MỤC LỤC

[MỞ ĐẦU 2](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570975)

[Chương 1 LÝ THUYẾT CƠ SỞ 3](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570976)

[1.1 Lập trình hướng đối tượng 3](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570977)

[1.1.1 Khái niệm về lập trình hướng đối tượng trong java 3](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570978)

[1.1.2 Tính chất của lập trình hướng đối tượng trong java 3](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570979)

[1.1.3 Các ưu điểm của lập trình hướng đối tượng 4](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570980)

[1.2 Đăng kết với JDBC 4](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570986)

[1.3 Kết nối cơ sở dữ liệu trong java 5](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570987)

[1.3.1 Nạp trình điều khiển 5](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570988)

[1.3.2 Tạo kết nối 5](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570989)

[Chương 2 Lập trình chương trình ”Quản lý Sinh Viên” 9](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570990)

[2.1 Mô tả yêu cầu của bài toán 9](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570991)

[2.2 Tổ chức chương trình 9](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570992)

[2.2.1 Xây dựng cơ sở dữ liệu DOAN\_JAVA 9](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570993)

[Chương 3 Chạy Demo chương trình 10](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570995)

[3.1 Các Class và công dụng 10](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570996)

[3.2 Chạy chương trinh 11](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570997)

[KẾT LUẬN 17](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61570999)

[TÀI LIỆU THAM KHẢO 18](file:///D:\download\BTLJVOOP\baocao_BTLLJVOOP_NTThanh.doc#_Toc61571001)

# MỞ ĐẦU

Trong thời đại CNTT phát triển như hiện nay, thì các nhà lập trình luôn phải suy nghĩ xem làm thế nào để tạo ra những chương trình hay, tiện ích...những chương trình tin học ứng dụng ngày càng nhiều, rất nhiều công việc thủ công trước đây đã được xử lý bằng các phần mềm chuyên dụng đã giảm đáng kể công sức, nhanh chóng và chính xác. Để có một phần mềm ứng dụng đáp ứng được yêu cầu công việc đặt ra thì những người làm tin học phải biết phân tích thiết kế hệ thống làm việc của chương trình để từ đó xây dựng lên một phần mềm ứng dụng quản lý chương trình đó. Khác với việc quản lý theo phương pháp thủ công truyền thống, việc quản lý bằng máy tính đã khắc phục được những khó khăn và yếu kém của quản lý theo phương pháp truyền thống, đó là giảm được số lượng người tham gia quản lý, sự vòng vèo trong các quy trình xử lý, tốc độ việc cập nhật và lấy thông tin tăng lên rất nhiều, thông tin tập trung và gọn nhẹ không cồng kềnh, việc quản lý bằng máy cũng giảm tối thiểu những sai sót.

Lập trình cho hệ thống quản lý sinh viên phục vụ cho giảng viên, giáo viên

Nội dung tiểu luận:

Tìm hiểu về lập trình hướng đối tượng trong môi trường Java.

Kết nối cơ sở dữ liệu bằng Java

Chương trình Demo.

**LỜI CẢM ƠN**

Để hoàn thành tiểu luận này, em xin chân thành cảm ơn sự giảng dạy và hướng dẫn của thầy giáo Nguyễn Hà Huy Cường. Tuy bản thân đã rất cố gắng xong vẫn không tránh khỏi thiếu sót, rất mong mong nhận được sự góp ý của Thầy và các bạn.

Xin chân thành cảm ơn!

*Đà Nẵng, ngày 17 tháng 1 năm 2021*

*Người thực hiện*

Nguyễn Võ Hoàng Long

# Chương 1 LÝ THUYẾT CƠ SỞ

## Lập trình hướng đối tượng

### Khái niệm về lập trình hướng đối tượng trong java

Lập trình hướng đối tượng (Object Oriented Programing – OOP) là một phương pháp để thiết kế một chương trình bởi sử dụng các lớp và các đối tượng.

Java là một ngôn ngữ lập trình hướng đối tượng vì vậy nó cũng hỗ trợ các đặc tính của lập trình hướng đối tượng:

* Đa hình (Polymorphism)
* Thừa kế (Inheritance)
* Đóng gói (Encapsulation)
* Trừu tượng (Abstraction)

### Tính chất của lập trình hướng đối tượng trong java

**Tính đóng gói (encapsulation) và che giấu thông tin (information hiding)**

* Tức là trạng thái của đối tượng được bảo vệ không cho các truy cập từ code bên ngoài như thay đổi trong thái hay nhìn trực tiếp. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn, bảo mật của đối tượng Trong Java, tính đóng gói được thể hiện thông qua phạm vi truy cập (access modifier). Ngoài ra, các lớp liên quan đến nhau có thể được gom chung lại thành package.

**Tính kế thừa (Inheritance)**

1. Tính kế thừa là khả năng cho phép ta xây dựng một lớp mới dựa trên các định nghĩa của một lớp đã có. Lớp đã có gọi là lớp Cha, lớp mới phát sinh gọi là lớp Con và đương nhiên kế thừa tất cả các thành phần của lớp Cha, có thể chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

**Tính đa hình (polymorphism)**

1. Khi một tác vụ được thực hiện theo nhiều cách khác nhau được gọi là tính đa hình.
2. Đối với tính chất này, nó được thể hiện rõ nhất qua việc gọi phương thức của đối tượng. Các phương thức hoàn toàn có thể giống nhau, nhưng việc xử lý luồng có thể khác nhau. Nói cách khác: Tính đa hình cung cấp khả năng cho phép người lập trình gọi trước một phương thức của đối tượng, tuy chưa xác định đối tượng có phương thức muốn gọi hay không. Đến khi thực hiện (run-time), chương trình mới xác định được đối tượng và gọi phương thức tương ứng của đối tượng đó. Kết nối trễ giúp chương trình được uyển chuyển hơn, chỉ yêu cầu đối tượng cung cấp đúng phương thức cần thiết là đủ.
3. Trong Java, chúng ta sử dụng nạp chồng phương thức (method overloading) và ghi đè phương thức (method overriding) để có tính đa hình.
4. **Nạp chồng (Overloading)**: Đây là khả năng cho phép một lớp có nhiều thuộc tính, phương thức cùng tên nhưng với các tham số khác nhau về loại cũng như về số lượng. Khi được gọi, dựa vào tham số truyền vào, phương thức tương ứng sẽ được thực hiện.
5. **Ghi đè (Overriding)**: là hai phương thức cùng tên, cùng tham số, cùng kiểu trả về nhưng thằng con viết lại và dùng theo cách của nó, và xuất hiện ở lớp cha và tiếp tục xuất hiện ở lớp con. Khi dùng override, lúc thực thi, nếu lớp Con không có phương thức riêng, phương thức của lớp Cha sẽ được gọi, ngược lại nếu có, phương thức của lớp Con được gọi.

**Tính trừu tượng (abstraction)**:

* Tính trừu tượng là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép bạn loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.
* Tính trừu tượng giúp bạn tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.
* Trong Java, chúng là sử dụng abstract class và abstract interface để có tính trừu tượng.

### Các ưu điểm của lập trình hướng đối tượng

1. Dựa trên nguyên lý kế thừa, trong quá trình mô tả các lớp có thể loại bỏ những chương trình bị lặp, dư. Và có thể mở rộng khả năng sử dụng các lớp mà không cần thực hiện lại. Tối ưu và tái sử dụng code hiệu quả.
2. Đảm bảo rút ngắn thời gian xây dựng hệ thống và tăng năng suất thực hiện.
3. Sự xuất hiện của 2 khái niệm mới là lớp và đối tượng chính là đặc trưng của phương pháp lập trình hướng đối tượng. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.

## Đăng kết với JDBC

JDBC (Java Database Connectivity) là một API chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ (database relationship). JDBC bao gồm một tập hợp các class và các interface dùng cho ứng dụng Java có thể giao tiếp với các cơ sở dữ liệu (database) khác nhau.



*Hình 1.2:Mô hình JDBC*

## Kết nối cơ sở dữ liệu trong java

### Nạp trình điều khiển

*import java.sql.Connection;*

*import java.sql.DriverManager;*

*import java.sql.PreparedStatement;*

*import java.sql.ResultSet;*

java.sql.Connection connection = null;

### Tạo kết nối

1. Ta tạo đối tượng Connection (có tên là conn) bằng cách gọi phương thức getConnection của lớp DriverManager như sau:

Connection conn = null;

conn=DriverManager.getConnection(url,”myLogicName”,”myPassword”)

Trong đó:

*url*: là một chuỗi nêu lên đặc điểm của CSDL có dạng: jdbc:subprotocol:subname

*subprotocol*: là giao thức con tương ứng với loại CSDL

*subname*: là tên CSDL (DataSource)

“*myLoginName*”: là tên người dùng khi đăng nhập vào CSDL

“*myPassword*”: là mật khẩu người dùng khi đăng nhập vào CSDL

1. Tạo đối tượng Statement hoặc PreparedStatement

Tất cả các lệnh dùng để tác động lên CSDL đều phải thông qua đối tượng Statement hoặc một đối tượng cùng tính chất như PreparedStatement hay CallableStatement

Tạo đối tượng Statement từ kết nối conn:

Statement stmt = conn.createStatement();

Hoặc

CallableStatement Callstmt=conn.prepareCall();

PreparedStatement Prepstmt=conn.prepareStatement();

Gọi thực thi lệnh sql như sau:

Statement stmt=conn.createStatement();

stmt.executeUpdate(sSQL);

với sSQL là câu lệnh sql

--- code demo

try {

Class.*forName*("com.microsoft.sqlserver.jdbc.SQLServerDriver");

String connectionURL = "jdbc:sqlserver://PHU\\SQLEXPRESS:1433;databaseName=DO\_AN\_JAVA;integratedSecurity=true;";

connection = DriverManager.*getConnection*(connectionURL, "sa", "123");

System.***out***.println("Kết nối sql Thành Công :))");

String sql = “SELECT \* FROM GIANGVIEN”

PreparedStatement dsql = connection.prepareStatement(sql);

dsql.execute();

return "success";

}catch (Exception e) {

System.***err***.println("Kết nối sql thất bại :((");

e.printStackTrace();

}

# Chương 2 Lập trình chương trình ”Quản lý sinh viên”

## 2.1 Mô tả yêu cầu của bài toán

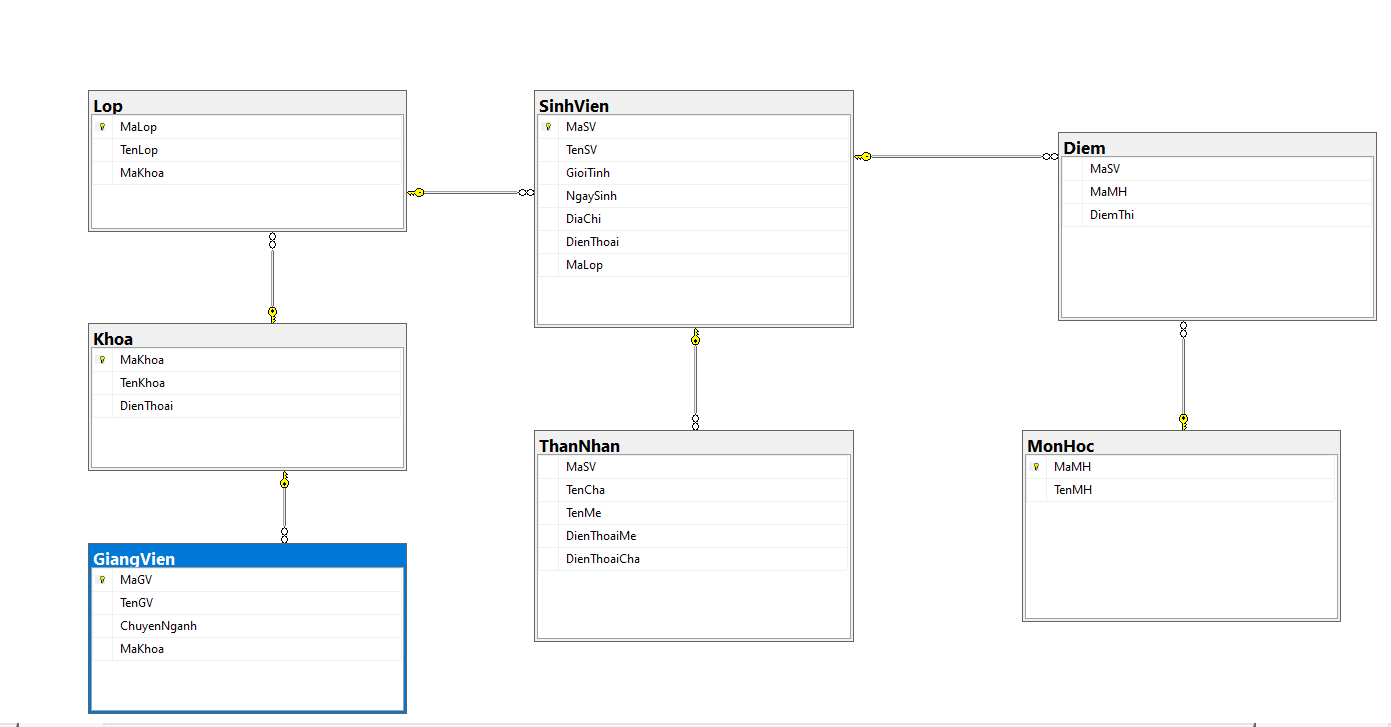
Xây dựng CSDL DOAN\_JAVA (Bài tập lớn java OOP) đặt tại Server.

Thực hiện một giao dịch quản lý sinh viên tại máy Client tác động lên cơ sở dữ liệu .

## 2.2 Tổ chức chương trình

### 2.2.1 Xây dựng cơ sở dữ liệu DOAN\_JAVA

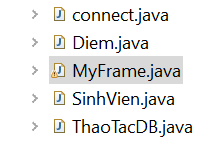
Cơ sở dữ liệu DOAN\_JAVA được xây dựng với mục đích để quản lý sinh viên:



*Hình 2.2-1: Cơ sở dữ liệu DOAN\_JAVA*

# Chương 3 Chạy Demo chương trình

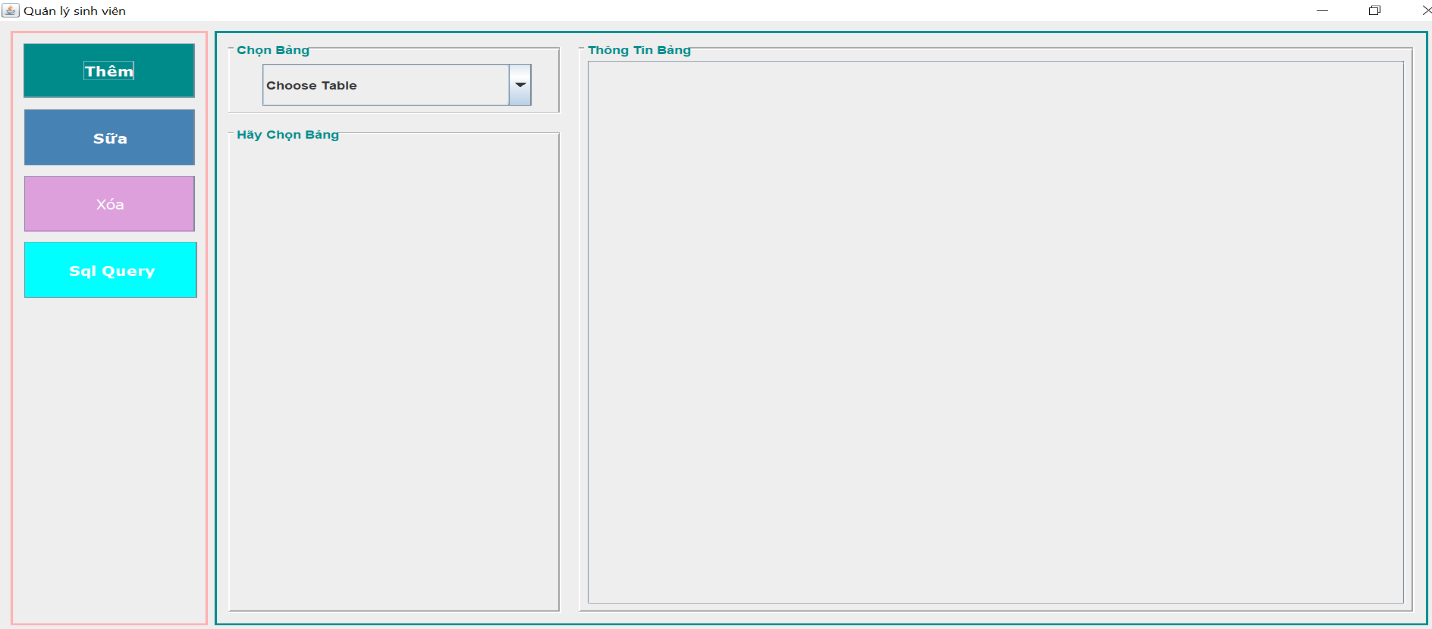
## 3.1 Các Class và công dụng



*Hình 3.1.1: Các Class trong đồ án Java Nâng Cao*

1. connect.java : kết nối java với SQL Server để lấy dữ liệu
2. Diem.java : chứa các phương thức getter, setter và hàm tạo của Điểm
3. SinhVien.java : chứa các phương thức getter, setter và hàm tạo của Sinh Viên
4. ThaoTacDB.java : chứa các hàm insert, update, delele để truy vấn dữ liệu từ cơ sở dữ liệu
5. MyFrame.java : giao diện chính của chương trình

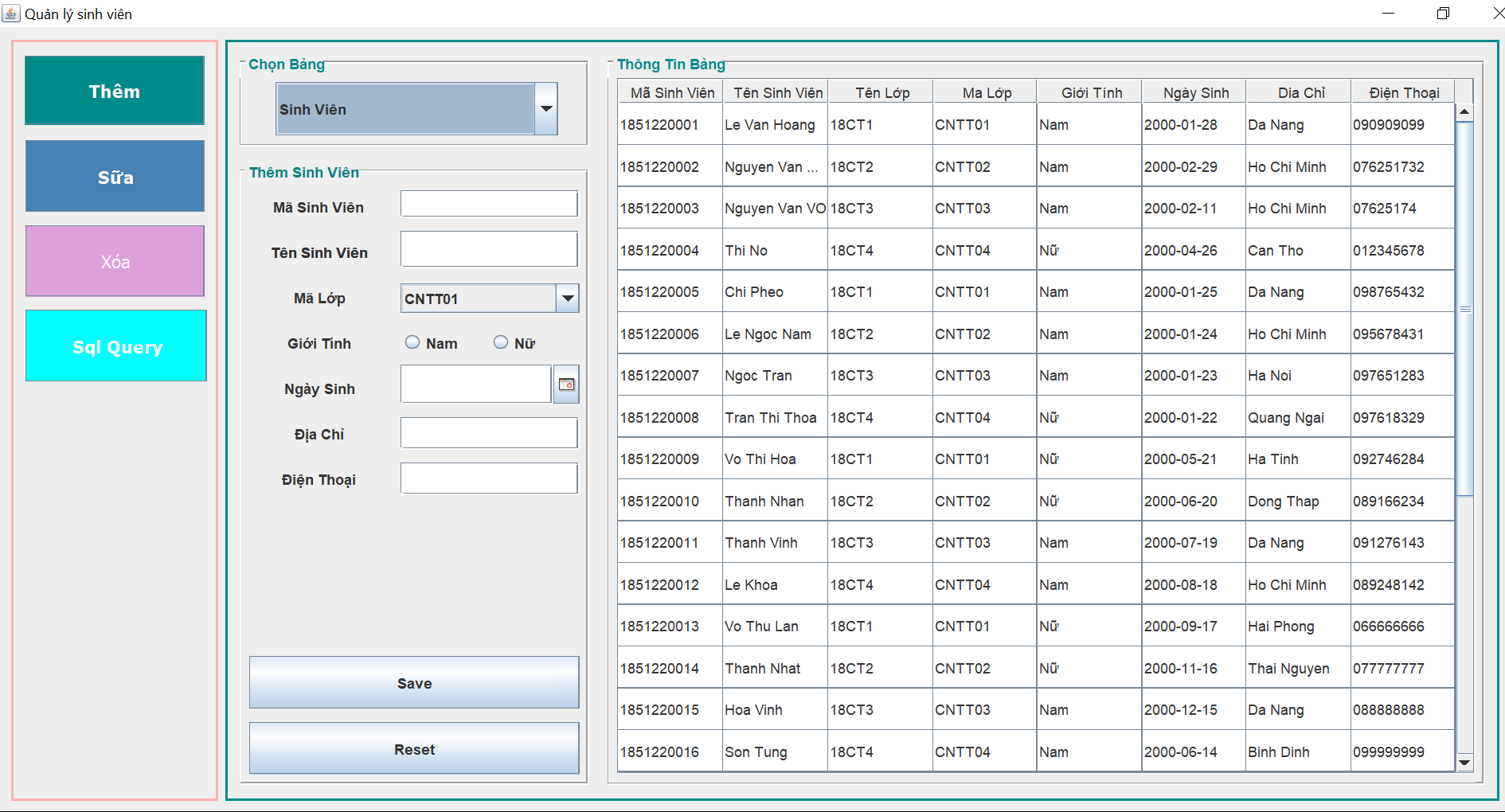
## 3.2 Chạy chương trình:



*Hình 3.2-1: giao diện chính khi chạy chương trình*

Khi chạy chương trình chính sẽ xuất hiện giao diện như trên. Chương trình gồm có 4 nút chính Thêm, Sữa, Xóa, Tìm Kiếm, SQL query (thực thi câu truy vấn tại Frame ) và các Jcombobox để chọn bảng.

Ở nút Thêm, khi click vào nút Jcombobox chọn bảng thì sẽ có danh sách các bảng của database, các bảng gồm có các nút Save , Reset thì sẽ xuất hiện giao diện sau:



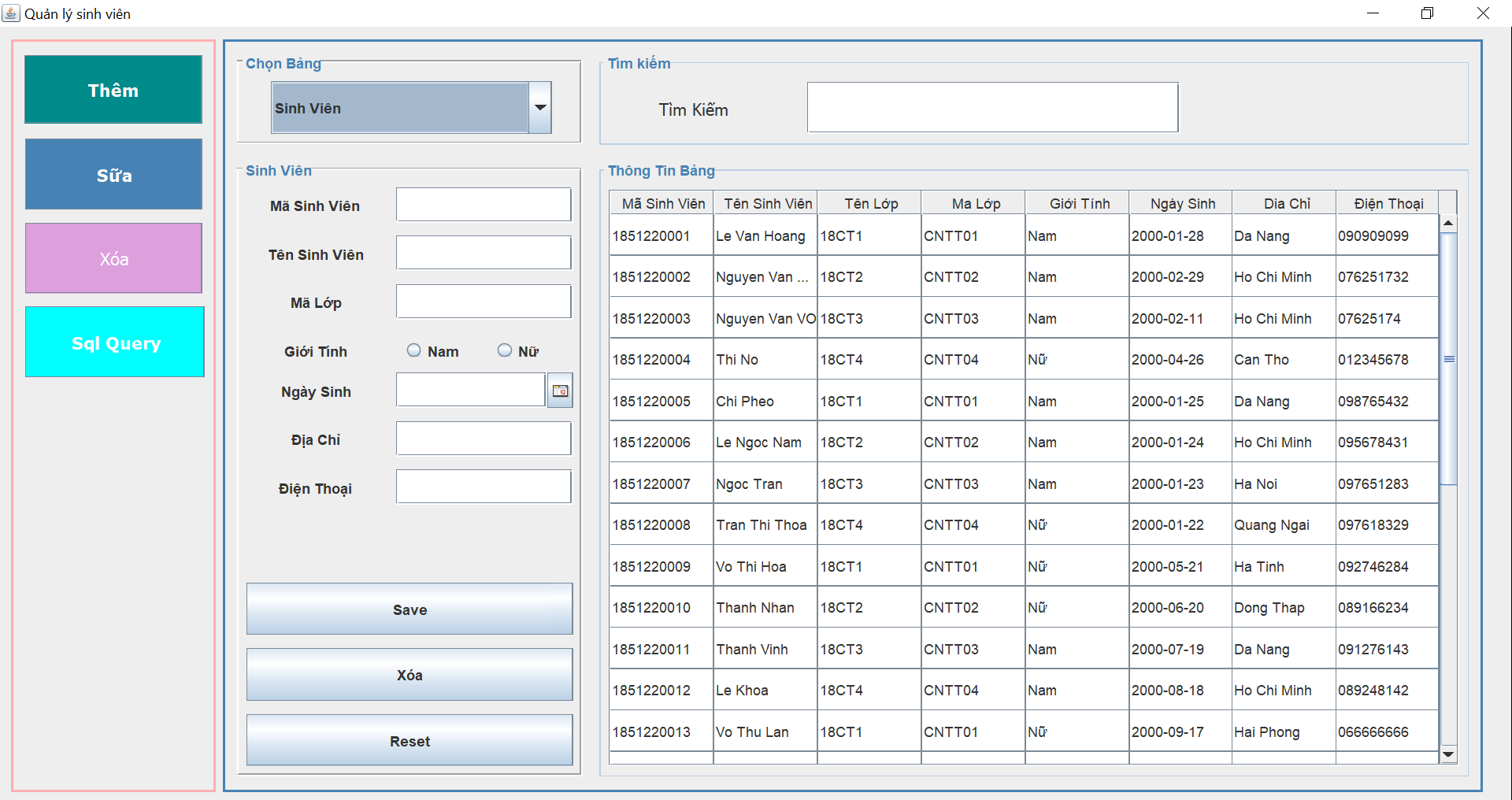
*Hình 3.2-2: giao diện khi chọn bảng* ***sinh viên***

Khi thêm thành công một sinh viên nào đó thì sẽ có một thông báo như giao diện ở dưới



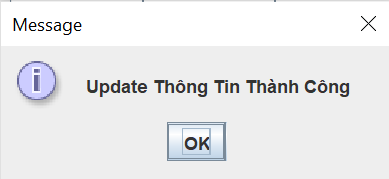
*Hình 3.3-3: thông báo thành công khi thêm sinh viên*

Ở nút Sửa khi chọn bảng Sinh Viên gồm các nút Save, Xóa, Reset và một ô input để điền thông tin tìm kiếm



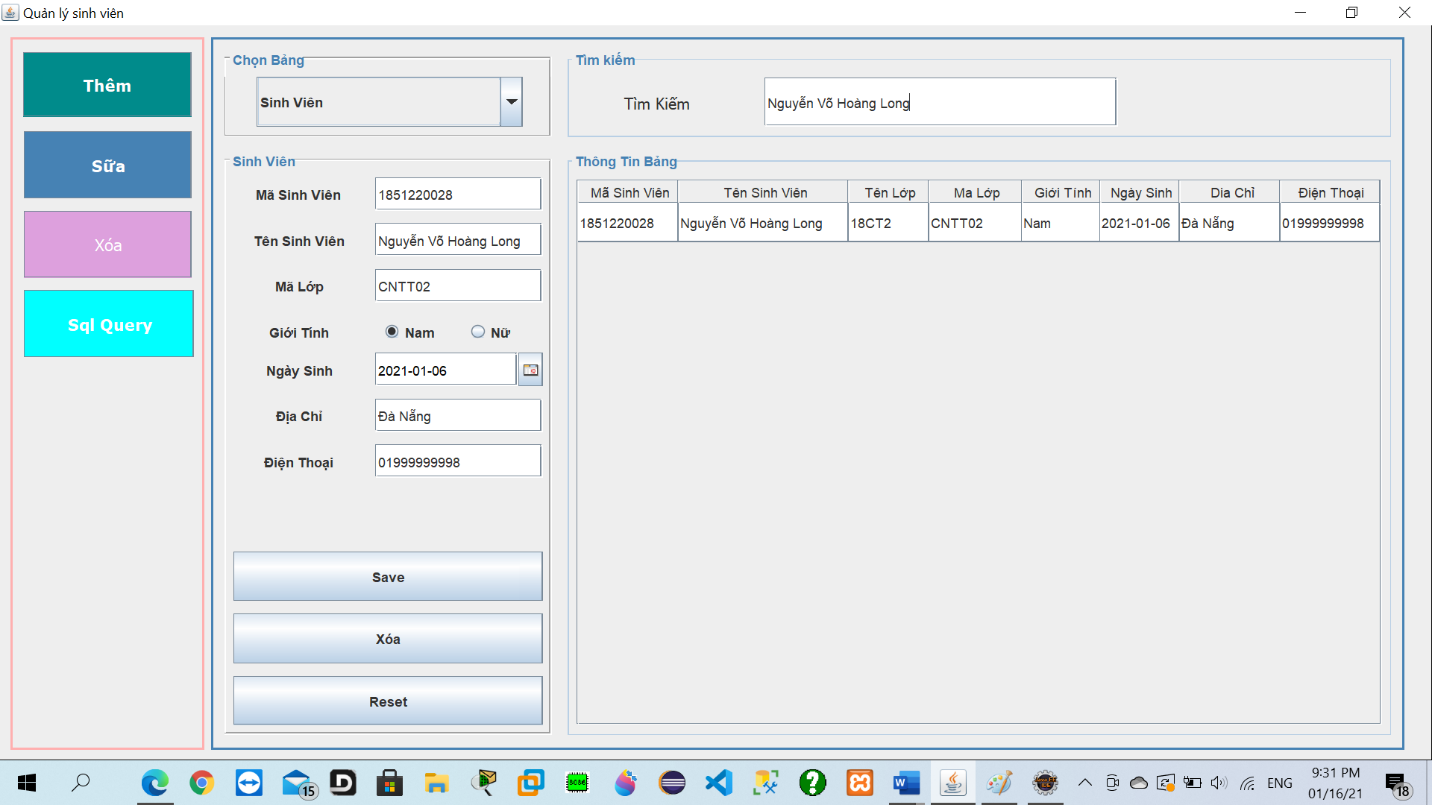
*Hình 3.2-4: giao diện khi chọn bảng sinh viên để sửa*

Khi sữa thành công sẽ có thông báo có giao diện:

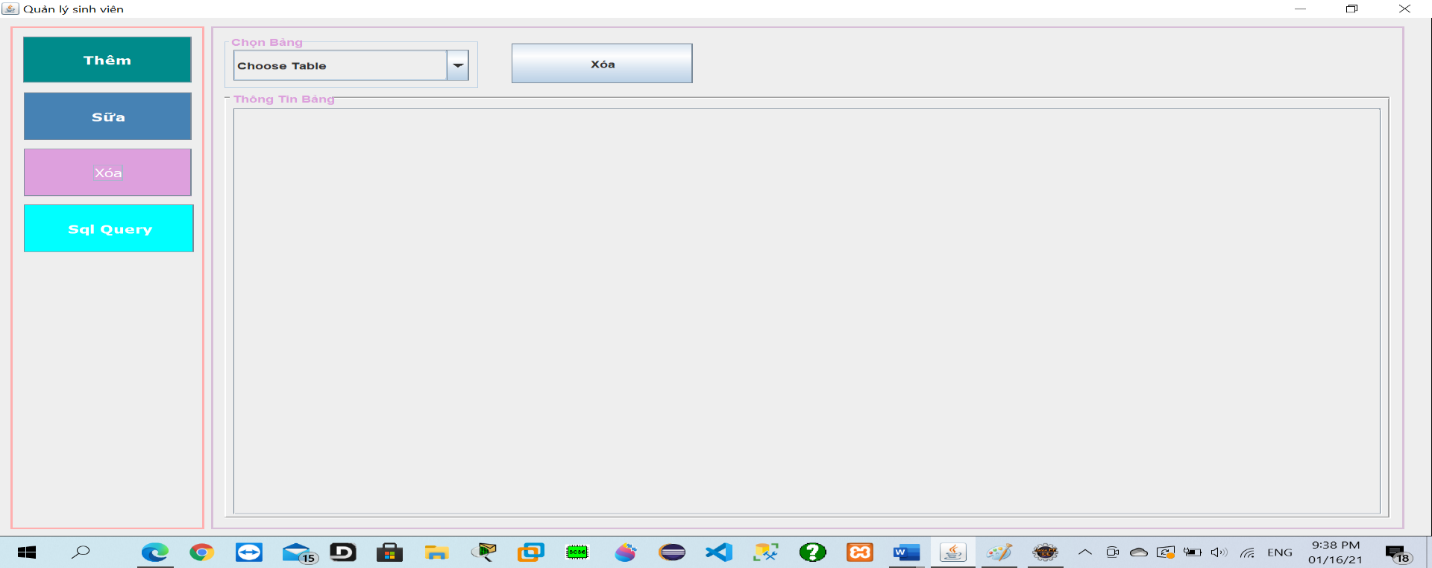


*Hình 3.2-5: thông báo thành công khi sửa sinh viên*

Khi bạn tìm kiếm một sinh viên chỉ cần gõ thông tin vào ô input dưới đây là giao diện khi tìm kiếm tên bạn sinh viên:

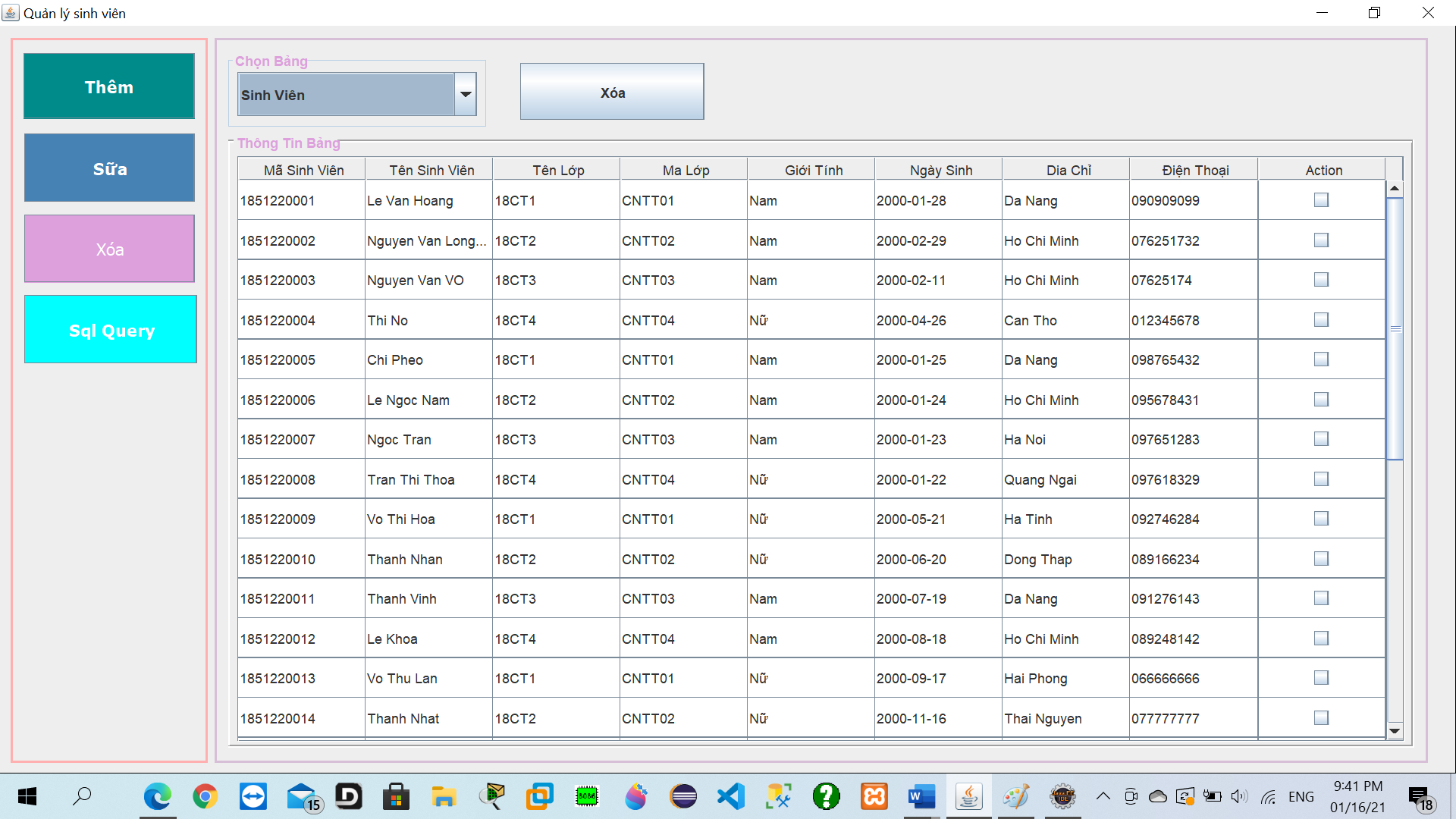


*Hình3.3-6 : tìm kiếm sinh viên*

Khi ở nút xóa có jcombobox lấy các danh sách các bảng trong cơ sở dữ liệu và một nút xóa để xóa các sinh viên được chọn. 

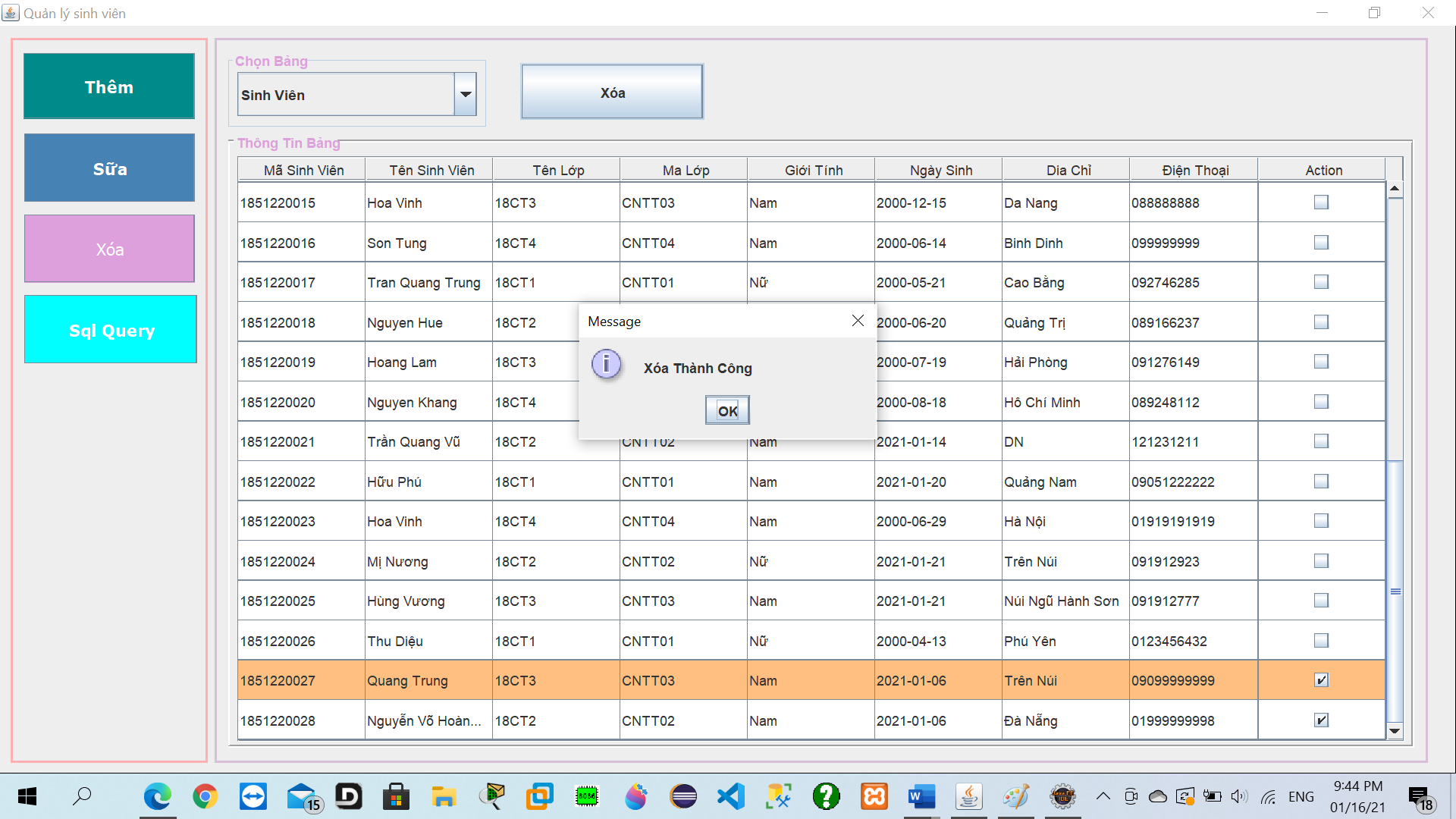
*Hình 3.3-7: giao diện khi ở nút xóa*

Khi chọn bảng Sinh Viên thì sẽ có thêm một cột có tên là Action để có thể chọn một hoặc nhiều sinh viên để xóa dưới đây là giao diện:



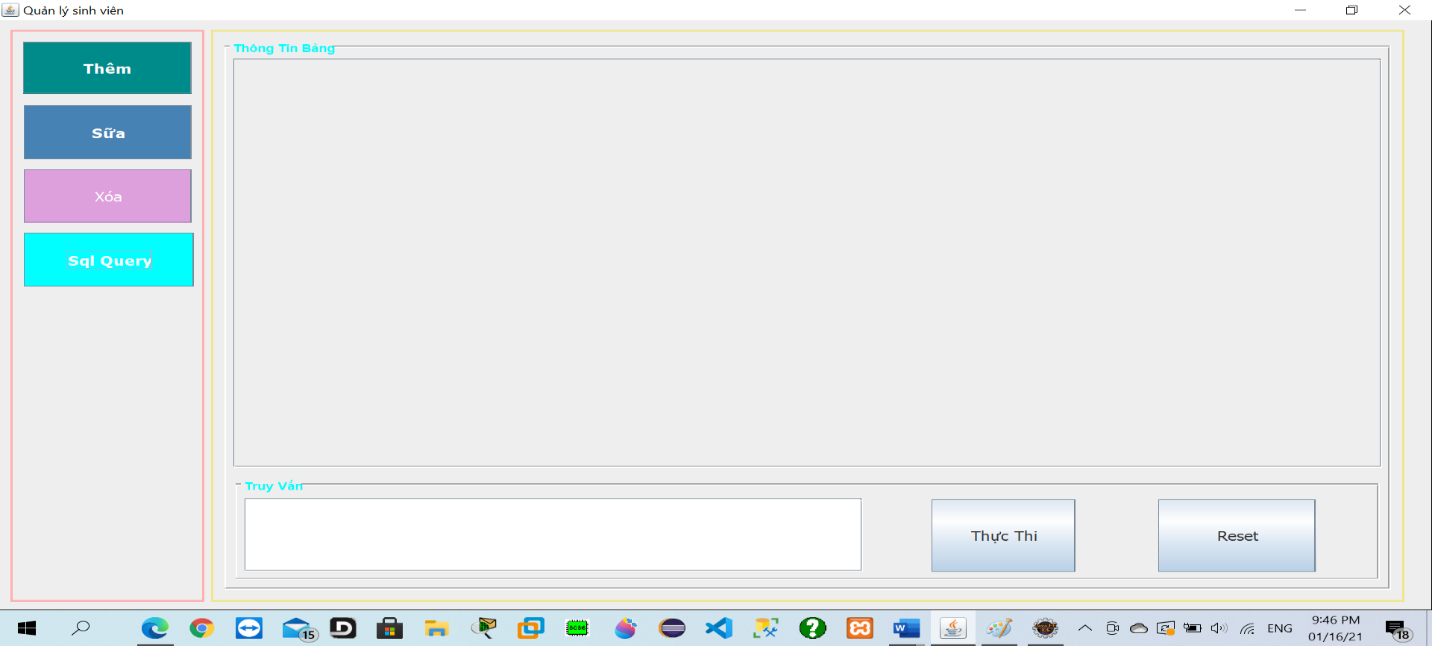
*Hình 3.3-8: Giao diện bảng Sinh Viên có thể xóa một hoặc nhiều.*

Sau khi bấm nút xóa thì sẽ có giao diện:

**

*Hình3.3-9 : giao diện sau khi xóa nhiều sinh viên một lúc.*

Khi ở nút SQL Query có một textarea để điền câu truy vấn sql server, một Jpanel để hiện thông tin bảng sau khi truy vấn và có một nút thực thi để thực hiện câu truy vấn có trong textarea và một nút reset, dưới dây là giao diện:



*Hình 3.3-10 giao diện thực thi câu lệnh sql server*

# KẾT LUẬN

Qua thời gian tìm hiểu về lập trình hướng đối tượng nói chung và thực hiện bài toán **Lập trình quản lý sinh viên.** đề tài đã đạt được một số yêu cầu sau:

Về lý thuyết

Tìm hiểu về lập trình hướng đối tượng trong môi trường Java.

Kết nối cơ sở dữ liệu bằng Java

Về thực hành: Sử dụng được ngôn ngữ Java trong lập trình mạng và kết nối cơ sở dữ liệu.

Tuy nhiên do thời gian và khả năng còn hạn chế nên đề tài chỉ dừng lại ở mức độ tìm hiểu lý thuyết, chưa triển khai lập trình giao dịch trên hệ thống đa Server với cơ sở dữ liệu phân tán.

Tôi xin chân thành cảm ơn ý kiến đóng góp của thầy giáo Nguyễn Hà Huy Cường và các bạn quan tâm để đề tài được hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

---🙠🕮🙢---

[1] TS. Lê Văn Sơn (2002), *Hệ tin học phân tán*, nhà xuất bản Đại học quốc Gia TP. Hồ Chí Minh.

[2] Nguyễn Tiến, Ngô Quốc Việt, Phạm Nguyễn Tuấn Kỳ (2002), *Giáo trình lý thuyết và thực hành Java 2*, nhà xuất bản thống kê.

[3] Trần Tiến Dũng (1999), *Giáo trình lý thuyết và bài tập*, Nhà xuất bản giáo dục.

[4] Nguyễn Tiến, Đặng Xuân Hương, Nguyễn Văn Hoài (1999), *Giáo trình căn bản về mạng*, Nhà xuất bản giáo dục.

[5] Nguyễn Phương Lan, Hoàng Đức Hải (2006), *Java lập trình mạng*, NXB Lao động xã hội .