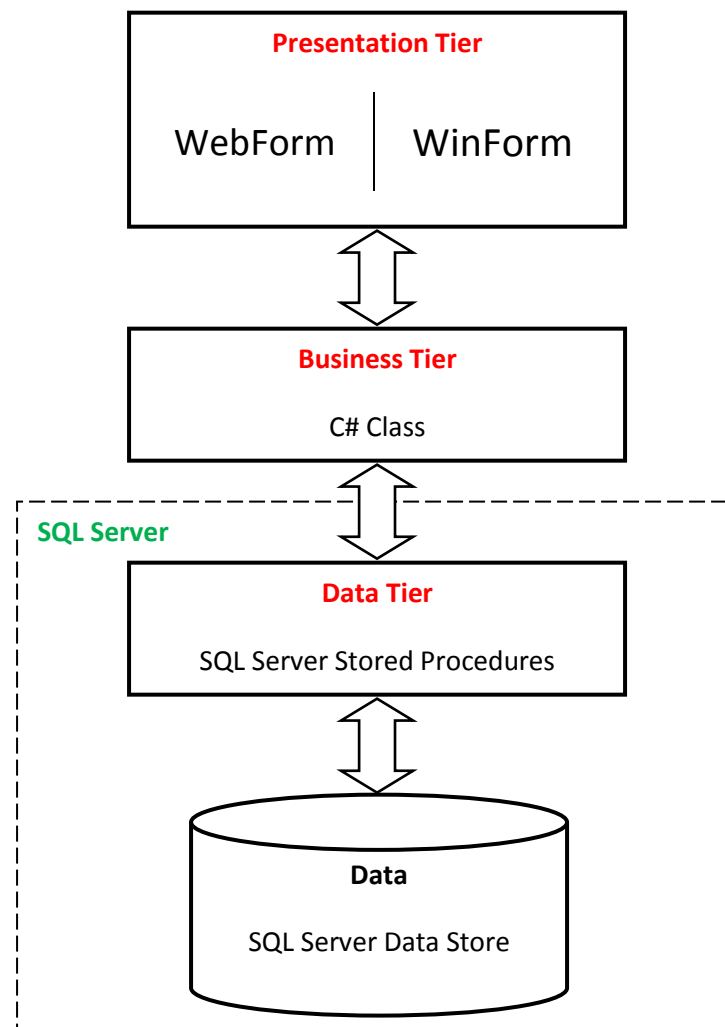


### Mô hình 3 tầng (three tier)

Chia nhỏ chức năng của ứng dụng thành các thành phần dựa trên công việc mà thành phần đó thực hiện và nhóm mỗi thành phần đó thành một tầng.

#### Mục đích

Phát triển phần mềm dựa trên mô hình 3 tầng cho phép các lập trình viên có thể cập nhật, thay đổi chức năng của mỗi tầng riêng biệt mà không phá vỡ code của các tầng khác.



Tên và chức năng của mỗi tầng trong mô hình 3 tầng

**Data Tier (Tầng dữ liệu):** bao gồm các bảng trong CSDL, các thủ tục ...

**Business Tier (Tầng nghiệp vụ):** các lớp C#. Trong tầng này có thư viện **DataAccess** chứa các lớp thực hiện việc lấy, thêm, xóa, sửa dữ liệu từ CSDL

**Presentation Tier (Tầng trình bày/giao diện):** có thể sử dụng nhiều công nghệ khác nhau và chứa phần giao diện của ứng dụng

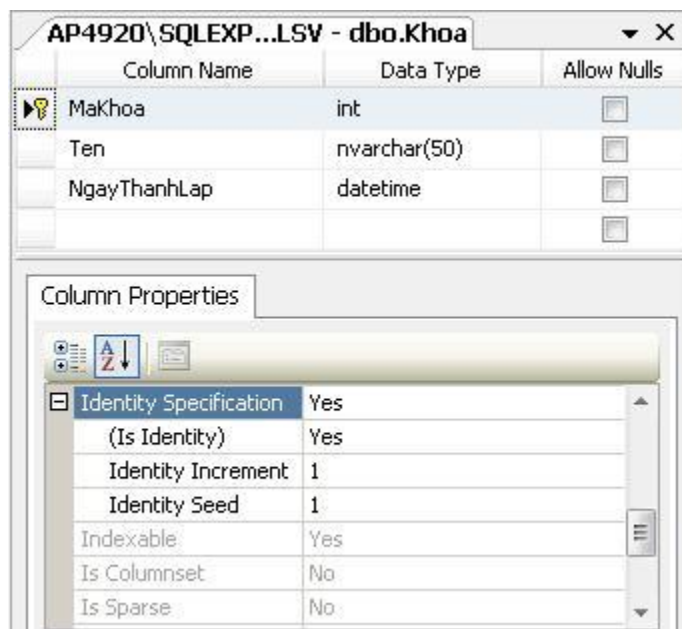
- Web: ASP.NET Web Form, ASP.NET User Control, Control ...
- Windows Form: Form, User Control, Button, Textbox, Label ...

## Ví dụ

### Tầng Data

#### Tạo CSDL và bảng

Tạo một CSDL đặt tên là **ViDu** và tạo một bảng đặt tên là **Khoa** như sau



Trong đó cột MaKhoa là khóa chính kiểu int có giá trị tự tăng (thiết lập theo như trong hình)

#### Tạo thủ tục

Tạo các thủ tục thực hiện các thao tác thêm, xóa, sửa dữ liệu trong bảng Khoa

*Thủ tục Khoa\_Them*

```

CREATE PROCEDURE [Khoa_Them]
    @MaKhoa int output,
    @Ten nvarchar(50),
    @Ngay datetime
AS
INSERT INTO Khoa (Ten, NgayThanhLap)
VALUES (@Ten, @Ngay)

SET @MaKhoa = @@IDENTITY

```

### *Thủ tục Khoa\_Sua*

```

CREATE PROCEDURE [Khoa_Sua]
    @MaKhoa int,
    @Ten nvarchar(50),
    @Ngay datetime
AS
IF EXISTS (SELECT MaKhoa FROM Khoa WHERE MaKhoa = @MaKhoa)
BEGIN
    UPDATE Khoa
    SET Ten = @Ten,
        NgayThanhLap = @Ngay
    WHERE MaKhoa = @MaKhoa
END

```

### *Thủ tục Khoa\_Xoa*

```

CREATE PROCEDURE [Khoa_Xoa]
    @MaKhoa int
AS

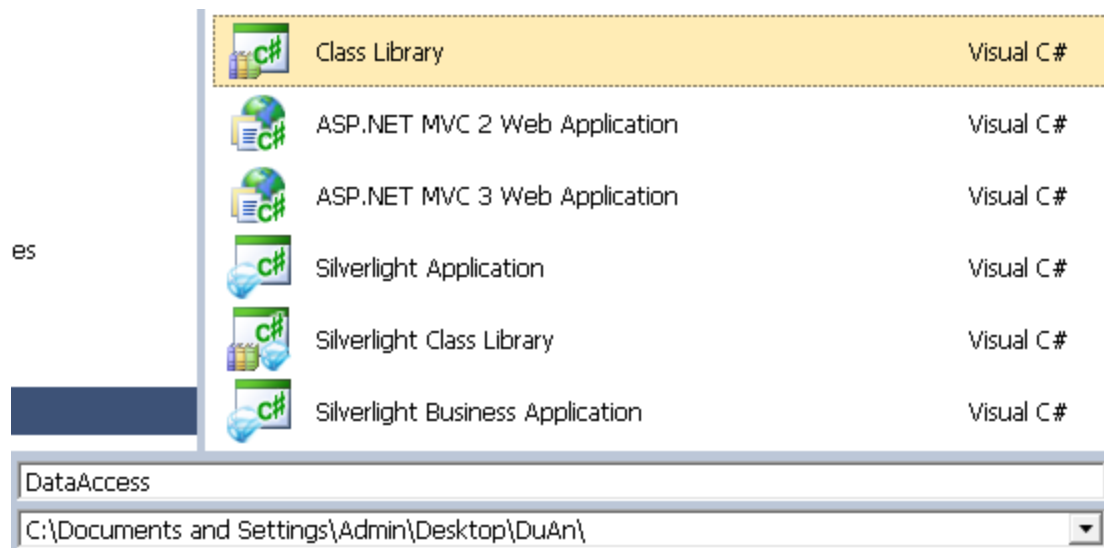
DELETE Khoa WHERE MaKhoa = @MaKhoa

```

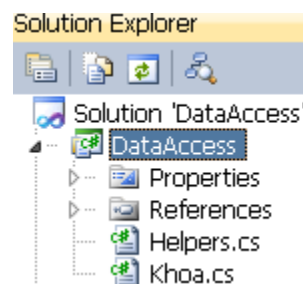
## Tầng Business

Tạo một thư mục mới ở Desktop và đặt tên là **DuAn**

Mở Visual Studio và tạo một dự án kiểu Class Library tên là DataAccess và lưu vào trong thư thực DuAn vừa tạo ở Desktop như hình sau



Tạo 2 class trong dự án DataAccess và đặt tên lần lượt là **Helpers.cs** và **Khoa.cs** như hình



- **Helpers.cs**: lớp chứa các thuộc tính / phương thức lấy chuỗi kết nối
- **Khoa.cs**: lớp ánh xạ từ bảng Khoa và chứa các phương thức gọi các thủ tục đã tạo ở bước trên

Mở lớp **Helpers.cs** và gõ đoạn mã sau

```
using System;

namespace DataAccess
{
    public class Helpers
    {
        private static string _ConnectionString = "";
        public static string ConnectionString
        {
            get
            {
                if (_ConnectionString == "")
                {
                    // gan tinh chuoi ket noi
                    _ConnectionString = @"server=.\sqlexpress;database=ViDu;integrated
security=true";
                }
            }
        }
    }
}
```

```

        // hoặc lay chuoi ket noi tu app.config
        // ...
    }
    return _ConnectionString;
}
}
}
}
}

```

Mở lớp **Khoa.cs** và gõ đoạn mã sau

```

using System;
using System.Data.SqlClient;

namespace DataAccess
{
    public class Khoa
    {
        // cac thuoc tinh & truong (anh xa tu cac cot trong ban Khoa)
        private int _MaKhoa;
        public int MaKhoa
        {
            get { return _MaKhoa; }
            set { _MaKhoa = value; }
        }

        private string _Ten;
        public string Ten
        {
            get { return _Ten; }
            set { _Ten = value; }
        }

        private DateTime _NgayThanhLap;
        public DateTime NgayThanhLap
        {
            get { return _NgayThanhLap; }
            set { _NgayThanhLap = value; }
        }

        // Cac phuong thuc goi cac thu tuc them xoa sua bang

        // Them 1 dong vao bang vao khoa va tra ve gia tri ID tu tang (MaKhoa)
        public static int Them(Khoa khoa)
        {
            using (SqlConnection cnn = new SqlConnection(Helpers.ConnectionString))
            {
                SqlCommand cmd = cnn.CreateCommand();

                // Dat ten thu tuc
                cmd.CommandText = "Khoa_Them";
                cmd.CommandType = System.Data.CommandType.StoredProcedure;

                // Gan tham so
                cmd.Parameters.Add("@MaKhoa", System.Data.SqlDbType.Int).Direction
                    = System.Data.ParameterDirection.Output;
            }
        }
    }
}

```

```

        cmd.Parameters.Add("@Ten", System.Data.SqlDbType.NVarChar, 50).Value =
khoa.Ten;
        cmd.Parameters.Add("@Ngay", System.Data.SqlDbType.DateTime).Value =
khoa.NgayThanhLap;

        // mo ket noi
        cnn.Open();
        // thuc thi cau truy van
        // rs = so dong duoc them vao
        int rs = cmd.ExecuteNonQuery();
        cnn.Close();

        // neu rs > 0: them thanh cong
        if (rs > 0)
        {
            // lay gia tri id tu tang thong qua tham so output
            int id = (int)cmd.Parameters["@MaKhoa"].Value;
            return id;
        }
        else
        {
            return 0;
        }
    }
}

public static int Sua(Khoa khoa)
{
    using (SqlConnection cnn = new SqlConnection(Helpers.ConnectionString))
    {
        SqlCommand cmd = cnn.CreateCommand();

        // Dat ten thu tuc
        cmd.CommandText = "Khoa_Sua";
        cmd.CommandType = System.Data.CommandType.StoredProcedure;

        // Gan tham so
        cmd.Parameters.Add("@MaKhoa", System.Data.SqlDbType.Int).Value =
khoa.MaKhoa;
        cmd.Parameters.Add("@Ten", System.Data.SqlDbType.NVarChar, 50).Value =
khoa.Ten;
        cmd.Parameters.Add("@Ngay", System.Data.SqlDbType.DateTime).Value =
khoa.NgayThanhLap;

        // mo ket noi
        cnn.Open();
        // thuc thi cau truy van
        // rs = so dong duoc cap nhat
        // > 0: cap nhat thanh cong
        int rs = cmd.ExecuteNonQuery();
        cnn.Close();
        return rs;

        //return cmd.ExecuteNonQuery();
    }
}

public static int Xoa(int maKhoa)

```

```

{
    using (SqlConnection cnn = new SqlConnection(Helpers.ConnectionString))
    {
        SqlCommand cmd = cnn.CreateCommand();

        // Dat ten thu tuc
        cmd.CommandText = "Khoa_Xoa";
        cmd.CommandType = System.Data.CommandType.StoredProcedure;

        // Gan tham so
        cmd.Parameters.Add("@MaKhoa", System.Data.SqlDbType.Int).Value = maKhoa;

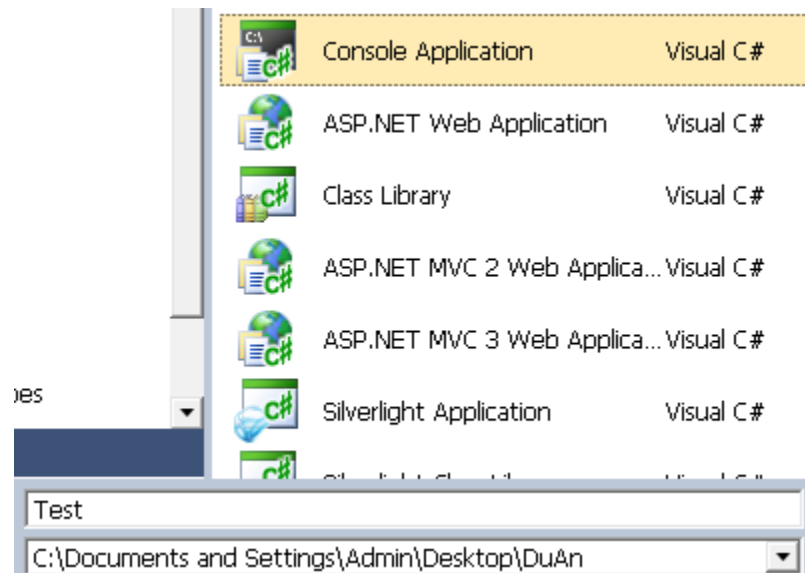
        // mo ket noi
        cnn.Open();
        // thuc thi cau truy van
        // rs = so dong duoc xoa
        // > 0: xoa thanh cong
        int rs = cmd.ExecuteNonQuery();
        cnn.Close();
        return rs;
        //return cmd.ExecuteNonQuery();
    }
}
}
}

```

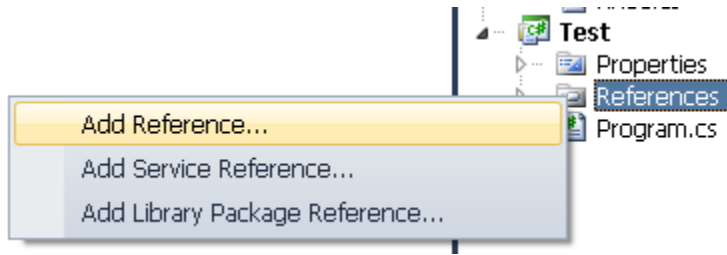
## Tầng Presentation

Thực hiện việc kiểm tra các lớp đã viết ở tầng Business trên ứng dụng Console. Thực

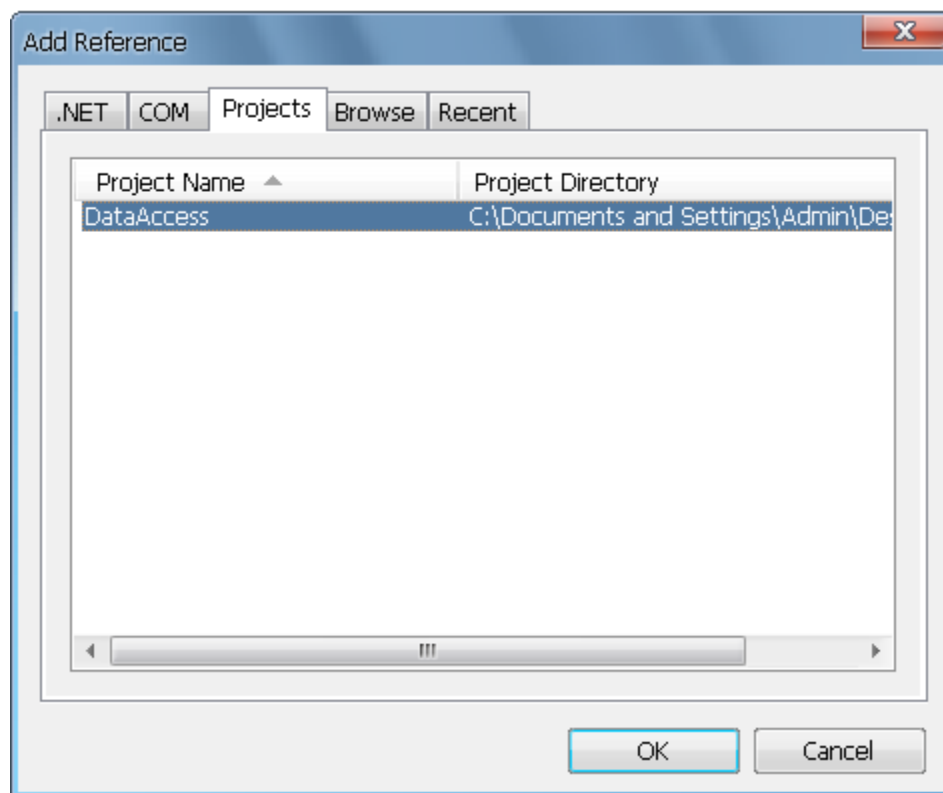
1. Vào File > Add > New Project để thêm một dự án mới.
2. Một cửa sổ mới sẽ xuất hiện và chọn kiểu dự án là Console trong cửa sổ đó



- Đặt tên dự án là Test và chọn thư mục lưu là **DuAn** (tương tự như bước trước)
- Bấm chuột phải vào thư mục Reference trong dự án Test và chọn Add Reference



- Chọn thẻ Projects, sau đó chọn DataAccess và bấm OK



- Viết một số đoạn mã để kiểm tra các phương thức ở tầng Business như sau (viết trong hàm Main)

Đoạn code mẫu để kiểm tra việc thêm dữ liệu

```
Khoa data = new Khoa();  
data.Ten = "Khoa CNTT";  
data.NgayThanhLap = DateTime.Parse("12/30/2002");  
  
int rs = Khoa.Them(data);  
if (rs > 0)  
    Console.WriteLine("Them thanh cong. MaKhoa = " + rs);
```



```
else  
    Console.WriteLine("Khong them duoc");
```

Đoạn code mẫu để kiểm tra việc cập nhật dữ liệu

```
Khoa data = new Khoa();  
data.MaKhoa = 2;  
data.Ten = "Khoa Cong nghe thong tin";  
data.NgayThanhLap = DateTime.Parse("12/30/2003");  
  
bool rs = Khoa.Sua(data);  
if (rs > 0)  
    Console.WriteLine("Cap nhat thanh cong");  
else  
    Console.WriteLine("Khong thanh cong");
```

Tương tự kiểm tra xóa dữ liệu (sinh viên tự thực hiện)

Đề nghị các bạn chỉnh sửa lại các đoạn code mẫu phía trên cho phép nhập các thông tin từ bàn phím