

Tài liệu hướng dẫn thực hành

Tài liệu này sẽ hướng dẫn cách làm việc với các đối tượng Connection, Command và DataReader thông qua việc xây dựng các ứng dụng trong môi trường Visual Studio. Bao gồm 2 bài:

- Bài thực hành số 1: Sử dụng đối tượng Connection
- Bài thực hành số 2: Sử dụng kết hợp các đối tượng Connection, Command và DataReader

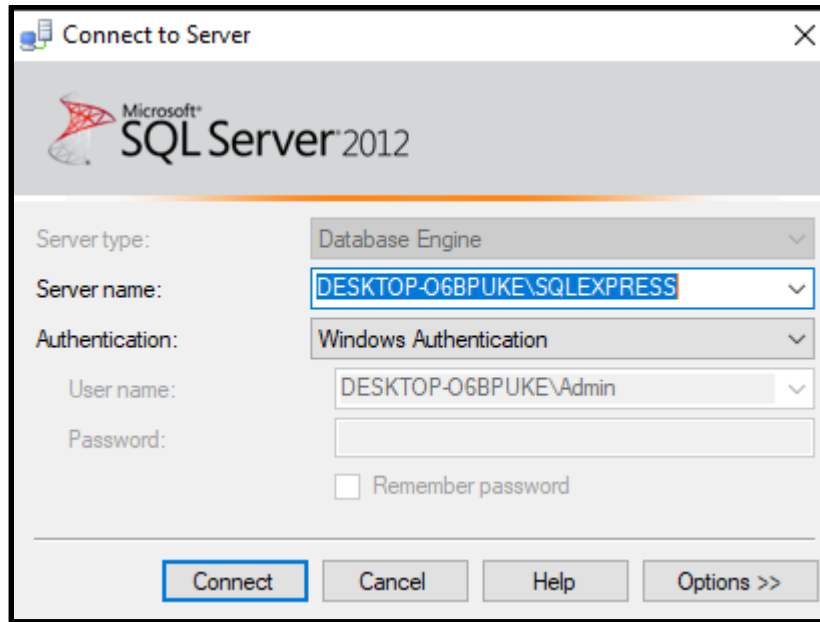
Một vài lưu ý trước khi thực hành:

- Tài liệu sử dụng Visual Studio 2019 Community. Các phiên bản Visual Studio trước đó các bước thực hiện tương tự.
- Tài liệu sử dụng SQL Server 2012 (Express).
- Có thể xem lại kiến thức cơ bản về WPF tại <https://ngocminhtran.com/lam-quen-ung-dung-wpf/> (Mặc dù dùng VS 2012 nhưng các thành phần cơ bản trong WPF vẫn tương tự).

Bài thực hành số 1

Thực hành sử dụng đối tượng Connection

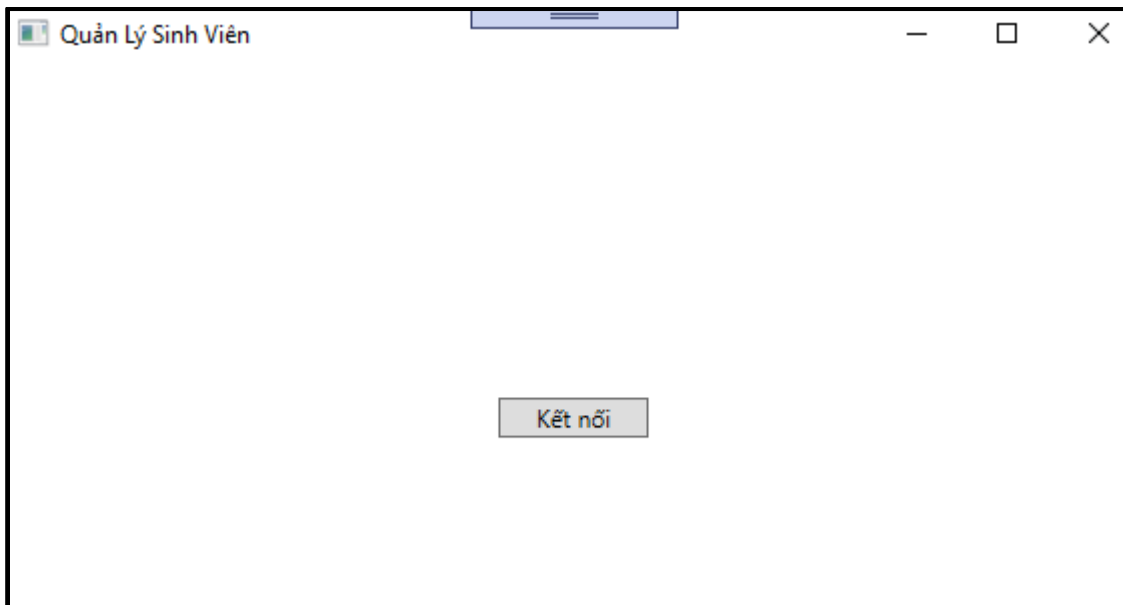
Bước 1: Mở MS SQL Server Management Studio và tạo cơ sở dữ liệu tên QuanLySinhVien. Lưu ý, trong tài liệu này sử dụng chế độ đăng nhập Windows Authentication và chúng ta cần đặc biệt chú ý đến Server name chúng ta dùng để đăng nhập như hình dưới:



Server name là tên máy cục bộ nên có thể khác nhau.

Bước 2: Mở Visual Studio 2019 Community và tạo dự án WPF app (.NET Framework) mới tên QuanLySinhVien.

Bước 3: Thay đổi tiêu đề cửa sổ thành Quản Lý Sinh Viên. Kế tiếp, kéo và thả một Button từ thanh Toolbox đến cửa sổ. Thay đổi thuộc tính Name của Button thành btnKetnoi và thuộc tính Content thành Kết nối. Giao diện:



Bước 4: Tạo hàm xử lý sự kiện Click cho Button Kết nối tên btnKetnoi_Click. Để sử dụng lớp SqlConnection, chúng ta cần thêm namespace sau:

```
using System.Data.SqlClient;
```

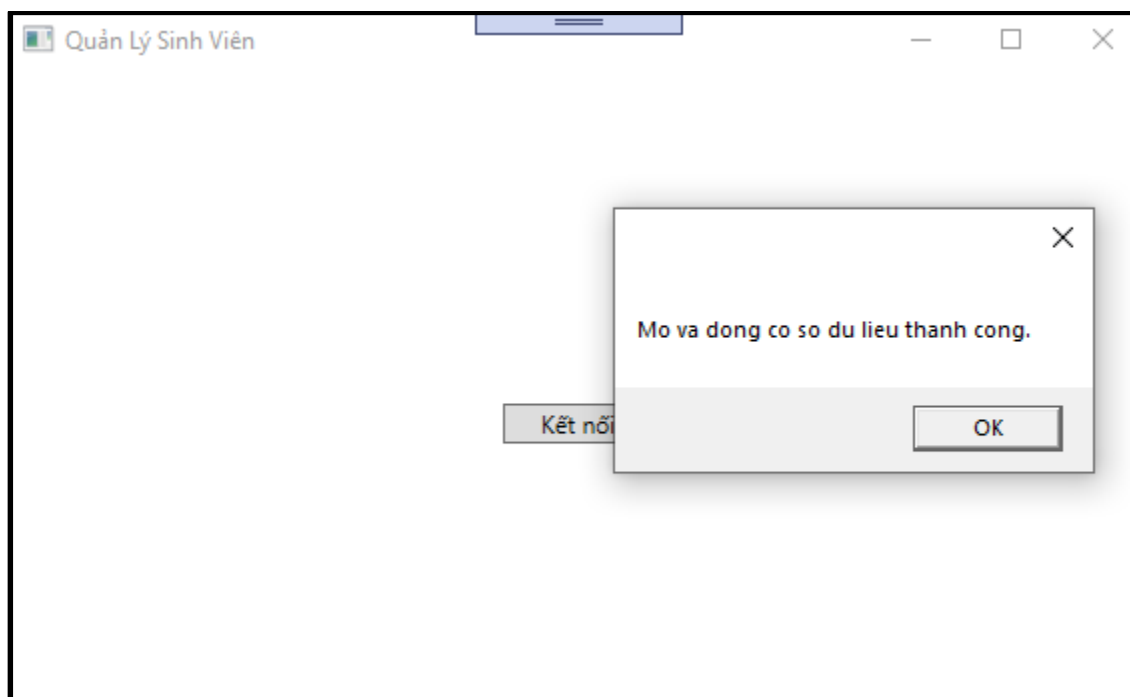
Bước 5: Thêm đoạn mã sau đến hàm btnKetnoi_Click(), chú ý tham số của phương thức khởi tạo SqlConnection:

```
private void btnKetnoi_Click(object sender, RoutedEventArgs e)
{
    try
    {
        using (SqlConnection connection =
            new SqlConnection(@"Server=DESKTOP06BPUKE\SQLEXPRESS;
                               Database=QuanLySinhVien;
                               Integrated Security=SSPI"))
        {
            connection.Open();
        }
        MessageBox.Show("Mở và đóng cơ sở dữ liệu thành công.");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Lỗi khi mở kết nối: " + ex.Message);
    }
}
```

Một số lưu ý:

- Chúng ta dùng lệnh *using* để gọi các phương thức Close() hay Dispose() để đóng kết nối và giải phóng tài nguyên.
- Dùng *try/catch* để bắt các ngoại lệ có thể phát sinh, ví dụ sai tên cơ sở dữ liệu.
- Chuỗi kết nối là tham số trong phương thức khởi tạo SqlConnection, bắt đầu bằng biểu tượng @ cho phép trình biên dịch bỏ qua các ký tự đặc biệt trong chuỗi kết nối.
- Chuỗi kết nối dùng từ khóa Server thay vì Data Source (tương đương).

Bước 6: Lưu tất cả và chạy ứng dụng. Kết quả khi nhấn nút Kết nối:




Đóng ứng dụng.

Kết thúc bài thực hành số 1.

Bài thực hành số 2

Thực hành sử dụng đối tượng Command và DataReader

Bước 1: Mở SQL Server Management Studio, chọn cơ sở dữ liệu QuanLySinhVien và thêm một bảng tên SinhVien gồm các cột MaSV, TenSV, Email, MaKH:

	Column Name	Data Type	Allow Nulls
	MaSV	nchar(10)	<input type="checkbox"/>
	TenSV	varchar(20)	<input checked="" type="checkbox"/>
	Email	nvarchar(50)	<input checked="" type="checkbox"/>
	MaKH	nchar(10)	<input type="checkbox"/>

Lược đồ bảng SinhVien

Với một vài mẫu dữ liệu như sau:

MaSV	TenSV	Email	MaKH
SV01	Hoang	hoang@gmail....	KH01
SV02	Binh	binh@gmail.com	KH01
SV03	Dong	dong@gmail.c...	KH02
SV04	Minh	minh@gmail.c...	KH03

Dữ liệu bảng SinhVien

Bước 2: Mở dự án QuanLySinhVien và tạo lớp mới tên SinhVien bằng cách thêm tập tin SinhVien.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace QuanLySinhVien
{
    class SinhVien
    {
    }
}
```

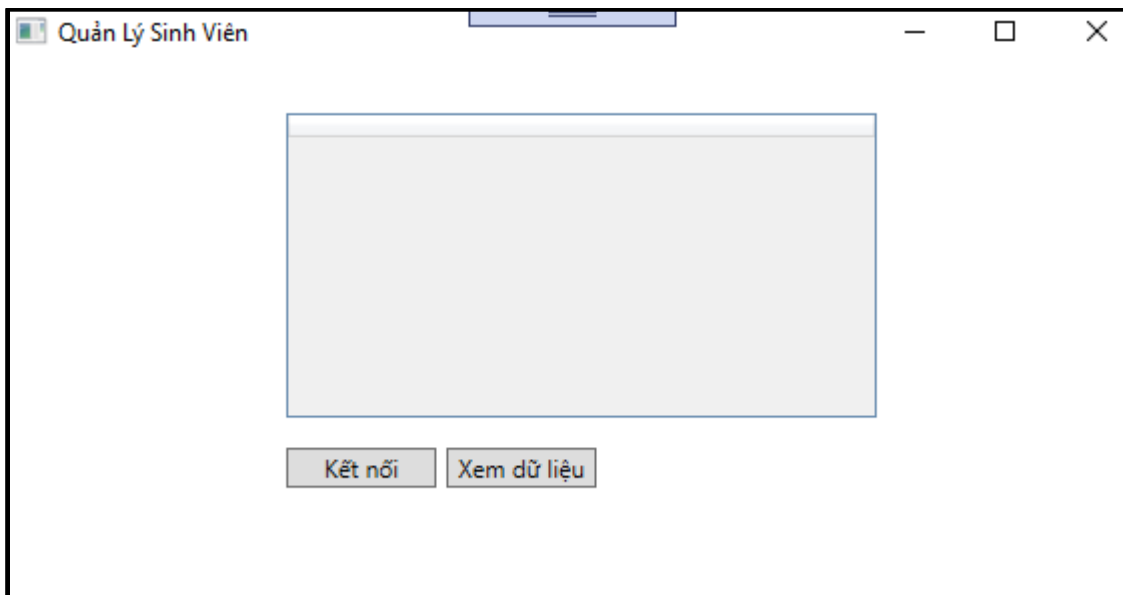
Bước 3: Thêm các thành viên đến lớp SinhVien như sau:

```
public class SinhVien
{
    public string MaSV { get; set; }
    public string TenSV { get; set; }
    public string Email { get; set; }
```

```
}
    public string MaKH { get; set; }
}
```

Đề ý rằng các thành viên của lớp SinhVien như MaSV, TenSV, Email và MaKH tương ứng với các thuộc tính (hay cột) của bảng SinhVien trong cơ sở dữ liệu QuanLySinhVien.

Bước 4: Thêm một điều khiển Button và một điều khiển DataGridView đến giao diện. Thay đổi thuộc tính Name của DataGridView thành *dulieu*, thuộc tính Name của Button là *btnDulieu* và thuộc tính Content của Button thành *Xem dữ liệu*. Điều chỉnh vị trí các điều khiển như hình sau:



Giao diện Form sau khi thêm nút Xem dữ liệu

Bước 5: Trong hàm xử lý sự kiện Click của nút Xem dữ liệu (*btnDulieu_Click*) chúng ta thêm đoạn mã sau:

```
private void btnDulieu_Click(object sender, RoutedEventArgs e)
{
    try
    {
        List<SinhVien> DanhSachSinhVien = new List<SinhVien>();
        using (SqlConnection connection =
            new SqlConnection(@"Server=DESKTOP-
                                O6BPUKE\SQLEXPRESS;Database=QuanLySinhVien;
                                Integrated Security=SSPI"))
        using (SqlCommand command =
            new SqlCommand("SELECT MaSV, TenSV, Email, MaKH FROM
                            SinhVien;", connection))
        {
            connection.Open();
            using (SqlDataReader reader = command.ExecuteReader())
```

```
        {
            while (reader.Read())
            {
                var sv = new SinhVien();
                sv.MaSV = reader.GetString(0);
                sv.TenSV = reader.GetString(1);
                sv.Email = reader.GetString(2);
                sv.MaKH = reader.GetString(3);
                DanhSachSinhVien.Add(sv);
            }
        }

        MessageBox.Show("Mở và đóng cơ sở dữ liệu thành công.");
        dulieu.ItemsSource = DanhSachSinhVien;
    }

    catch (Exception ex)
    {
        MessageBox.Show("Lỗi khi mở kết nối: " + ex.Message);
    }
}
```

Ở đây chúng ta dùng đối tượng Command để tạo lệnh truy vấn dữ liệu đến bảng SinhVien trong cơ sở dữ liệu QuanLySinhVien:

```
using (SqlCommand command =
    new SqlCommand("SELECT MaSV, TenSV, Email, MaKH FROM SinhVien;",
        connection))
```

Nếu kết nối và truy vấn thành công, dữ liệu nhận được sẽ được gán đến đối tượng DataReader thông qua phương thức ExecuteReader của đối tượng Command:

```
SqlDataReader reader = command.ExecuteReader()
```

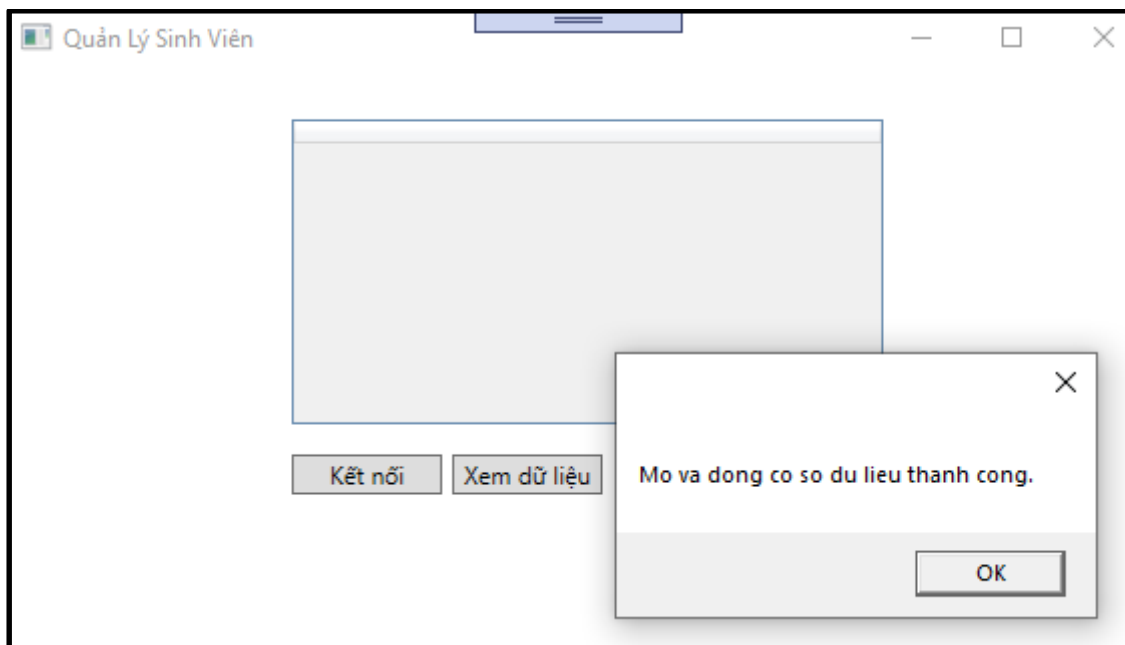
Từng hàng dữ liệu sẽ được đọc bằng phương thức Read của đối tượng DataReader và lưu trong một biến kiểu danh sách:

```
while (reader.Read())
{
    var sv = new SinhVien();
    sv.MaSV = reader.GetString(0);
    sv.TenSV = reader.GetString(1);
    sv.Email = reader.GetString(2);
    sv.MaKH = reader.GetString(3);
    DanhSachSinhVien.Add(sv);
}
```

Cuối cùng là hiển thị danh sách sinh viên nhận được đến DataGrid:

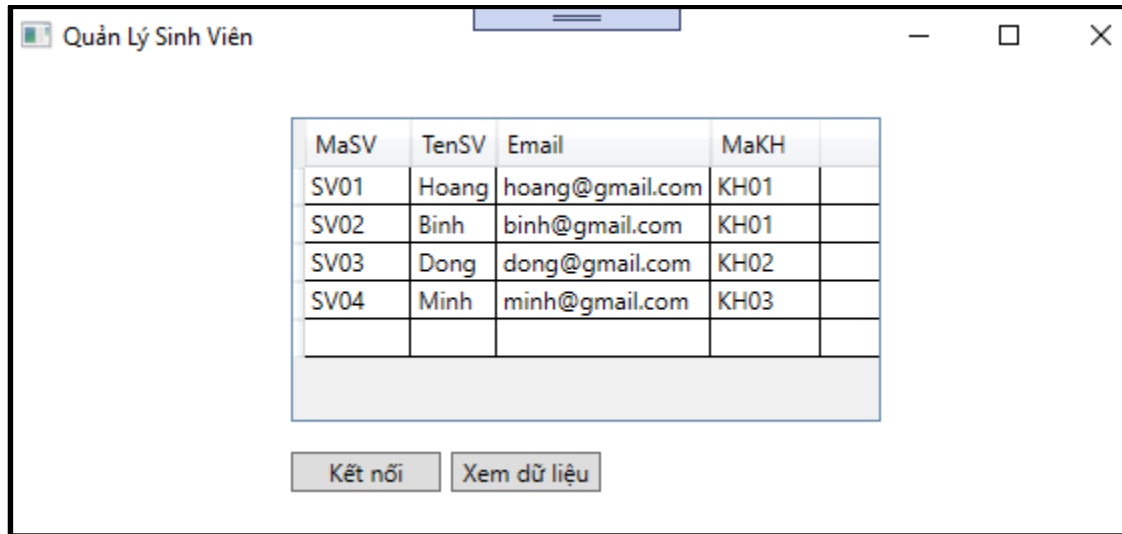
```
dulieu.ItemsSource = DanhSachSinhVien;
```

Bước 6: Kiểm tra kết quả bằng cách thực thi ứng dụng và nhấn nút *Xem dữ liệu* như sau:



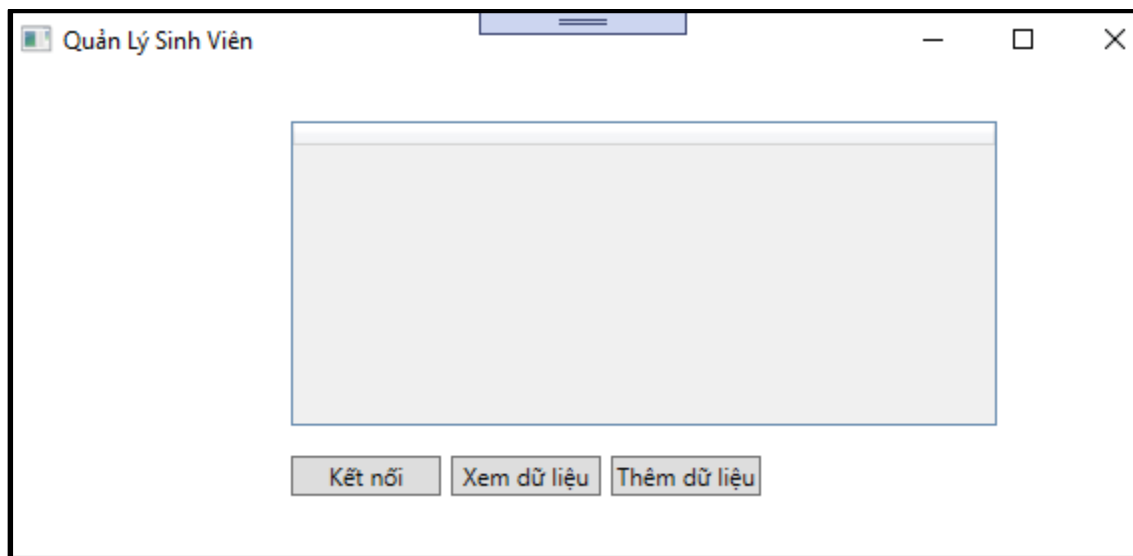
Kết nối cơ sở dữ liệu thành công khi nhấn nút Xem dữ liệu

Nhấn OK từ hộp thông điệp, kết quả:



Dữ liệu từ bảng SinhVien hiển thị trong DataGrid

Bước 7: Đóng ứng dụng. Thêm một Button mới đến giao diện, thay đổi thuộc tính Name của Button thành *btnThem* và thuộc tính Content thành *Thêm dữ liệu*:



Giao diện Form sau khi thêm nút Thêm dữ liệu

Bước 8: Thêm hàm xử lý sự kiện Click đến nút này với tên *btnThem_Click*. Phía dưới hàm này chúng ta thêm một hàm mới tên *Them_sinh_vien()* như sau:

```
private int Them_sinh_vien(SinhVien sinhvien)
{
    try
    {
        using (SqlConnection connection =
            new SqlConnection(@"Server=DESKTOP-
                                06BPUKE\SQLEXPRESS;Database=QuanLySinhVien;
```

```

        Integrated Security=SSPI"))
using (SqlCommand command = new SqlCommand("INSERT INTO
    SinhVien(MaSV,TenSV,Email,MaKH) " +
    "VALUES(@MaSV,@TenSV,@Email,@MaKH);", connection))
{
    command.Parameters.Add("MaSV", SqlDbType.NChar, 10).Value =
        sinhvien.MaSV;
    object dbTenSV = sinhvien.TenSV;
    if (dbTenSV == null)
    {
        dbTenSV = DBNull.Value;
    }
    command.Parameters.Add("TenSV", SqlDbType.VarChar, 20).Value =
        dbTenSV;
    object dbEmail = sinhvien.Email;
    if (dbEmail == null)
    {
        dbEmail = DBNull.Value;
    }
    command.Parameters.Add("Email", SqlDbType.NVarChar, 50).Value =
        dbEmail;
    command.Parameters.Add("MaKH", SqlDbType.NChar, 10).Value =
        sinhvien.MaKH;
    connection.Open();
    return command.ExecuteNonQuery();
}
}
catch (Exception ex)
{
    MessageBox.Show("Loi khi mo ket noi:" + ex.Message);
    return -1;
}
}

```

Chúng ta dùng đối tượng Command để tạo lệnh truy vấn thêm hàng mới đến bảng SinhVien như sau:

```

SqlCommand command = new SqlCommand("INSERT INTO
    SinhVien(MaSV,TenSV,Email,MaKH) " +
    "VALUES(@MaSV,@TenSV,@Email,@MaKH);", connection)

```

Chú ý các biểu tượng @ dùng để tham chiếu đến các giá trị các thuộc tính từ đối tượng lớp SinhVien. Cũng lưu ý các tham số MaSV, TenSV, Email, và MaKH khi thêm đến tập Parameters của đối tượng Command phải có kiểu dữ liệu tương ứng với kiểu dữ liệu chúng ta dùng cho các cột của bảng SinhVien trong cơ sở dữ liệu, ví dụ cột MaSV trong bảng SinhVien có kiểu nchar(10) nên trong phương thức Add của Parameters chúng ta viết:

```
command.Parameters.Add("MaSV", SqlDbType.NChar, 10).Value = sinhvien.MaSV;
```

Sử dụng SqlDbType để lấy kiểu dữ liệu tương ứng từ cơ sở dữ liệu SQL Server. Và vì rằng các cột TenSV (tương tự Email) trong bảng SinhVien được phép nhận giá trị NULL nên chúng ta cần kiểm tra như sau:

```
object dbTenSV = sinhvien.TenSV;
if (dbTenSV == null)
{
    dbTenSV = DBNull.Value;
}
```

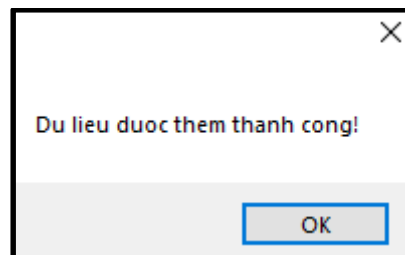
Khi kết nối và truy vấn thành công, chúng ta thực hiện phương thức NonExecuteQuery() của đối tượng Command:

```
command.ExecuteNonQuery();
```

Bước 9: Gọi phương thức Them_sinh_vien() trong hàm xử lý sự kiện btnThem_Click():

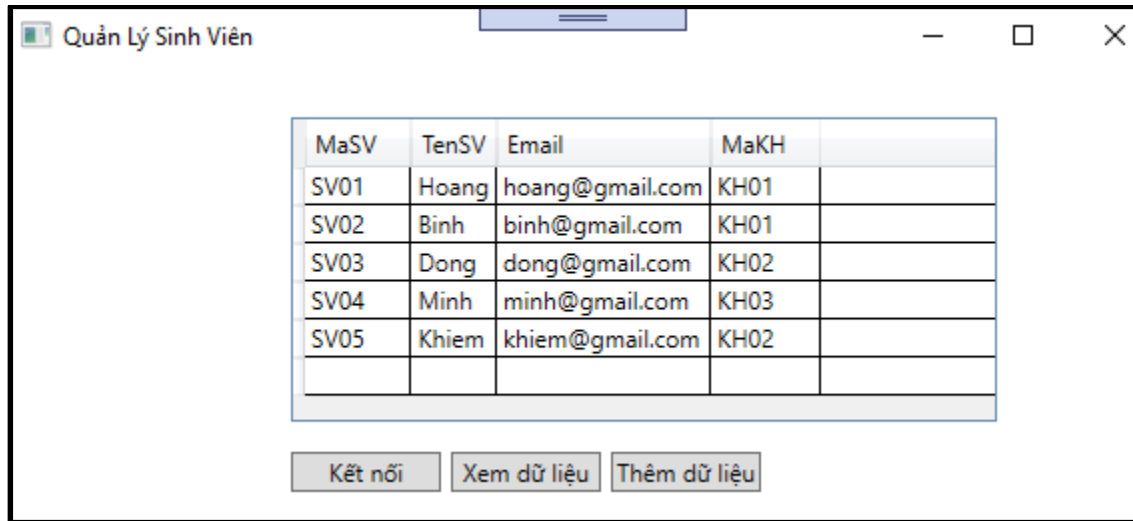
```
private void btnThem_Click(object sender, EventArgs e)
{
    SinhVien sv = new SinhVien();
    sv.MaSV = "SV06";
    sv.TenSV = "Kiem";
    sv.Email = "kiem@gmail.com";
    sv.MaKH = "KH02";
    if (Them_sinh_vien(sv) > 0)
        MessageBox.Show("Du lieu duoc them thanh cong!");
}
```

Bước 10: Lưu tất cả và thực thi ứng dụng. Nhấn nút Thêm dữ liệu nếu hiện thông điệp sau là chúng ta đã thêm thành công hàng mới:



Hộp thông báo khi thêm dữ liệu thành công

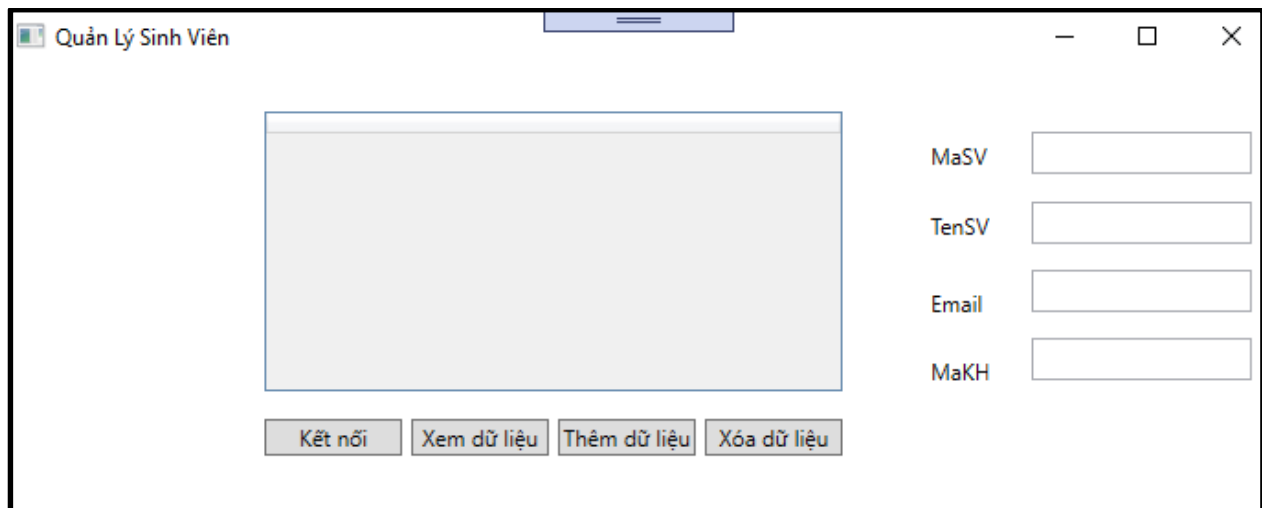
Để xem hàng vừa thêm, chúng ta nhấn nút Xem dữ liệu, kết quả:



Dữ liệu bảng SinhVien sau khi được thêm

Đóng ứng dụng. Nếu chúng ta mở dữ liệu bảng SinhVien trong cơ sở dữ liệu QuanLySinhVien, hàng mới cũng đã được thêm vào.

Bước 11: Thêm một Button mới đến giao diện, thay đổi các thuộc tính Name là btnXoa, thuộc tính Content là Xóa dữ liệu. Để tiện cho việc cập nhật dữ liệu chúng ta sẽ thực hiện trong phần sau của bài thực hành này, chúng ta thêm 4 điều khiển TextBox với các thuộc tính Name là txtMaSV, txtTenSV, txtEmail và txtMaKH tương ứng với 4 điều khiển Label. Thay đổi các thuộc tính Text của các TextBox thành chuỗi rỗng. Điều chỉnh giao diện như sau:



Giao diện sau khi thêm nút Xóa dữ liệu và các Textbox

Bước 12: Thêm hàm xử lý sự kiện đến nút Xóa dữ liệu tên btnXoa_Click(). Dưới hàm này chúng ta định nghĩa hàm mới tên Xoa_sinh_vien() như sau:

```
private int Xoa_sinh_vien(SinhVien sinhvien)
{
```

```

try
{
    using (SqlConnection connection =
        new SqlConnection(@"Server=DESKTOP-06BPUKE\SQLEXPRESS;
            Database=QuanLySinhVien;
            Integrated Security=SSPI"))
    using (SqlCommand command = new SqlCommand("DELETE FROM
        SinhVien " + "WHERE MaSV = @MaSV", connection))
    {
        command.Parameters.Add("MaSV", SqlDbType.NChar,
            10).Value = sinhvien.MaSV;

        connection.Open();

        return command.ExecuteNonQuery();
    }
}

catch (Exception ex)
{
    MessageBox.Show("Loi khi mo ket noi:" + ex.Message);

    return -1;
}
}

```

Chúng ta sử dụng đối tượng Command để tạo truy vấn Delete xóa một hàng theo khóa chính (MaSV):

```

SqlCommand command = new SqlCommand("DELETE FROM SinhVien " +
    "WHERE MaSV = @MaSV;", connection)

```

Phương thức ExecuteNonQuery() của đối tượng Command được sử dụng để thực thi truy vấn.

Bước 13: Gọi hàm Xoa_sinh_vien() trong hàm btnXoa_Click() như sau:

```

SinhVien sv = new SinhVien();
sv.MaSV = txtMaSV.Text;
if (Xoa_sinh_vien(sv) > 0)
    MessageBox.Show("Du lieu duoc xoa thanh cong!");

```

Bước 14: Lưu tất cả và thực thi ứng dụng. Nhấn nút Xem dữ liệu:

Quản Lý Sinh Viên

MaSV	TenSV	Email	MaKH	
SV01	Hoang	hoang@gmail.com	KH01	
SV02	Binh	binh@gmail.com	KH01	
SV03	Dong	dong@gmail.com	KH02	
SV04	Minh	minh@gmail.com	KH03	
SV05	Khiem	khiem@gmail.com	KH02	

MaSV:
TenSV:
Email:
MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu

Dữ liệu bảng Sinh Viên trước khi xóa

Nhập SV05 vào ô MaSV và nhấn nút Xóa dữ liệu:

Quản Lý Sinh Viên

MaSV	TenSV	Email	MaKH	
SV01	Hoang	hoang@gmail.com	KH01	
SV02	Binh	binh@gmail.com	KH01	
SV03	Dong	dong@gmail.com	KH02	
SV04	Minh	minh@gmail.com	KH03	
SV05	Khiem	khiem@gmail.com	KH02	

MaSV: SV05
TenSV:
Email:
MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu

Du lieu duoc xoa thanh cong!

OK

Hộp thông báo khi xóa thành công

Nhấn OK và nhấn lại nút Xem dữ liệu:

MaSV	TenSV	Email	MaKH
SV01	Hoang	hoang@gmail.com	KH01
SV02	Binh	binh@gmail.com	KH01
SV03	Dong	dong@gmail.com	KH02
SV04	Minh	minh@gmail.com	KH03

MaSV:

TenSV:

Email:

MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu

Dữ liệu bảng SinhVien sau khi xóa

Đóng ứng dụng.

Bước 15: Thêm một Button mới đến giao diện, thay đổi các thuộc tính Name thành btnCapnhat, Content thành Cập nhật dữ liệu. Điều chỉnh lại giao diện như sau:

MaSV:

TenSV:

Email:

MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu Cập nhật dữ liệu

Giao diện Form sau khi thêm nút Cập nhật dữ liệu

Bước 16: Thêm hàm xử lý sự kiện Click đến nút Cập nhật dữ liệu. Bên dưới hàm btnCapnhat_Click() định nghĩa hàm Cap_nhat_sinh_vien() như sau:

```
private int Cap_nhat_sinh_vien(SinhVien sinhvien)
{
    try
    {
        using (SqlConnection connection =
            new SqlConnection(@"Server=DESKTOP06BPUKE\SQLEXPRESS;
                               Database=QuanLySinhVien;
                               Integrated Security=SSPI"))
```

```

using (SqlCommand command = new SqlCommand("UPDATE SinhVien " +
    "SET TenSV = @TenSV, Email = @Email,
        MaKH = @MaKH " +
    "WHERE MaSV = @MaSV", connection))
{
    command.Parameters.Add("MaSV", SqlDbType.NChar, 10).Value =
        sinhvien.MaSV;
    command.Parameters.Add("TenSV", SqlDbType.VarChar, 20).Value =
        sinhvien.TenSV;
    command.Parameters.Add("Email", SqlDbType.NVarChar, 50).Value =
        sinhvien.Email;
    command.Parameters.Add("MaKH", SqlDbType.NChar, 10).Value =
        sinhvien.MaKH;
    connection.Open();
    return command.ExecuteNonQuery();
}
}

catch (Exception ex)
{
    MessageBox.Show("Loi khi mo ket noi:" + ex.Message);

    return -1;
}
}

```

Chúng ta dùng đối tượng Command để tạo truy vấn Update để chỉnh sửa và cập nhật dữ liệu theo khóa chính:

```

SqlCommand command = new SqlCommand("UPDATE SinhVien " +
    "SET TenSV = @TenSV, Email = @Email,
        MaKH = @MaKH " +
    "WHERE MaSV = @MaSV;", connection)

```

Dùng phương thức NonExecuteQuery() để thực thi truy vấn.

Bước 17: Thêm đoạn mã sau đến hàm xử lý sự kiện btnCapnhat_Click:

```

SinhVien sv = new SinhVien();
sv.MaSV = txtMaSV.Text;
sv.TenSV = txtTenSV.Text;
sv.Email = txtEmail.Text;
sv.MaKH = txtMaKH.Text;
if (Cap_nhat_sinh_vien(sv) > 0)
    MessageBox.Show("Du lieu duoc cap nhat thanh cong!");

```

Bước 18: Lưu tất cả và thực thi ứng dụng. Nhấn nút Xem dữ liệu:

Quản Lý Sinh Viên

MaSV	TenSV	Email	MaKH	
SV01	Hoang	hoang@gmail.com	KH01	
SV02	Binh	binh@gmail.com	KH01	
SV03	Dong	dong@gmail.com	KH02	
SV04	Minh	minh@gmail.com	KH03	

MaSV:
 TenSV:
 Email:
 MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu Cập nhật dữ liệu

Dữ liệu bảng SinhVien trước khi cập nhật

Giả sử chúng ta muốn thay đổi thông tin về TenSV, Email và MaKH của sinh viên có MaSV là SV04. Nhập các thông tin về sinh viên cần cập nhật thông tin vào các TextBox và nhấn nút Cập nhật dữ liệu:

Quản Lý Sinh Viên

MaSV	TenSV	Email	MaKH	
SV01	Hoang	hoang@gmail.com	KH01	
SV02	Binh	binh@gmail.com	KH01	
SV03	Dong	dong@gmail.com	KH02	
SV04	Minh	minh@gmail.com	KH03	

MaSV:
 TenSV:
 Email:
 MaKH:

Kết nối Xem dữ liệu Thêm dữ liệu Xóa dữ liệu Cập nhật dữ liệu

Du lieu duoc cap nhat thanh cong!

OK

Hộp thông báo khi cập nhật thành công

Nhấn OK đóng MessageBox và nhấn lại nút Xem dữ liệu:

The screenshot shows a Windows application window titled "Quản Lý Sinh Viên". Inside the window, there is a table with the following data:

MaSV	TenSV	Email	MaKH	
SV01	Hoang	hoang@gmail.com	KH01	
SV02	Binh	binh@gmail.com	KH01	
SV03	Dong	dong@gmail.com	KH02	
SV04	Dung	dung@gmail.com	KH03	

Below the table, there are five buttons: "Kết nối", "Xem dữ liệu", "Thêm dữ liệu", "Xóa dữ liệu", and "Cập nhật dữ liệu". To the right of the table, there is a form with four input fields labeled "MaSV", "TenSV", "Email", and "MaKH". The values entered in these fields are "SV04", "Dung", "dung@gmail.com", and "KH03" respectively.

Đóng ứng dụng. Chúng ta đã hoàn thành các chức năng kết nối, xem, thêm, xóa và cập nhật dữ liệu dùng các đối tượng Connection, Command và DataReader.

Kết thúc bài thực hành số 2.