

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo bộ môn
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
ĐỀ TÀI:
QUẢN LÝ SINH VIÊN



Giáo Viên: TS. Nguyễn Thành Sơn

Sinh viên thực hiện:

Đoàn Huỳnh Thiện

MSSV: 15110129

Nguyễn Di Thông

MSSV: 15110383

Mục lục

1. Tổng quan về phần mềm.....	5
1.1 Mục tiêu	5
1.2 Phạm vi	5
1.3 Mô tả.....	5
1.4 Thông tin của phần mềm	6
1.5 Chức năng của phần mềm.....	6
2. Lược đồ quan hệ	6
2.1 Mô hình ERD	6
2.2 Mô tả cơ sở dữ liệu	7
2.3 Biểu đồ quan hệ giữa các bảng	11
3. Danh sách các Stored Procedure, Function, Trigger	11
3.1 Stored Procedure	11
3.1.1 Thêm 1 bảng điểm.....	11
3.1.2 Thêm 1 môn học.....	12
3.1.3 Thêm 1 ngành học	13
3.1.4 Thêm 1 sinh viên	13
3.1.5 Xóa 1 bảng điểm.....	14
3.1.6 Xóa 1 môn học	15
3.1.7 Xóa 1 ngành học.....	15
3.1.8 Xóa 1 sinh viên.....	16
3.1.9 Lấy thông tin sinh viên tương ứng mã lớp	17
3.1.10 Lấy thông tin chi tiết điểm từng môn	17
3.1.11 Lấy thông tin lớp học tương ứng mã lớp	18
3.1.12 Lấy mã khoa.....	19
3.1.13 Lấy mã lớp	20

3.1.14 Lấy mã môn học	20
3.1.15 Lấy mã sinh viên	21
3.1.16 Lấy tên khoa tương ứng mã khoa.....	21
3.1.17 Lấy tên môn học tương ứng mã môn học	22
3.1.18 Sửa điểm của bảng điểm.....	22
3.1.19 Sửa thông tin môn học	23
3.1.20 Sửa thông tin ngành	24
3.1.21 Sửa thông tin sinh viên	24
3.2 Function	25
3.2.1 Lấy thông tin tất cả sinh viên.....	25
3.2.2 Lấy thông tin tất cả môn học.....	26
3.2.3 Lấy thông tin sinh viên tương ứng mã sinh viên	27
3.2.4 Tìm kiếm trong bảng sinh viên	27
3.2.5 Tìm kiếm trong bảng môn học	28
3.2.6 Tìm kiếm trong bảng ngành học	29
3.2.7 Lấy thông tin tất cả ngành học	29
3.2.8 Tính điểm trung bình tương ứng mã sinh viên	30
3.2.9 Đếm số lượng môn học.....	31
3.2.10 Đếm số lượng ngành học	31
3.2.11 Đếm số lượng sinh viên	32
3.3 Trigger.....	32
3.3.1 Kiểm tra khi nhập điểm	32
3.3.2 Thông báo khi cập nhật điểm.....	33
3.3.3 Kiểm tra khi nhập thông tin môn học.....	33
3.3.4 Thông báo khi xóa môn học.....	34
3.3.5 Thông báo khi cập nhật thông tin môn học	34
3.3.6 Kiểm tra nhập thông tin ngành học khi muốn thêm hoặc cập nhật.....	34
3.3.7 Thông báo khi xóa ngành học	35
3.3.8 Thông báo khi cập nhật thông tin ngành học.....	35

3.3.9 Kiểm tra nhập thông tin sinh viên khi muốn thêm hoặc cập nhật	35
3.3.10 Thông báo khi xóa sinh viên	36
3.3.11 Thông báo khi cập nhật thông tin sinh viên.....	36
3.3.12 Kiểm tra giới tính khi thêm hoặc cập nhật.....	36
4. Phân quyền	37
4.1 Tạo user và role	37
4.2 Phân quyền quản trị	37
4.3 Phân quyền giảng viên.....	38

1. Tổng quan về phần mềm

1.1 Mục tiêu

Ngày nay, tin học đã có những bước tiến nhanh chóng về ứng dụng của nó trong mọi lĩnh vực của cuộc sống trên phạm vi toàn thế giới nói chung và Việt Nam nói riêng. Tin học được người ta quan tâm và nhắc đến nhiều hơn bao giờ hết vì nó là một phần không thể thiếu trong cuộc sống văn minh, góp phần đẩy mạnh công cuộc công nghiệp hoá hiện đại hoá đất nước, tiến đến nền kinh tế tri thức. Máy vi tính cùng với những phần mềm là công cụ đắc lực giúp ta quản lý, tổ chức, sắp xếp và xử lý công việc một cách nhanh chóng và chính xác. Ở Việt Nam hiện nay, máy tính điện tử đặc biệt là máy vi tính trong nhiều năm qua đã được sử dụng rất rộng rãi. Sự phát triển của tin học, các công nghệ phần mềm, phần cứng, các tài liệu tham khảo đã đưa chúng ta từng bước tiếp cận với công nghệ thông tin trong mọi lĩnh vực nhằm đáp ứng nhu cầu của con người. Quản lý sinh viên là một đề tài không còn mới mẻ với các bài toán quản lý. Việc đưa tin học vào ứng dụng để quản lý là rất hữu ích, vì chúng ta phải bỏ ra rất ít thời gian mà lại thu được hiệu quả cao, rất chính xác và tiện lợi nhanh chóng.

Với việc đưa vào hoạt động phần mềm quản lý sinh viên hướng tới có thể đáp ứng nhu cầu của toàn bộ sinh viên và cán bộ giảng viên của trường có thể tra cứu điểm và danh sách viên một cách nhanh chóng hiệu quả nhất.

1.2 Phạm vi

Phạm vi đề tài áp dụng trong trường đại học. Các chức năng tập trung vào việc quản lý các thông tin cần thiết của trường đại học.

Phần mềm có thể sử dụng riêng lẻ (sử dụng cá nhân) hoặc nhiều người dùng trên nhiều máy tính khác nhau (chạy trên mạng LAN với hệ thống máy chủ nội bộ).

1.3 Mô tả

Trong trường mỗi sinh viên có một mã sinh viên duy nhất để phân biệt với các sinh viên khác. Ngoài ra cũng cần biết các thông tin khác như tên, giới tính, ngày sinh, nơi sinh, địa chỉ, học bổng. Trường có nhiều lớp được phân biệt bởi mã lớp, ngoài ra còn

các thông tin như tên lớp, và mỗi sinh viên chỉ thuộc 1 lớp duy nhất. Một lớp sẽ chỉ thuộc một khóa học phân biệt bởi mã khóa học và chỉ thuộc một hệ đào tạo duy nhất được phân biệt bởi mã hệ. Các khóa học có thêm thông tin tên khóa học. Hệ đào tạo còn thông tin tên hệ đào tạo. Một ngành học có nhiều lớp, nhưng một lớp chỉ thuộc 1 ngành học. Thông tin ngành học gồm mã ngành, tên ngành. Một khoa sẽ có nhiều ngành học, nhưng một ngành chỉ thuộc 1 khoa. Thông tin khoa gồm mã khoa, tên khoa. Mỗi sinh viên sẽ học nhiều môn để có thể đủ điều kiện tốt nghiệp. Mỗi môn học có thông tin như mã môn học, tên môn học, số tiết lý thuyết, số tiết thực hành.

1.4 Thông tin của phần mềm

Tên phần mềm: Quản lý sinh viên

Ngôn ngữ: C#

Phần mềm hỗ trợ: DevExpress, SQL Server

1.5 Chức năng của phần mềm

1.5.1 Quản lý danh sách sinh viên

Thêm, cập nhật thông tin, xóa sinh viên, tìm kiếm sinh viên theo từ khóa bất kì.

1.5.2 Quản lý bảng điểm

Thêm, cập nhật, xóa điểm của 1 môn học

1.5.3 Quản lý danh sách ngành học

Thêm, cập nhật thông tin, xóa ngành học, tìm kiếm ngành học theo từ khóa bất kì.

1.5.4 Quản lý danh sách môn học

Thêm, cập nhật thông tin, xóa môn học, tìm kiếm môn học theo từ khóa bất kì.

2. Lược đồ quan hệ

2.1 Mô hình ERD

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaLop	Text	Là khóa chính để phân biệt với các lớp, mỗi lớp có 1 mã lớp
2	TenLop	Text	Tên của lớp học
3	MaKhoaHoc	Text	Mã khóa học phân biệt từng khóa học (Khóa ngoại)
4	MaHe	Text	Mã hệ (Khóa ngoại)
5	MaNganh	Text	Mã ngành học (Khóa ngoại)

b. Mô tả bảng NganhHoc

Thông tin của bảng ngành học: MaNganh, TenNganh, MaKhoa

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	MaNganh	Text	Thuộc tính để phân biệt các ngành học (khóa chính). Mỗi ngành có 1 mã ngành để phân biệt
2	TenNganh	Text	Tên của Ngành học
3	MaKhoa	Text	Ngành thuộc một mã khoa nào đó

c. Mô tả bảng MonHoc

Thông tin của bảng MonHoc: MaMH, TenMH, LyThuyet, ThucHan

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaMH	Text	Mã môn học để phân biệt các môn trong danh sách các môn là khóa chính.
2	TenMH	Text	Tên môn học
3	LyThuyet	Float	Số tiết lý thuyết
4	ThucHan	Float	Số tiết thực hành

d. Mô tả bảng KhoaHoc

Thông tin của bảng KhoaHoc: MaKhoaHoc, TenKhoaHoc

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaKhoaHoc	Text	Là khóa chính để phân biệt với các khóa học , mỗi khóa học được cung cấp một mã riêng.
2	TenKhoaHoc	Text	Tên của khóa học

e. Mô tả bảng HeDaoTao

Thông tin của bảng HeDaoTao: MaHe, TenHe

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaHe	Text	Là khóa chính để phân biệt với các hệ học, mỗi hệ được cung cấp một mã riêng.
2	TenHe	Text	Tên của hệ

f. Mô tả bảng BangDiem

Thông tin của bảng BangDiem: MaSV, MaMH, LanThi, HocKy, Diem

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaSV	Text	Mã sinh viên để phân biệt các sinh viên
2	MaMH	Text	Mã môn học
3	LanThi	Text	Số lần thi của sinh viên
4	HocKy	Text	Học kỳ trong năm học
5	Diem	Float	Điểm của sinh viên

g. Mô tả bảng Khoa

Thông tin của bảng Khoa: MaKhoa, TenKhoa

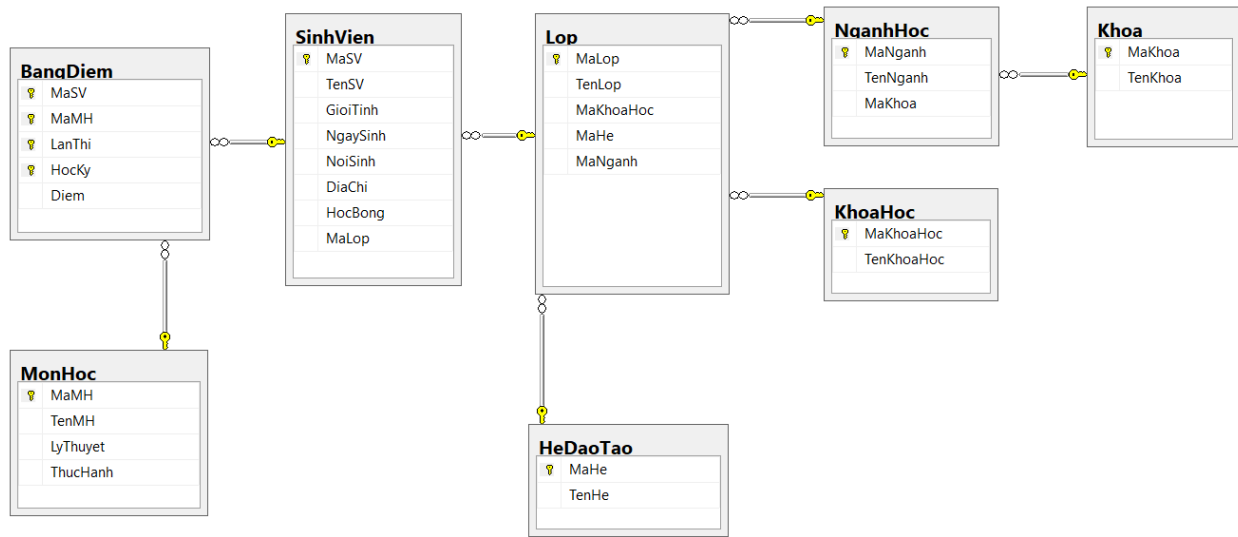
STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaKhoa	Text	Mã khoa để phân biệt các khoa khác
2	TenKhoa	Text	Tên khoa

h. Mô tả bảng SinhVien

Thông tin của bảng SinhVien: MaSV, TenSV, GioiTinh, NgaySinh, NoiSinh, DiaChi, MaLop, HocBong

STT	Thuộc tính	Kiểu dữ Liệu	Diễn giải
1	MaSV	Text	Mã sinh viên để phân biệt các sinh viên
2	TenSV	Text	Tên sinh viên
3	GioiTinh	Text	Giới tính
4	NgaySinh	Date time	Ngày sinh của sinh viên
5	NoiSinh	Text	Nơi sinh của sinh viên
6	DiaChi	Text	Địa chỉ sinh viên
7	MaLop	Text	Mỗi sinh viên sẽ có 1 mã lớp
8	HocBong	Int	Học bổng mà sinh viên nhận được

2.3 Biểu đồ quan hệ giữa các bảng



3. Danh sách các Stored Procedure, Function, Trigger

3.1 Stored Procedure

3.1.1 Thêm 1 bảng điểm

```
CREATE PROCEDURE [dbo].[Add]
@MaSV nchar(10),
@MaMH nchar(10),
@LanThi nchar(10),
@HocKy int,
@Diem float
AS
BEGIN
    INSERT INTO BangDiem(MaSV, MaMH, LanThi, HocKy, Diem)
    VALUES (@MaSV, @MaMH, @LanThi, @HocKy, @Diem)
END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "Add"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu stored
procedure
```

```

cmd.Parameters.AddWithValue("MaSV", cboMaSV.Text); //Lấy giá trị
của ComboBox cboMaSV truyền cho tham số @MaSV
cmd.Parameters.AddWithValue("MaMH", cboMaMH.Text); //Lấy giá trị
của ComboBox cboMaMH truyền cho tham số @MaMH
cmd.Parameters.AddWithValue("LanThi", cboLanThi.Text); //Lấy giá trị
của ComboBox cboLanThi truyền cho tham số @LanThi
cmd.Parameters.AddWithValue("Hocky", cboHocky.Text); //Lấy giá trị
của ComboBox cboHocky truyền cho tham số @Hocky
cmd.Parameters.AddWithValue("Diem", txtDiem.Text); //Lấy giá trị
của TextBox txtDiem truyền cho tham số @Diem
cmd.ExecuteNonQuery(); //Thực thi Stored Procedure

```

3.1.2 Thêm 1 môn học

```

CREATE PROCEDURE AddMH
    @MaMH nchar(10),
    @TenMH nvarchar(50),
    @LyThuyet float,
    @ThucHanh float
AS
    BEGIN
        INSERT INTO MonHoc
        VALUES (@MaMH, @TenMH, @LyThuyet, @ThucHanh)
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "AddMH"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu
Stored Procedure
cmd.Parameters.AddWithValue("MaMH", txtMaMH5.Text);
//Lấy giá trị của TextBox txtMaMH5 truyền cho tham số @MaMH
cmd.Parameters.AddWithValue("TenMH", txtTenMH5.Text);
//Lấy giá trị của TextBox txtTenMH5 truyền cho tham số @TenMH
cmd.Parameters.AddWithValue("LyThuyet", txtSoTietLT5.Text);
//Lấy giá trị của TextBox txtSoTietLT5 truyền cho tham số
@LyThuyet
cmd.Parameters.AddWithValue("ThucHanh", txtSoTietTH5.Text);
//Lấy giá trị của TextBox txtSoTietTH5 truyền cho tham số
@ThucHanh
cmd.ExecuteNonQuery(); //Thực thi Stored Procedure

```

3.1.3 Thêm 1 ngành học

```
CREATE PROCEDURE AddNganh
    @MaNganh nchar(10),
    @TenNganh nvarchar(50),
    @MaKhoa nchar(10)
AS
    BEGIN
        INSERT INTO NganhHoc
        VALUES (@MaNganh, @TenNganh, @MaKhoa)
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "AddNganh"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaNganh", txtMN4.Text); //Lấy giá trị
của TextBox txtMN4 truyền cho tham số @MaNganh
cmd.Parameters.AddWithValue("TenNganh", txtTN4.Text); //Lấy giá trị
của TextBox txtTN4 truyền cho tham số @TenNganh
cmd.Parameters.AddWithValue("MaKhoa", cboMaKhoa4.Text); //Lấy giá trị
của ComboBox cboMaKhoa4 truyền cho tham số @MaKhoa
cmd.ExecuteNonQuery(); //Thực thi Stored Procedure
```

3.1.4 Thêm 1 sinh viên

```
CREATE PROCEDURE ADDSV
    @MaSV nchar(10),
    @TenSV nvarchar(50),
    @GioiTinh nvarchar(50),
    @NgaySinh datetime,
    @NoiSinh nvarchar(50),
    @DiaChi nvarchar(50),
    @HocBong float,
    @MaLop nchar(10)
AS
    BEGIN
        INSERT INTO SinhVien(MaSV, TenSV, GioiTinh, NgaySinh, NoiSinh,
        DiaChi, HocBong, MaLop)
        VALUES (@MaSV, @TenSV, @GioiTinh, @NgaySinh, @NoiSinh, @DiaChi,
        @HocBong, @MaLop)
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "ADDSV";           //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaSV", txtMaSV3.Text);           //Lấy giá
trị của TextBox txtMaSV3 truyền cho tham số @MaSV
cmd.Parameters.AddWithValue("TenSV", txtTenSV3.Text);           //Lấy giá
trị của TextBox txtTenSV3 truyền cho tham số @TenSV
cmd.Parameters.AddWithValue("GioiTinh", cboGT3.Text);           //Lấy giá
trị của ComboBox cboGT3 truyền cho tham số @GioiTinh
cmd.Parameters.AddWithValue("NgaySinh", dtpNS3.Value);           //Lấy giá
trị của DateTimePicker dtpNS3 truyền cho tham số @NgaySinh
cmd.Parameters.AddWithValue("NoiSinh", txtNoiSinh3.Text);           //Lấy giá
trị của TextBox txtNoiSinh3 truyền cho tham số @NoiSinh
cmd.Parameters.AddWithValue("DiaChi", txtDiaChi3.Text);           //Lấy giá
trị của TextBox txtDiaChi3 truyền cho tham số @DiaChi
cmd.Parameters.AddWithValue("MaLop", cboMaLop3.Text);           //Lấy giá
trị của ComboBox cboMaLop3 truyền cho tham số @MaLop
cmd.Parameters.AddWithValue("HocBong", txtHB3.Text);           //Lấy giá
trị của TextBox txtHB3 truyền cho tham số @HocBong
cmd.ExecuteNonQuery();           //Thực thi Stored Procedure
```

3.1.5 Xóa 1 bảng điểm

```
CREATE PROCEDURE Delete
    @MaSV nchar(10),
    @MaMH nchar(10),
    @LanThi nchar(10),
    @HocKy int
AS
    BEGIN
        DELETE
        FROM BangDiem
        WHERE [MaSV]=@MaSV AND [MaMH]=@MaMH AND [LanThi] = @LanThi AND
[LanThi] = @LanThi
        [HocKy] = @HocKy
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
```

```

conn.Open();           //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "Delete";    //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored
Procedure
cmd.Parameters.AddWithValue("MaSV", cboMaSV.Text); //Lấy giá trị của
ComboBox cboMaSV truyền cho tham số @MaSV
cmd.Parameters.AddWithValue("MaMH", cboMaMH.Text); //Lấy giá trị của
ComboBox cboMaMH truyền cho tham số @MaMH
cmd.Parameters.AddWithValue("LanThi", cboLanThi.Text); //Lấy giá trị
của ComboBox cboLanThi truyền cho tham số @LanThi
cmd.Parameters.AddWithValue("HocKy", cboHocKy.Text); //Lấy giá trị
của ComboBox cboHocKy truyền cho tham số @HocKy
cmd.ExecuteNonQuery(); //Thực thi

```

3.1.6 Xóa 1 môn học

```

CREATE PROCEDURE DeleteMH
    @MaMH nchar(10)
AS
    BEGIN
        DELETE
        FROM BangDiem
        WHERE [MaMH] = @MaMH

        DELETE
        FROM MonHoc
        WHERE [MaMH] = @MaMH
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "DeleteMH"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaMH", txtMaMH5.Text); //Lấy giá trị của
TextBox txtMaMH5 truyền cho tham số @MaMH
cmd.ExecuteNonQuery(); //Thực thi

```

3.1.7 Xóa 1 ngành học

```

CREATE PROCEDURE DeleteNganh
    @MaNganh nchar(10)
AS
    BEGIN
        DELETE
        FROM Lop
        WHERE [MaNganh] = @MaNganh
        DELETE
        FROM NganhHoc
        WHERE [MaNganh] = @MaNganh
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "DeleteNganh";    //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaNganh", txtMN4.Text); //Lấy giá trị
của TextBox txtMN4 truyền cho tham số @MaNganh
cmd.ExecuteNonQuery(); //Thực thi

```

3.1.8 Xóa 1 sinh viên

```

CREATE PROCEDURE DeleteTTSV
    @MaSV nchar(10)
AS
    BEGIN
        DELETE
        FROM BangDiem
        WHERE [MaSV] = @MaSV
        DELETE
        FROM SinhVien
        WHERE [MaSV] = @MaSV
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "DeleteTTSV";    //Gọi tên Stored Procedure

```



```
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaSV", txtMaSV3.Text); //Lấy giá trị của
TextBox txtMaSV3 truyền cho tham số @MaSV
cmd.ExecuteNonQuery(); //Thực thi
```

3.1.9 Lấy thông tin sinh viên tương ứng mã lớp

```
CREATE PROCEDURE LoadData
    @MaLop nchar(10)
AS
BEGIN
    SELECT MaSV, TenSV, GioiTinh, NgaySinh, NoiSinh, DiaChi, HocBong
    FROM SinhVien
    WHERE [MaLop] = @MaLop
END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối đến SQL Server
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "LoadData"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaLop", cboMaLop.Text); //Lấy giá trị
của ComboBox cboMaLop truyền cho tham số @MaLop
daSV = new SqlDataAdapter(cmd); //Thực thi câu lệnh, đưa vào
DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
dgvSV.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên
DataGridView
```

3.1.10 Lấy thông tin chi tiết điểm từng môn

```
CREATE PROCEDURE LoadDataDiem
    @MaSV nchar(10)
AS
BEGIN
    SELECT a.MaMH, b.TenMH, a.LanThi, a.HocKy, a.Diem
    FROM BangDiem as a, MonHoc as b
    WHERE [MaSV] = @MaSV AND a.MaMH = b.MaMH
END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "LoadDataDiem";           //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaSV", cboMaSV.Text); //Lấy giá trị của
ComboBox cboMaSV truyền cho tham số @MaSV
daSV = new SqlDataAdapter(cmd); //Thực thi câu lệnh, đưa vào
DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
dgvBangDiem.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên
DataGridView dgvBangDiem
```

3.1.11 Lấy thông tin lớp học tương ứng mã lớp

```
CREATE PROCEDURE LoadTTLop
    @MaLop nchar(10)
AS
    BEGIN
        SELECT TenLop, MaKhoaHoc, MaHe, MaNganh
        FROM Lop
        WHERE [MaLop] = @MaLop
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "LoadTTLop";           //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaLop", cboMaLop.Text); //Lấy giá trị
của ComboBox cboMaLop truyền cho tham số @MaLop
daSV = new SqlDataAdapter(cmd); //Thực thi câu lệnh, đưa dữ liệu
vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
```

```

daSV.Fill(dtSV);    //Từ DataAdapter đổ vào DataTable

//Đưa các thông tin của lớp lên các TextBox
txtTenLop.Text = dtSV.Rows[0][0].ToString();
txtMaKhoa.Text = dtSV.Rows[0][1].ToString();
txtMaHe.Text = dtSV.Rows[0][2].ToString();
txtMaNganh.Text = dtSV.Rows[0][3].ToString();

```

Mỗi khi thay đổi giá trị ComboBox cboMaLop thì các TextBox cũng thay đổi tương ứng với thông tin của cboMaLop được chọn.

3.1.12 Lấy mã khoa

```

CREATE PROCEDURE SelectMaKhoa
AS
    BEGIN
        SELECT MaKhoa
        FROM Khoa
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

LoadComboBox("SelectMaKhoa", cboMaKhoa4, "MaKhoa", "MaKhoa"); //Load
MaKhoa lên cboMaKhoa4

```

Hàm LoadComboBox có nội dung như sau:

```

private void LoadComboBox(string sp, ComboBox cbo, string ma, string
ten)
{
    try
    {
        // Khởi động connection
        conn = new SqlConnection(strConnectionString);
        if (conn.State == ConnectionState.Open)
            conn.Close();
        conn.Open();    //Mở kết nối
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandText = sp; //sp là tên store procedure truyền vào
        cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored
        Procedure
        SqlDataAdapter da = new SqlDataAdapter(cmd);    //Thực thi
        Stored Procedure, đưa vào DataAdapter
    }
    catch { }
}

```

```

        DataTable dt = new DataTable();
        da.Fill(dt);           //Từ DataAdapter đổ vào DataTable
        cbo.DataSource = dt; //Load giá trị từ DataTable lên ComboBox
cbo
        cbo.DisplayMember = ten; //Trường hiển thị
        cbo.ValueMember = ma; //Trường giá trị
    }
    catch (SqlException se)
    {
        MessageBox.Show(se.ToString());
    }
}

```

3.1.13 Lấy mã lớp

```

CREATE PROCEDURE SelectMaLop
AS
    BEGIN
        SELECT MaLop
        FROM Lop
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

LoadComboBox("SelectMaLop", cboMaLop, "MaLop", "MaLop"); //Load
MaLop lên cboMaLop
LoadComboBox("SelectMaLop", cboMaLop3, "MaLop", "MaLop"); //Load
MaLop lên cboMaLop3

```

3.1.14 Lấy mã môn học

```

CREATE PROCEDURE SelectMaMH
AS
    BEGIN
        SELECT MaMH
        FROM MonHoc
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

LoadComboBox("SelectMaMH", cboMaMH, "MaMH", "MaMH"); //Load MaMH lên
cboMaMH

```

3.1.15 Lấy mã sinh viên

```
CREATE PROCEDURE SelectMaSV
AS
    BEGIN
        SELECT MaSV
        FROM SinhVien
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
LoadComboBox("SelectMaSV", cboMaSV, "MaSV", "MaSV"); //Load MaSV lên
cboMaSV
```

3.1.16 Lấy tên khoa tương ứng mã khoa

```
CREATE PROCEDURE SelectTenKhoa
    @MaKhoa nchar(10)
AS
    BEGIN
        SELECT TenKhoa
        FROM Khoa
        WHERE [MaKhoa] = @MaKhoa
    END
RETURN 0
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "SelectTenKhoa"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaKhoa", cboMaKhoa4.Text); //Lấy giá trị
của ComboBox cboMaKhoa4 truyền cho tham số @MaKhoa
daSV = new SqlDataAdapter(cmd); //Thực thi câu lệnh, đưa dữ liệu vào
DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
txtTkhoa4.Text = dtSV.Rows[0][0].ToString(); //Hiện tên khoa
```

Giá trị txtTkhoa4 sẽ thay đổi tương ứng với tên khoa của mã khoa được chọn.

3.1.17 Lấy tên môn học tương ứng mã môn học

```
CREATE PROCEDURE SelectTenMH
    @MaMH nchar(10)
AS
    BEGIN
        SELECT TenMH FROM MonHoc
        WHERE [MaMH] = @MaMH
    END
```

Stored Procedure được sử dụng trong C# như sau:

```
// Khởi động connection
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "SelectTenMH"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaMH", cboMaMH.Text); //Lấy giá trị của
//ComboBox cboMaMH truyền cho tham số @MaMH
daSV = new SqlDataAdapter(cmd); //Thực thi câu lệnh, đưa dữ liệu vào
//DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
txtTenMH.Text = dtSV.Rows[0][0].ToString(); //Hiện tên môn học
```

Giá trị txtTenMH sẽ thay đổi tương ứng với tên khoa của mã khoa được chọn.

3.1.18 Sửa điểm của bảng điểm

```
CREATE PROCEDURE Update
    @MaSV nchar(10),
    @MaMH nchar(10),
    @LanThi nchar(10),
    @HocKy int,
    @Diem float
AS
    BEGIN
        UPDATE BangDiem
        SET [HocKy] = @HocKy,
            [Diem] = @Diem
        WHERE [MaSV] = @MaSV AND [MaMH] = @MaMH AND [LanThi] = @LanThi
```

END

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "Update"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaSV", cboMaSV.Text); //Lấy giá trị
của ComboBox cboMaSV truyền cho tham số @MaSV
cmd.Parameters.AddWithValue("MaMH", cboMaMH.Text); //Lấy giá trị
của ComboBox cboMaMH truyền cho tham số @MaMH
cmd.Parameters.AddWithValue("LanThi", cboLanThi.Text); //Lấy giá trị
của ComboBox cboLanThi truyền cho tham số @LanThi
cmd.Parameters.AddWithValue("HocKy", cboHocKy.Text); //Lấy giá trị
của ComboBox cboHocKy truyền cho tham số @HocKy
cmd.Parameters.AddWithValue("Diem", txtDiem.Text); //Lấy giá trị
của TextBox txtDiem truyền cho tham số @Diem
cmd.ExecuteNonQuery(); //Thực thi Stored Procedure
```

3.1.19 Sửa thông tin môn học

```
CREATE PROCEDURE UpdateMH
    @MaMH nchar(10),
    @TenMH nvarchar(50),
    @LyThuyet float,
    @ThucHanh float
AS
BEGIN
    UPDATE MonHoc
    SET [TenMH] = @TenMH,
        [LyThuyet] = @LyThuyet,
        [ThucHanh] = @ThucHanh
    WHERE [MaMH] = @MaMH
END
```

Stored Procedure được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
```

```

cmd.CommandText = "UpdateMH"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaMH", txtMaMH5.Text); //Lấy giá
trị của TextBox txtMaMH5 truyền cho tham số @MaMH
cmd.Parameters.AddWithValue("TenMH", txtTenMH5.Text); //Lấy giá
trị của TextBox txtTenMH5 truyền cho tham số @TenMH
cmd.Parameters.AddWithValue("LyThuyet", txtSoTietLT5.Text); //Lấy giá
trị của TextBox txtSoTietLT5 truyền cho tham số @LyThuyet
cmd.Parameters.AddWithValue("ThucHanh", txtSoTietTH5.Text); //Lấy giá
trị của TextBox txtSoTietTH5 truyền cho tham số @ThucHanh
cmd.ExecuteNonQuery(); //Thực thi

```

3.1.20 Sửa thông tin ngành

```

CREATE PROCEDURE UpdateNganh
    @MaNganh nchar(10),
    @TenNganh nvarchar(50),
    @MaKhoa nchar(10)
AS
BEGIN
    UPDATE NganhHoc
    SET [TenNganh] = @TenNganh,
        [MaKhoa] = @MaKhoa
    WHERE [MaNganh] = @MaNganh
END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "UpdateNganh"; //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaNganh", txtMN4.Text); //Lấy giá trị
của TextBox txtMN4 truyền cho tham số @MaNganh
cmd.Parameters.AddWithValue("TenNganh", txtTN4.Text); //Lấy giá trị
của TextBox txtTN4 truyền cho tham số @TenNganh
cmd.Parameters.AddWithValue("MaKhoa", cboMaKhoa4.Text); //Lấy giá trị
của ComboBox cboMaKhoa4 truyền cho tham số @MaKhoa
cmd.ExecuteNonQuery(); //Thực thi

```

3.1.21 Sửa thông tin sinh viên

```

CREATE PROCEDURE UpdateSV
    @MaSV nchar(10),

```



```

        @TenSV nvarchar(50),
        @GioiTinh nvarchar(50),
        @NgaySinh datetime,
        @NoiSinh nvarchar(50),
        @DiaChi nvarchar(50),
        @HocBong float
AS
    BEGIN
        UPDATE SinhVien
        SET [TenSV] = @TenSV,
            [GioiTinh] = @GioiTinh,
            [NgaySinh] = @NgaySinh,
            [NoiSinh] = @NoiSinh,
            [DiaChi] = @DiaChi,
            [HocBong] = @HocBong
        WHERE [MaSV] = @MaSV
    END

```

Stored Procedure được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "UpdateSV";           //Gọi tên Stored Procedure
cmd.CommandType = CommandType.StoredProcedure; //Kiểu Stored Procedure
cmd.Parameters.AddWithValue("MaSV", txtMaSV.Text);           //Lấy giá trị
của TextBox txtMaSV truyền cho tham số @MaSV
cmd.Parameters.AddWithValue("TenSV", txtTenSV.Text);           //Lấy giá trị
của TextBox txtTenSV truyền cho tham số @TenSV
cmd.Parameters.AddWithValue("GioiTinh", cboGT.Text);           //Lấy giá trị
của ComboBox cboGT truyền cho tham số @GioiTinh
cmd.Parameters.AddWithValue("NS", dtpNgaySinh.Value);           //Lấy giá trị
của DateTimePicker dtpNgaySinh truyền cho tham số @NS
cmd.Parameters.AddWithValue("NoiSinh", txtNoiSinh.Text); //Lấy giá trị
của TextBox txtNoiSinh truyền cho tham số @NoiSinh
cmd.Parameters.AddWithValue("DiaChi", txtDiaChi.Text);           //Lấy giá trị
của TextBox txtDiaChi truyền cho tham số @DiaChi
cmd.Parameters.AddWithValue("HocBong", txtHB.Text);           //Lấy giá trị
của TextBox txtHB truyền cho tham số @HocBong
cmd.ExecuteNonQuery();           //Thực thi

```

3.2 Function

3.2.1 Lấy thông tin tất cả sinh viên

```

CREATE FUNCTION fLoadDSSV()
RETURNS TABLE AS RETURN

```

```
(
    SELECT *
    FROM SinhVien
)
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fLoadDSSV()", conn); //Thực
thi Function fLoadDSSV và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
dgvDSSV.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên DataGridView
dgvDSSV
```

3.2.2 Lấy thông tin tất cả môn học

```
CREATE FUNCTION fLoadMH()
RETURNS TABLE AS RETURN
(
    SELECT *
    FROM MonHoc
)
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fLoadMH()", conn); //Thực thi
Function fLoadMH và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
dgvMH.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên DataGridView
dgvMH
```

3.2.3 Lấy thông tin sinh viên tương ứng mã sinh viên

```
CREATE FUNCTION fLoadTTTSV
(
    @MaSV nchar(10)
)
RETURNS TABLE AS RETURN
(
    SELECT TenSV, NgaySinh, NoiSinh
    FROM SinhVien
    WHERE [MaSV] = @MaSV
)
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fLoadTTTSV('" + cboMaSV.Text +
    "'")", conn); //Thực thi Function fLoadTTTSV với tham số được truyền vào
                //là cboMaSV và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV);       //Từ DataAdapter đổ vào DataTable
//Đưa thông tin sinh viên tương ứng cboMaSV lên các TextBox
txtHoTen.Text = dtSV.Rows[0][0].ToString();
txtNgaySinh.Text = dtSV.Rows[0][1].ToString().Replace("12:00:00 AM",
    "");
txtNS.Text = dtSV.Rows[0][2].ToString();
```

Mỗi khi thay đổi giá trị của cboMaSV thì thông tin sẽ thay đổi tương ứng với Mã SV

3.2.4 Tìm kiếm trong bảng sinh viên

```
CREATE FUNCTION fSearch
(
    @string nvarchar(50)
)
RETURNS TABLE AS RETURN
(
    SELECT *
    FROM SinhVien
    WHERE MaSV LIKE '%' + @string + '%' OR TenSV LIKE N'%' + @string + '%' OR
    GioiTinh LIKE N'%' + @string + '%' OR NgaySinh LIKE '%' + @string + '%' OR
    NoiSinh LIKE 'N%' + @string + '%' OR DiaChi LIKE N'%' + @string + '%' OR HocBong
    LIKE '%' + @string + '%' OR MaLop LIKE '%' + @string + '%'
```

)

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fSearch(N'" + txtSearch.Text
+ "')", conn);         //Thực thi Function fSearch với tham số được truyền
                        //vào là txtSearch và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV);       //Từ DataAdapter đổ vào DataTable
dgvDSSV.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên
DataGridView dgvDSSV
```

Nhập bất kỳ từ khóa cần tìm vào TextBox txtSearch thì thông tin sinh viên có liên quan tới từ khóa sẽ hiện lên DataGridView.

3.2.5 Tìm kiếm trong bảng môn học

```
CREATE FUNCTION fSearchMH
(
    @string nvarchar(50)
)
RETURNS TABLE AS RETURN
(
    SELECT *
    FROM MonHoc
    WHERE MaMH LIKE '%' + @string + '%' OR TenMH LIKE N'%' + @string + '%' OR
    LyThuyet LIKE '%' + @string + '%' OR ThucHanh LIKE '%' + @string + '%'
)
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fSearchMH(N'" +
txtSearch5.Text + "')", conn); //Thực thi Function fSearchMH với tham
số được truyền vào là txtSearch5 và đưa vào DataAdapter
dtSV = new DataTable();
```

```
dtSV.Clear();
daSV.Fill(dtSV);    //Từ DataAdapter đổ vào DataTable
dgvMH.DataSource = dtSV;    //Đưa dữ liệu từ DataTable lên
DataGridView dgvMH
```

Nhập bất kì từ khóa cần tìm vào TextBox txtSearch5 thì thông tin môn học có liên quan tới từ khóa sẽ hiện lên DataGridView.

3.2.6 Tìm kiếm trong bảng ngành học

```
CREATE FUNCTION fSearchNganh
(
    @string nvarchar(50)
)
RETURNS TABLE AS RETURN
(
    SELECT MaNganh, TenNganh, a.MaKhoa, TenKhoa
    FROM NganhHoc as a, Khoa as b
    WHERE a.MaKhoa = b.MaKhoa AND (MaNganh LIKE '%' + @string + '%' OR
    TenNganh LIKE '%' + @string + '%' OR a.MaKhoa LIKE '%' + @string + '%' OR
    TenKhoa LIKE '%' + @string + '%')
)
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();    //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fSearchNganh(N'" +
txtSearch4.Text + "')", conn); //Thực thi Function fSearchNganh với
tham số được truyền vào là txtSearch4 và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV);    //Từ DataAdapter đổ vào DataTable
dgvNganh4.DataSource = dtSV;    //Đưa dữ liệu từ DataTable lên
DataGridView dgvNganh4
```

Nhập bất kì từ khóa cần tìm vào TextBox txtSearch4 thì thông tin ngành học có liên quan tới từ khóa sẽ hiện lên DataGridView.

3.2.7 Lấy thông tin tất cả ngành học

```

CREATE FUNCTION fSelectNganh()
RETURNS TABLE
AS
RETURN
(
    SELECT MaNganh, TenNganh, a.MaKhoa, TenKhoa
    FROM NganhHoc as a, Khoa as b
    WHERE a.MaKhoa = b.MaKhoa
)

```

Function được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();
daSV = new SqlDataAdapter("SELECT * FROM fSelectNganh()", conn);
//Thực thi Function fSelectNganh và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV);       //Từ DataAdapter đổ vào DataTable
dgvNganh4.DataSource = dtSV; //Đưa dữ liệu từ DataTable lên
DataGridView dgvNganh4

```

3.2.8 Tính điểm trung bình tương ứng mã sinh viên

```

CREATE FUNCTION fDiemTB
(
    @MaSV nchar(10)
)
RETURNS FLOAT AS
BEGIN
    DECLARE @DTB FLOAT
    SELECT @DTB = AVG(DiemMax.Diem)
    FROM (SELECT MaMH, MAX(Diem) AS Diem
          FROM BangDiem
          WHERE MaSV = @MaSV AND Diem >= 5.0
          GROUP BY MaMH) AS DiemMax
    RETURN @DTB
END

```

Function được sử dụng trong C# như sau:

```

conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open();           //Mở kết nối
SqlCommand cmd = new SqlCommand();

```

```
daSV = new SqlDataAdapter("SELECT dbo.fDiemTB('" + cboMaSV.Text +
    "')", conn); //Thực thi Function fDiemTB với tham số được truyền vào
    là cboMaSV và đưa vào DataAdapter
dtSV = new DataTable();
dtSV.Clear();
daSV.Fill(dtSV); //Từ DataAdapter đổ vào DataTable
txtDTB.Text = dtSV.Rows[0][0].ToString(); //Lấy điểm trung bình vừa
    tính đưa lên txtDTB
```

Mỗi khi thay đổi giá trị của ComboBox cboMaSV thì txtDTB cũng thay đổi tương ứng với điểm trung bình của Mã SV được chọn.

3.2.9 Đếm số lượng môn học

```
CREATE FUNCTION fSoLuongMH()
RETURNS INT AS
BEGIN
    DECLARE @SL INT
    SELECT @SL = COUNT(MaMH)
    FROM MonHoc
    RETURN @SL
END
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand("SELECT dbo.fSoLuongMH()", conn);
//Khai báo câu lệnh cần để thực thi Function fSoLuongMH
int i;
i = Convert.ToInt32(cmd.ExecuteScalar()); //Thực thi Function và gán
    vào biến i
txtSL5.Text = i.ToString(); //Hiện giá trị lên txtSL5
```

3.2.10 Đếm số lượng ngành học

```
CREATE FUNCTION fSoLuongNganh()
RETURNS INT AS
BEGIN
    DECLARE @SL INT
    SELECT @SL = COUNT(MaNganh)
    FROM NganhHoc
    RETURN @SL
END
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand("SELECT dbo.fSoLuongNganh()", conn);
//Khai báo câu lệnh cần để thực thi Function fSoLuongNganh
int i;
i = Convert.ToInt32(cmd.ExecuteScalar()); //Thực thi Function và gán
vào biến i
txtSL4.Text = i.ToString(); //Hiện giá trị lên txtSL4
```

3.2.11 Đếm số lượng sinh viên

```
CREATE FUNCTION fSoLuongSV
(
)
RETURNS INT AS
BEGIN
    DECLARE @SL INT
    SELECT @SL = COUNT(MaSV)
    FROM SinhVien
    RETURN @SL
END
```

Function được sử dụng trong C# như sau:

```
conn = new SqlConnection(strConnectionString);
if (conn.State == ConnectionState.Open)
    conn.Close();
conn.Open(); //Mở kết nối
SqlCommand cmd = new SqlCommand("SELECT dbo.fSoLuongSV()", conn);
//Khai báo câu lệnh cần để thực thi Function fSoLuongSV
int i;
i = Convert.ToInt32(cmd.ExecuteScalar()); //Thực thi Function và gán
vào biến i
txtSL.Text = i.ToString(); //Hiện giá trị lên txtSL
```

3.3 Trigger

3.3.1 Kiểm tra khi nhập điểm

```
CREATE TRIGGER tg_ktDiem on BangDiem
FOR INSERT, UPDATE
AS
```



```

BEGIN
    DECLARE @Diem float
    SELECT @Diem = ne.Diem
    FROM INSERTED as ne
    IF (@Diem > 10 OR @Diem < 0)
        BEGIN
            PRINT(N'Diểm phải từ 0-10')
            ROLLBACK
        END
END

```

3.3.2 Thông báo khi cập nhật điểm

```

CREATE TRIGGER tg_NotiUpdateBangDiem on BangDiem
FOR UPDATE
AS
BEGIN
    DECLARE @DiemCu float
    DECLARE @DiemMoi float

    SELECT @DiemCu = ol.Diem
    FROM Deleted as ol
    SELECT @DiemMoi = ne.Diem
    FROM Inserted as ne

    IF (@DiemCu != @DiemMoi)
        PRINT N'Đã chỉnh sửa điểm'
END

```

3.3.3 Kiểm tra khi nhập thông tin môn học

```

CREATE TRIGGER tg_ktNhapTTMonHoc on MonHoc
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @MaMH nchar(10), @TenMH nvarchar(50), @LyThuyet int, @ThucHanh int
    SELECT @MaMH = MaMH, @TenMH = TenMH, @LyThuyet = LyThuyet, @ThucHanh =
ThucHanh
    FROM INSERTED
    IF (RTRIM(@MaMH) = '')
        BEGIN
            PRINT(N'Mã MH không được rỗng')
            ROLLBACK
        END
    ELSE IF (RTRIM(@TenMH) = '')
        BEGIN
            PRINT(N'Tên MH không được rỗng')
            ROLLBACK
        END
END

```

3.3.4 Thông báo khi xóa môn học

```
CREATE TRIGGER tg_NotiDeleteMH on MonHoc
FOR DELETE
AS
BEGIN
    DECLARE @MaMH nchar(10)

    SELECT @MaMH = ol.MaMH
    FROM Deleted as ol
    IF (@MaMH NOT IN (SELECT MaMH FROM MonHoc))
        PRINT N'Đã xóa môn học ' + RTRIM(@MaMH)
END
```

3.3.5 Thông báo khi cập nhật thông tin môn học

```
CREATE TRIGGER tg_NotiUpdateMH on MonHoc
FOR UPDATE
AS
BEGIN
    DECLARE @TenMHCu nvarchar(50), @SoTietLTCu float, @SoTietTHCu float
    DECLARE @TenMHMoi nvarchar(50), @SoTietLTMoi float, @SoTietTHMoi float

    SELECT @TenMHCu = ol.TenMH, @SoTietLTCu = ol.LyThuyet, @SoTietTHCu =
ol.ThucHanh
    FROM Deleted as ol
    SELECT @TenMHMoi = ne.TenMH, @SoTietLTMoi = ne.LyThuyet, @SoTietTHMoi =
ne.ThucHanh
    FROM Inserted as ne

    IF (@TenMHCu != @TenMHMoi OR @SoTietLTCu != @SoTietLTMoi OR @SoTietTHCu !=
@SoTietTHMoi)
        PRINT N'Đã chỉnh sửa thông tin ngành'
END
```

3.3.6 Kiểm tra nhập thông tin ngành học khi muốn thêm hoặc cập nhật

```
CREATE TRIGGER tg_ktNhapTTNganhHoc on NganhHoc
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @MaNganh nchar(10), @TenNganh nvarchar(50), @MaKhoa nchar(10)
    SELECT @MaNganh = ne.MaNganh, @TenNganh = ne.TenNganh
    FROM INSERTED AS ne
    IF (RTRIM(@MaNganh) = '')
        BEGIN
            PRINT(N'Hãy nhập mã ngành')
            ROLLBACK
        END
    ELSE IF (RTRIM(@TenNganh) = '')
        BEGIN
            PRINT(N'Hãy nhập tên ngành')
            ROLLBACK
        END
END
```

```
END  
END
```

3.3.7 Thông báo khi xóa ngành học

```
CREATE TRIGGER tg_NotiDeleteNganh on NganhHoc  
FOR DELETE  
AS  
BEGIN  
    DECLARE @MaNganh nchar(10)  
  
    SELECT @MaNganh = ol.MaNganh  
    FROM Deleted as ol  
    IF (@MaNganh NOT IN (SELECT MaSV FROM SinhVien))  
        PRINT N'Đã xóa mã ngành ' + RTRIM(@MaNganh)  
END
```

3.3.8 Thông báo khi cập nhật thông tin ngành học

```
CREATE TRIGGER tg_NotiUpdateNganh on NganhHoc  
FOR UPDATE  
AS  
BEGIN  
    DECLARE @TenNganhCu nvarchar(50), @MaKhoaCu nchar(10)  
    DECLARE @TenNganhMoi nvarchar(50), @MaKhoaMoi nchar(10)  
  
    SELECT @TenNganhCu = ol.TenNganh, @MaKhoaCu = ol.MaKhoa  
    FROM Deleted as ol  
    SELECT @TenNganhMoi = ne.TenNganh, @MaKhoaMoi = ne.MaKhoa  
    FROM Inserted as ne  
  
    IF (@TenNganhCu != @TenNganhMoi OR @MaKhoaCu != @MaKhoaMoi)  
        PRINT N'Đã chỉnh sửa thông tin ngành'  
END
```

3.3.9 Kiểm tra nhập thông tin sinh viên khi muốn thêm hoặc cập nhật

```
CREATE TRIGGER tg_KTNhapTTSinhVien on SinhVien  
FOR INSERT, UPDATE  
AS  
BEGIN  
    DECLARE @MaSV nchar(10), @TenSV nvarchar(50), @MaLop nchar(10)  
    SELECT @MaSV = ne.MaSV, @TenSV = ne.TenSV, @MaLop = MaLop  
    FROM Inserted as ne  
    IF (RTRIM(@MaSV) = '')  
        BEGIN  
            PRINT(N'Hãy nhập mã SV')  
            ROLLBACK  
        END  
    ELSE IF (RTRIM(@TenSV) = '')  
        BEGIN
```

```

                                PRINT(N'Hãy nhập tên SV')
                                ROLLBACK
                                END
END

```

3.3.10 Thông báo khi xóa sinh viên

```

CREATE TRIGGER tg_NotiDeleteSV on SinhVien
FOR DELETE
AS
BEGIN
    DECLARE @MaSV nchar(10)

    SELECT @MaSV = ol.MaSV
    FROM Deleted as ol
    IF (@MaSV NOT IN (SELECT MaSV FROM SinhVien))
        PRINT N'Đã xóa sinh viên ' + RTRIM(@MaSV)
END

```

3.3.11 Thông báo khi cập nhật thông tin sinh viên

```

CREATE TRIGGER tg_NotiUpdateSV on SinhVien
FOR UPDATE
AS
BEGIN
    DECLARE @TenSVMoi nvarchar(50), @GTMoi nvarchar(50), @HocBongMoi float,
    @MaLopMoi nchar(10)
    DECLARE @TenSVCu nvarchar(50), @GTCu nvarchar(50), @HocBongCu float,
    @MaLopCu nchar(10)

    SELECT @TenSVCu = ol.TenSV, @GTCu = ol.GioiTinh, @HocBongCu = ol.HocBong,
    @MaLopCu = ol.MaLop
    FROM Deleted as ol
    SELECT @TenSVMoi = ne.TenSV, @GTMoi = ne.GioiTinh, @HocBongMoi =
    ne.HocBong, @MaLopMoi = ne.MaLop
    FROM Inserted as ne

    IF (@TenSVCu != @TenSVMoi OR @GTCu != @GTMoi OR @HocBongCu != @HocBongMoi
    OR @MaLopCu != @MaLopMoi)
        PRINT N'Đã chỉnh sửa'
END

```

3.3.12 Kiểm tra giới tính khi thêm hoặc cập nhật

```

CREATE TRIGGER tg_ktGioiTinh ON SinhVien
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE @MaSV nchar(10), @GioiTinh nvarchar(50)
    SELECT @MaSV = ne.MaSV, @GioiTinh = ne.GioiTinh
    FROM INSERTED ne
    IF (RTRIM(@GioiTinh) = '')

```

```

BEGIN
    UPDATE SinhVien
    SET GioiTinh = N'Chưa xác định'
    WHERE MaSV = @MaSV
END

```

4. Phân quyền

4.1 Tạo user và role

```

CREATE LOGIN QuanTri WITH PASSWORD = '123'
CREATE USER QuanTri FOR LOGIN QuanTri

CREATE LOGIN GiangVien WITH PASSWORD = '123'
CREATE USER GiangVien FOR LOGIN GiangVien

EXEC sp_addrole 'RoleQuanTri'
EXEC sp_addrole 'RoleGiangVien'

EXEC sp_addrolemember 'RoleQuanTri', 'QuanTri'
EXEC sp_addrolemember 'RoleGiangVien', 'GiangVien'

```

4.2 Phân quyền quản trị

```

GRANT EXECUTE ON dbo.[Add] TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.AddMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.AddNganh TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.ADDSV TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.[Delete] TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.DeleteMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.DeleteNganh TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.DeleteTTSV TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadData TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadDataDiem TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadTTLop TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadTTSV TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaKhoa TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMalop TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaSV TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectTenKhoa TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectTenMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.[Update] TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.UpdateMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.UpdateNganh TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.UpdateSV TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.UpdateTTSV TO RoleQuanTri WITH GRANT OPTION

GRANT SELECT ON dbo.fLoadDSSV TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT ON dbo.fLoadMH TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT ON dbo.fLoadTTSV TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT ON dbo.fSearch TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT ON dbo.fSearchMH TO RoleQuanTri WITH GRANT OPTION

```

```

GRANT SELECT ON dbo.fSearchNganh TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT ON dbo.fSelectNganh TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.fDiemTB TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongMH TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongNganh TO RoleQuanTri WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongSV TO RoleQuanTri WITH GRANT OPTION

GRANT SELECT, INSERT, UPDATE, DELETE ON BangDiem TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON HeDaoTao TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON Khoa TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON KhoaHoc TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON Lop TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON MonHoc TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON NganhHoc TO RoleQuanTri WITH GRANT OPTION
GRANT SELECT, INSERT, UPDATE, DELETE ON SinhVien TO RoleQuanTri WITH GRANT OPTION

```

4.3 Phân quyền giảng viên

```

GRANT EXECUTE ON dbo.[Add] TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.[Delete] TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.[Update] TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadData TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadDataDiem TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadTTLop TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.LoadTTSV TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaKhoa TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMalop TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaMH TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectMaSV TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectTenKhoa TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.SelectTenMH TO RoleGiangVien WITH GRANT OPTION

GRANT SELECT ON dbo.fLoadDSSV TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fLoadMH TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fLoadTTSV TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fSearch TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fSearchMH TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fSearchNganh TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON dbo.fSelectNganh TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.fDiemTB TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongMH TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongNganh TO RoleGiangVien WITH GRANT OPTION
GRANT EXECUTE ON dbo.fSoLuongSV TO RoleGiangVien WITH GRANT OPTION

GRANT SELECT, INSERT, UPDATE, DELETE ON BangDiem TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON HeDaoTao TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON Khoa TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON KhoaHoc TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON Lop TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON MonHoc TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON NganhHoc TO RoleGiangVien WITH GRANT OPTION
GRANT SELECT ON SinhVien TO RoleGiangVien WITH GRANT OPTION

```