
                // Execute the update statement
                ps.executeUpdate();
                System.out.println("-".repeat(140));
                System.out.println("Updated!");
        } catch (Exception e) {
                System.out.println("\nError: " + e.getMessage());
                exit(0);
        }

}

public Staff get(String id) {
        // Declare staff class
        Staff staff = new Staff();
        try {
                // Establish connection to the database
                Connection con = StaffDB.getConnection();
                // Query to select data from Id
                String sql = "SELECT * FROM staff WHERE id = ?";
                // Creates a PreparedStatement object for sending parameterized SQL statements
                // to the database
                PreparedStatement ps = con.prepareStatement(sql);
                ps.setString(1, id);
                // Execute the statement and store it in a resultset
                ResultSet rs = ps.executeQuery();
                if (rs.next()) {
                        // Add data values in staff class
                        staff.setId(rs.getString("id"));
                        staff.setLastName(rs.getString("lastname"));
                        staff.setFirstName(rs.getString("firstname"));
                        staff.setMi(rs.getString("mi"));
                        staff.setAddress(rs.getString("address"));
                        staff.setCity(rs.getString("city"));
                        staff.setState(rs.getString("state"));
                        staff.setTelephone(rs.getString("telephone"));
                        staff.setEmail(rs.getString("email"));
                }
        } catch (Exception e) {
                System.out.println("\nError: " + e.getMessage());
                exit(0);
        }
        // Return staff class with data
        return staff;
}
---------- method list to return  the result of select query ----------

public List<Staff> list() {
        // Declare the list of type Staff
        List<Staff> list = new ArrayList<Staff>();
        try {
```

```
                // Establish connection to the database
                Connection con = StaffDB.getConnection();
                // Query to update data
                String sql = "SELECT * FROM staff ";
                // Creates a PreparedStatement object for sending parameterized SQL statements
                // to the database
                PreparedStatement ps = con.prepareStatement(sql);
                // Execute the statement and store it in a resultset
                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                        // Declare class staff and add data values
                        Staff staff = new Staff();
                        staff.setId(rs.getString("id"));
                        staff.setLastName(rs.getString("lastname"));
                        staff.setFirstName(rs.getString("firstname"));
                        staff.setMi(rs.getString("mi"));
                        staff.setAddress(rs.getString("address"));
                        staff.setCity(rs.getString("city"));
                        staff.setState(rs.getString("state"));
                        staff.setTelephone(rs.getString("telephone"));
                        staff.setEmail(rs.getString("email"));

                        // Add staff in list created
                        list.add(staff);
                }
        } catch (Exception e) {
                System.out.println("\nError: " + e.getMessage());
                exit(0);
        }
        // Return list with data
        return list;
    }
}
```

# Testing

The program was tested based on data selection, insert and update.

## *Data Selection*

eclipse-workspace - Project2/src/Project2.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- javaProjects
- Project1
- Project2
  - src
    - (default package)
      - Project2.java
      - Staff.java
      - StaffDB.java
      - StaffSQL.java
    - Project2.java
  - JRE System Library [JavaSE-17]
  - Referenced Libraries

Project2.java | Staff.java | StaffDB.java | StaffSQL.java

```java
/*
 * AMANJOT SIGNH (8807626)
 * DELMI ALARCON (8787512)
 * SAEED MOGHADAM (8799319)
 * PROG8580 - Computer Programming - Section 1
 * Project 2
 */

import java.util.*;
import static java.lang.System.exit;

public class Project2 {
    // Declare variables to store data to display
    static String sid;
    static String lname;
    static String fname;
    static String mname;
    static String address;
    static String city;
    static String state;
    static String telephone;
    static String email;

    static int listSize = 0;
    static Staff staff = new Staff();
    static StaffSQL sql = new StaffSQL();
    static List<Staff> staffs;

    public static void main(String[] args) throws Exception {
        // Declare options with menu list
        String[] options = {"1- Select",
                            "2- Insert",
                            "3- Update",
                            "4- Exit",
        };

        Scanner input = new Scanner(System.in);

        int option = 1;
        while (option != 4){
            // Call printMenu method sending options
            printMenu(options);
            try {
                option = input.nextInt();
                switch (option){
                    case 1: optionSelect(); break; // Call optionSelect() method
                    case 2: optionInsert(); break; // Call optionInsert() method
                    case 3: optionUpdate(); break; // Call optionUpdate() method
                    case 4: exit(0);
                }
                StaffDB.closeConnection();
            }
            catch (Exception ex){
                System.out.println("Please enter an integer value between 1 and " + options.length);
                input.next();
            }
        }
    }

    public static void printMenu(String[] options){
        for (String option : options){
            System.out.println(option);
        }
        System.out.print("Choose your option : ");
    }

    private static void optionSelect() {
        staffs = sql.list();
        listSize = staffs.size();

        if (listSize > 0) {
            System.out.println("-".repeat(140));
            System.out.printf("%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s", "ID", "Last Name", "First Name", "Mid", "Address", "City", "State", "Telephone", "Email");
            System.out.println();
            System.out.println("-".repeat(140));
            for(Staff staff: staffs) {
                sid = staff.getId();
                lname = staff.getLastName();
                fname = staff.getFirstName();
                mname = staff.getMi();
                address = staff.getAddress();
                city = staff.getCity();
                state = staff.getState();
                telephone = staff.getTelephone();
                email = staff.getEmail();

                String output = "%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s";
                System.out.println(String.format(output, sid, lname, fname, mname, address, city, state, telephone, email));
            }
            System.out.println("-".repeat(140));
        }
        else {
            System.out.println("\nThere is no data in the Staff table.\n");
        }
    }

    private static void optionInsert() {
        Scanner input = new Scanner(System.in);
        staffs = sql.list();

        int lastIdx = staffs.size() - 1;
        int newId;

        if (lastIdx >= 0) {
            Staff lastElement = staffs.get(lastIdx);
            newId = Integer. parseInt(lastElement.getId()) + 1;
        }
        else {
            newId = 1;
        }

        System.out.print("Enter Last Name: ");
        lname = input.nextLine();
        System.out.print("Enter First Name: ");
        fname = input.nextLine();
        System.out.print("Enter Middle Name: ");
        mname = input.nextLine();
        System.out.print("Enter Address: ");
        address = input.nextLine();
        System.out.print("Enter City: ");
        city = input.nextLine();
        System.out.print("Enter State: ");
        state = input.nextLine();
        System.out.print("Enter Telephone: ");
        telephone = input.nextLine();
        System.out.print("Enter Email: ");
        email = input.nextLine();

        staff.setId(Integer.toString(newId));
        staff.setLastName(lname);
        staff.setFirstName(fname);
        staff.setMi(mname);
        staff.setAddress(address);
        staff.setCity(city);
        staff.setState(state);
        staff.setTelephone(telephone);
        staff.setEmail(email);


        sql.insert(staff);
        optionSelect();
    }

    private static void optionUpdate() {
        System.out.println();
        optionSelect();

        if (listSize > 0) {
            Scanner input = new Scanner(System.in);

            System.out.print("Enter an ID of the list to Update: ");
            String id = input.nextLine();
            Staff staff = sql.get(id);

            while(staff.getId() == null) {
                System.out.print("Enter an ID of the list to Update: ");
                id = input.nextLine();

                staff = sql.get(id);
            };
            System.out.println("-".repeat(140));
            System.out.println("Enter a new value or press enter if you want the value in parentheses not to be updated, if you want to delete them add a space.");
            System.out.println("-".repeat(140));

            System.out.print("Last Name (" + staff.getLastName() + "): ");
            lname = input.nextLine();
            System.out.print("First Name (" + staff.getFirstName() + "): ");
            fname = input.nextLine();
            System.out.print("Middle Name (" + staff.getMi() + "): ");
            mname = input.nextLine();
            System.out.print("Address (" + staff.getAddress() + "): ");
            address = input.nextLine();
            System.out.print("City (" + staff.getCity() + "): ");
            city = input.nextLine();
            System.out.print("State (" + staff.getState() + "): ");
            state = input.nextLine();
            System.out.print("Telephone (" + staff.getTelephone() + "): ");
            telephone = input.nextLine();
            System.out.print("Email (" + staff.getEmail() + "): ");
            email = input.nextLine();

            staff.setLastName((lname=="")?staff.getLastName():lname);
            staff.setFirstName((fname=="")?staff.getFirstName():fname);
            staff.setMi((mname=="")?staff.getMi():mname);
            staff.setAddress((address=="")?staff.getAddress():address);
            staff.setCity((city=="")?staff.getCity():city);
            staff.setState((state=="")?staff.getState():state);
            staff.setTelephone((telephone=="")?staff.getTelephone():telephone);
            staff.setEmail((email=="")?staff.getEmail():email);

            sql.update(staff);
            optionSelect();
        }
    }
}
```

Console

Project [Java Application] C:\Program Files\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Apr. 12, 2022, 8:01:03 p.m.)

```
1- Select
2- Insert
3- Update
4- Exit
Choose your option : 1
----------------------------------------------------------------------------------------------------------------------------------------
 ID | Last Name       | First Name      | Mid | Address             | City       | State | Telephone  | Email
----------------------------------------------------------------------------------------------------------------------------------------
  1 | Alarcon         | Delmi           | E   | 1425 BlockLine      | Kitchener  | ON    | 2262014387 | delmialarcon@gmail.com
  2 | Moghadam        | Saeed           |     | 475 King Street     | Kitchener  | ON    | 6538743298 | saeedmoghadam@gmail.com
  3 | Signh           | Amanjot         |     | 878 Ottawa Street   | Toronto    | ON    | 5195882903 | amanjotsignh@gmail.com
----------------------------------------------------------------------------------------------------------------------------------------
```

***Data Insert:*** After entering the requested data, the data of the table is displayed


```java
/*
 * AMANJOT SIGNH (8807626)
 * DELMI ALARCON (8787512)
 * SAEED MOGHADAM (8799319)
 * PROG8580 - Computer Programming - Section 1
 * Project 2
 */

import java.util.*;
import static java.lang.System.exit;

public class Project2 {
    // Declare variables to store data to display
    static String sid;
    static String lname;
    static String fname;
    static String mname;
    static String address;
    static String city;
    static String state;
    static String telephone;
    static String email;

    static int listSize = 0;
    static Staff staff = new Staff();
    static StaffSQL sql = new StaffSQL();
    static List<Staff> staffs;

    public static void main(String[] args) throws Exception {
        // Declare options with menu list
        String[] options = {"1- Select",
                            "2- Insert",
                            "3- Update",
                            "4- Exit",
        };

        Scanner input = new Scanner(System.in);

        int option = 1;
        while (option != 4){
            // Call printMenu method sending options
            printMenu(options);
            try {
                option = input.nextInt();
                switch (option){
                    case 1: optionSelect(); break; // Call optionSelect() method
                    case 2: optionInsert(); break; // Call optionInsert() method
                    case 3: optionUpdate(); break; // Call optionUpdate() method
                    case 4: exit(0);
                }
                StaffDB.closeConnection();
            }
            catch (Exception ex){
                System.out.println("Please enter an integer value between 1 and " + options.length);
                input.next();
            }
        }
    }

    public static void printMenu(String[] options){
        for (String option : options){
            System.out.println(option);
        }
        System.out.print("Choose your option : ");
    }

    private static void optionSelect() {
        staffs = sql.list();
        listSize = staffs.size();

        if (listSize > 0) {
            System.out.println("-".repeat(140));
            System.out.printf("%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s", "ID", "Last Name", "First Name", "Mid", "Address", "City", "State", "Telephone", "Email");
            System.out.println();
            System.out.println("-".repeat(140));
            for(Staff staff: staffs) {
                sid = staff.getId();
                lname = staff.getLastName();
                fname = staff.getFirstName();
                mname = staff.getMi();
                address = staff.getAddress();
                city = staff.getCity();
                state = staff.getState();
                telephone = staff.getTelephone();
                email = staff.getEmail();

                String output = "%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s";
                System.out.println(String.format(output, sid, lname, fname, mname, address, city, state, telephone, email));
            }
            System.out.println("-".repeat(140));
        }
        else {
            System.out.println("\nThere is no data in the Staff table.\n");
        }
    }

    private static void optionInsert() {
        Scanner input = new Scanner(System.in);
        staffs = sql.list();

        int lastIdx = staffs.size() - 1;
        int newId;

        if (lastIdx >= 0) {
            Staff lastElement = staffs.get(lastIdx);
            newId = Integer. parseInt(lastElement.getId()) + 1;
        }
        else {
            newId = 1;
        }

        System.out.print("Enter Last Name: ");
        lname = input.nextLine();
        System.out.print("Enter First Name: ");
        fname = input.nextLine();
        System.out.print("Enter Middle Name: ");
        mname = input.nextLine();
        System.out.print("Enter Address: ");
        address = input.nextLine();
        System.out.print("Enter City: ");
        city = input.nextLine();
        System.out.print("Enter State: ");
        state = input.nextLine();
        System.out.print("Enter Telephone: ");
        telephone = input.nextLine();
        System.out.print("Enter Email: ");
        email = input.nextLine();

        staff.setId(Integer.toString(newId));
        staff.setLastName(lname);
        staff.setFirstName(fname);
        staff.setMi(mname);
        staff.setAddress(address);
        staff.setCity(city);
        staff.setState(state);
        staff.setTelephone(telephone);
        staff.setEmail(email);


        sql.insert(staff);
        optionSelect();
    }

    private static void optionUpdate() {
        System.out.println();
        optionSelect();

        if (listSize > 0) {
            Scanner input = new Scanner(System.in);

            System.out.print("Enter an ID of the list to Update: ");
            String id = input.nextLine();
            Staff staff = sql.get(id);

            while(staff.getId() == null) {
                System.out.print("Enter an ID of the list to Update: ");
                id = input.nextLine();

                staff = sql.get(id);
            };
            System.out.println("-".repeat(140));
            System.out.println("Enter a new value or press enter if you want the value in parentheses not to be updated, if you want to delete them add a space.");
            System.out.println("-".repeat(140));

            System.out.print("Last Name (" + staff.getLastName() + "): ");
            lname = input.nextLine();
            System.out.print("First Name (" + staff.getFirstName() + "): ");
            fname = input.nextLine();
            System.out.print("Middle Name (" + staff.getMi() + "): ");
            mname = input.nextLine();
            System.out.print("Address (" + staff.getAddress() + "): ");
            address = input.nextLine();
            System.out.print("City (" + staff.getCity() + "): ");
            city = input.nextLine();
            System.out.print("State (" + staff.getState() + "): ");
            state = input.nextLine();
            System.out.print("Telephone (" + staff.getTelephone() + "): ");
            telephone = input.nextLine();
            System.out.print("Email (" + staff.getEmail() + "): ");
            email = input.nextLine();

            staff.setLastName((lname=="")?staff.getLastName():lname);
            staff.setFirstName((fname=="")?staff.getFirstName():fname);
            staff.setMi((mname=="")?staff.getMi():mname);
            staff.setAddress((address=="")?staff.getAddress():address);
            staff.setCity((city=="")?staff.getCity():city);
            staff.setState((state=="")?staff.getState():state);
            staff.setTelephone((telephone=="")?staff.getTelephone():telephone);
            staff.setEmail((email=="")?staff.getEmail():email);

            sql.update(staff);
            optionSelect();
        }
    }
}
```

```
Console ×
<terminated> Project2 [Java Application] C:\Users\saeed\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.0.v20211012-1059\jre\bin\javaw.exe
1- Select
2- Insert
3- Update
4- Exit
Choose your option : 2
Enter Last Name: saeed
Enter First Name: moghadam
Enter Middle Name:
Enter Address:
Enter City:
Enter State: ontarioooooo
Enter Telephone:
Enter Email:

Error: Data truncation: Data too long for column 'state' at row 1
```

Error message

***Data Update:*** To update the program, it shows us the current data, asking us for the Id to update, the information will be modified only from the entered data.

```
eclipse-workspace - Project2/src/Project2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer × | Project2.java × | Staff.java | StaffDB.java | StaffSQL.java
javaProjects
Project1
Project2
  src
    (default package)
      Project2.java
      Staff.java
      StaffDB.java
      StaffSQL.java
    Project2.java
  JRE System Library [JavaSE-17]
  Referenced Libraries
```

```java
/*
 * AMANJOT SIGNH (8807626)
 * DELMI ALARCON (8787512)
 * SAEED MOGHADAM (8799319)
 * PROG8580 - Computer Programming - Section 1
 * Project 2
 */

import java.util.*;
import static java.lang.System.exit;

public class Project2 {
    // Declare variables to store data to display
    static String sid;
    static String lname;
    static String fname;
    static String mname;
    static String address;
    static String city;
    static String state;
    static String telephone;
    static String email;

    static int listSize = 0;
    static Staff staff = new Staff();
    static StaffSQL sql = new StaffSQL();
    static List<Staff> staffs;

    public static void main(String[] args) throws Exception {
        // Declare options with menu list
        String[] options = {"1- Select",
                            "2- Insert",
                            "3- Update",
                            "4- Exit",
        };

        Scanner input = new Scanner(System.in);

        int option = 1;
        while (option != 4){
            // Call printMenu method sending options
            printMenu(options);
            try {
                option = input.nextInt();
                switch (option){
                    case 1: optionSelect(); break; // Call optionSelect() method
                    case 2: optionInsert(); break; // Call optionInsert() method
                    case 3: optionUpdate(); break; // Call optionUpdate() method
                    case 4: exit(0);
                }
                StaffDB.closeConnection();
            }
            catch (Exception ex){
                System.out.println("Please enter an integer value between 1 and " + options.length);
                input.next();
            }
        }
    }

    public static void printMenu(String[] options){
        for (String option : options){
            System.out.println(option);
        }
        System.out.print("Choose your option : ");
    }

    private static void optionSelect() {
        staffs = sql.list();
        listSize = staffs.size();

        if (listSize > 0) {
            System.out.println("-".repeat(140));
            System.out.printf("%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s", "ID", "Last Name", "First Name", "Mid", "Address", "City", "State", "Telephone", "Email");
            System.out.println();
            System.out.println("-".repeat(140));
            for(Staff staff: staffs) {
                sid = staff.getId();
                lname = staff.getLastName();
                fname = staff.getFirstName();
                mname = staff.getMi();
                address = staff.getAddress();
                city = staff.getCity();
                state = staff.getState();
                telephone = staff.getTelephone();
                email = staff.getEmail();

                String output = "%3s | %-15s | %-15s | %-3s | %-20s | %-10s | %-5s | %-10s | %-20s";
                System.out.println(String.format(output, sid, lname, fname, mname, address, city, state, telephone, email));
            }
            System.out.println("-".repeat(140));
        }
        else {
            System.out.println("\nThere is no data in the Staff table.\n");
        }
    }

    private static void optionInsert() {
        Scanner input = new Scanner(System.in);
        staffs = sql.list();

        int lastIdx = staffs.size() - 1;
        int newId;

        if (lastIdx >= 0) {
            Staff lastElement = staffs.get(lastIdx);
            newId = Integer. parseInt(lastElement.getId()) + 1;
        }
        else {
            newId = 1;
        }

        System.out.print("Enter Last Name: ");
        lname = input.nextLine();
        System.out.print("Enter First Name: ");
        fname = input.nextLine();
        System.out.print("Enter Middle Name: ");
        mname = input.nextLine();
        System.out.print("Enter Address: ");
        address = input.nextLine();
        System.out.print("Enter City: ");
        city = input.nextLine();
        System.out.print("Enter State: ");
        state = input.nextLine();
        System.out.print("Enter Telephone: ");
        telephone = input.nextLine();
        System.out.print("Enter Email: ");
        email = input.nextLine();

        staff.setId(Integer.toString(newId));
        staff.setLastName(lname);
        staff.setFirstName(fname);
        staff.setMi(mname);
        staff.setAddress(address);
        staff.setCity(city);
        staff.setState(state);
        staff.setTelephone(telephone);
        staff.setEmail(email);


        sql.insert(staff);
        optionSelect();
    }

    private static void optionUpdate() {
        System.out.println();
        optionSelect();

        if (listSize > 0) {
            Scanner input = new Scanner(System.in);

            System.out.print("Enter an ID of the list to Update: ");
            String id = input.nextLine();
            Staff staff = sql.get(id);

            while(staff.getId() == null) {
                System.out.print("Enter an ID of the list to Update: ");
                id = input.nextLine();

                staff = sql.get(id);
            };
            System.out.println("-".repeat(140));
            System.out.println("Enter a new value or press enter if you want the value in parentheses not to be updated, if you want to delete them add a space.");
            System.out.println("-".repeat(140));

            System.out.print("Last Name (" + staff.getLastName() + "): ");
            lname = input.nextLine();
            System.out.print("First Name (" + staff.getFirstName() + "): ");
            fname = input.nextLine();
            System.out.print("Middle Name (" + staff.getMi() + "): ");
            mname = input.nextLine();
            System.out.print("Address (" + staff.getAddress() + "): ");
            address = input.nextLine();
            System.out.print("City (" + staff.getCity() + "): ");
            city = input.nextLine();
            System.out.print("State (" + staff.getState() + "): ");
            state = input.nextLine();
            System.out.print("Telephone (" + staff.getTelephone() + "): ");
            telephone = input.nextLine();
            System.out.print("Email (" + staff.getEmail() + "): ");
            email = input.nextLine();

            staff.setLastName((lname=="")?staff.getLastName():lname);
            staff.setFirstName((fname=="")?staff.getFirstName():fname);
            staff.setMi((mname=="")?staff.getMi():mname);
            staff.setAddress((address=="")?staff.getAddress():address);
            staff.setCity((city=="")?staff.getCity():city);
            staff.setState((state=="")?staff.getState():state);
            staff.setTelephone((telephone=="")?staff.getTelephone():telephone);
            staff.setEmail((email=="")?staff.getEmail():email);

            sql.update(staff);
            optionSelect();
```

```
193         }
194     }
195 }
```

Console ×

Project [Java Application] C:\Program Files\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe (Apr. 12, 2022, 8:12:27 p.m.)

```
1- Select
2- Insert
3- Update
4- Exit
Choose your option : 3

-----------------------------------------------------------------------------------------------------------------------------
 ID | Last Name       | First Name      | Mid | Address             | City       | State | Telephone  | Email
-----------------------------------------------------------------------------------------------------------------------------
  1 | Alarcon         | Delmi           | E   | 1425 BlockLine      | Kitchener  | ON    | 2262014387 | delmialarcon@gmail.com
  2 | Moghadam        | Saeed           |     | 475 King Street     | Kitchener  | ON    | 6538743298 | saeedmoghadam@gmail.com
  3 | Signh           | Amanjot         |     | 878 Ottawa Street   | Toronto    | ON    | 5195882903 | amanjotsignh@gmail.com
  4 | Tripathy        | Dikshya         | A   | 527 University Av   | Hamilton   | ON    | 3295491248 | dikshyatri@gmail.com
-----------------------------------------------------------------------------------------------------------------------------
Enter an ID of the list to Update: 3
-----------------------------------------------------------------------------------------------------------------------------
Enter a new value or press enter if you want the value in parentheses not to be updated, if you want to delete them add a space.
-----------------------------------------------------------------------------------------------------------------------------
Last Name (Signh):
First Name (Amanjot): Neha
Middle Name (): S
Address (878 Ottawa Street): 870 Ottawa Street
City (Toronto):
State (ON):
Telephone (5195882903):
Email (amanjotsignh@gmail.com):
-----------------------------------------------------------------------------------------------------------------------------
Updated!
-----------------------------------------------------------------------------------------------------------------------------
 ID | Last Name       | First Name      | Mid | Address             | City       | State | Telephone  | Email
-----------------------------------------------------------------------------------------------------------------------------
  1 | Alarcon         | Delmi           | E   | 1425 BlockLine      | Kitchener  | ON    | 2262014387 | delmialarcon@gmail.com
  2 | Moghadam        | Saeed           |     | 475 King Street     | Kitchener  | ON    | 6538743298 | saeedmoghadam@gmail.com
  3 | Signh           | Neha            | S   | 870 Ottawa Street   | Toronto    | ON    | 5195882903 | amanjotsignh@gmail.com
  4 | Tripathy        | Dikshya         | A   | 527 University Av   | Hamilton   | ON    | 3295491248 | dikshyatri@gmail.com
-----------------------------------------------------------------------------------------------------------------------------
```