Điều khiển ADO.NET

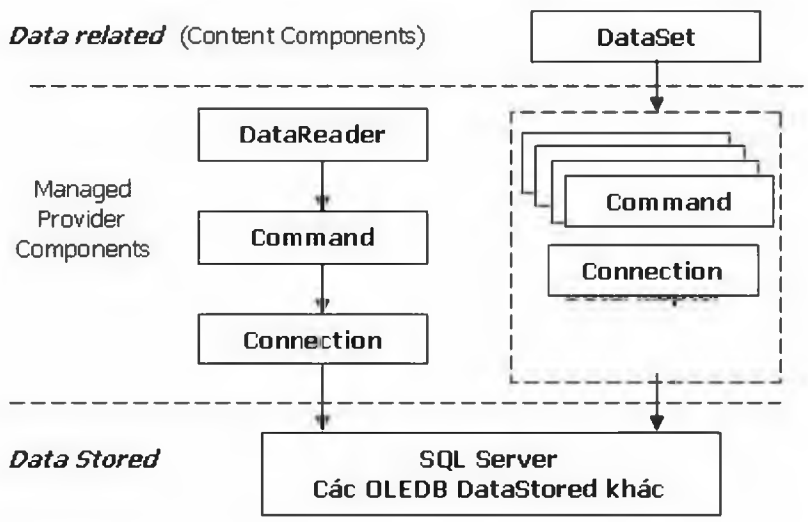
Hầu hết ứng dụng hay các website đều cần phải có cơ sở dữ liệu, để lưu trữ dữ liệu, xử lý thông tin và đưa ra các báo cáo, hỗ trợ tìm kiếm...

Khi dữ liệu trở thành trung tâm của ứng dụng thì cung cấp các chức năng tới người dùng phụ thuộc vào khả năng thao tác dữ liệu, vấn đề mà người thiết kế và người xây dựng ứng dụng quan tâm khi sử dụng dữ liệu là:

- Lưu dữ liệu tập trung
- Đảm bảo toàn vẹn dữ liệu
- Đảm bảo khả năng truy xuất đồng thời của nhiều người dùng trên dữ liệu
- Đảm bảo thời gian hồi đáp ngắn cho mỗi người dùng
- Bảo mật dữ liệu
- Trao đổi dữ liệu giữa các hệ thống khác nhau

Những vấn đề này được giải quyết dựa vào khả năng của các hệ quản trị cơ sở dữ liệu(HQT CSDL) và các phần mềm xử lý dữ liệu do HQL CSDL cung cấp. Net truy xuất dữ liệu qua ADO.NET, đặc điểm chính của ADO.NET là khả năng làm việc vơi dữ liệu không kết nối, dữ liệu được lưu trữ trong bộ nhớ như một csdl thu nhỏ gọi là dataset, nhằm tăng tốc độ tính toán, xử lý tính toán và hạn chế sử dụng tài nguyên trên Database Server. ĐẶc điểm quan trọng thứ 2 là khả năng xử lý dữ liệu chuẩn XML, dữ liệu ở dạng XMl có thể trao đổi giữa bất kỳ hệ thống nào nên ứng dụng của bạn sẽ có nhiều khả năng làm việc với nhiều ứng dụng khác.

I. Kiến trúc ADO .Net



Hình 1

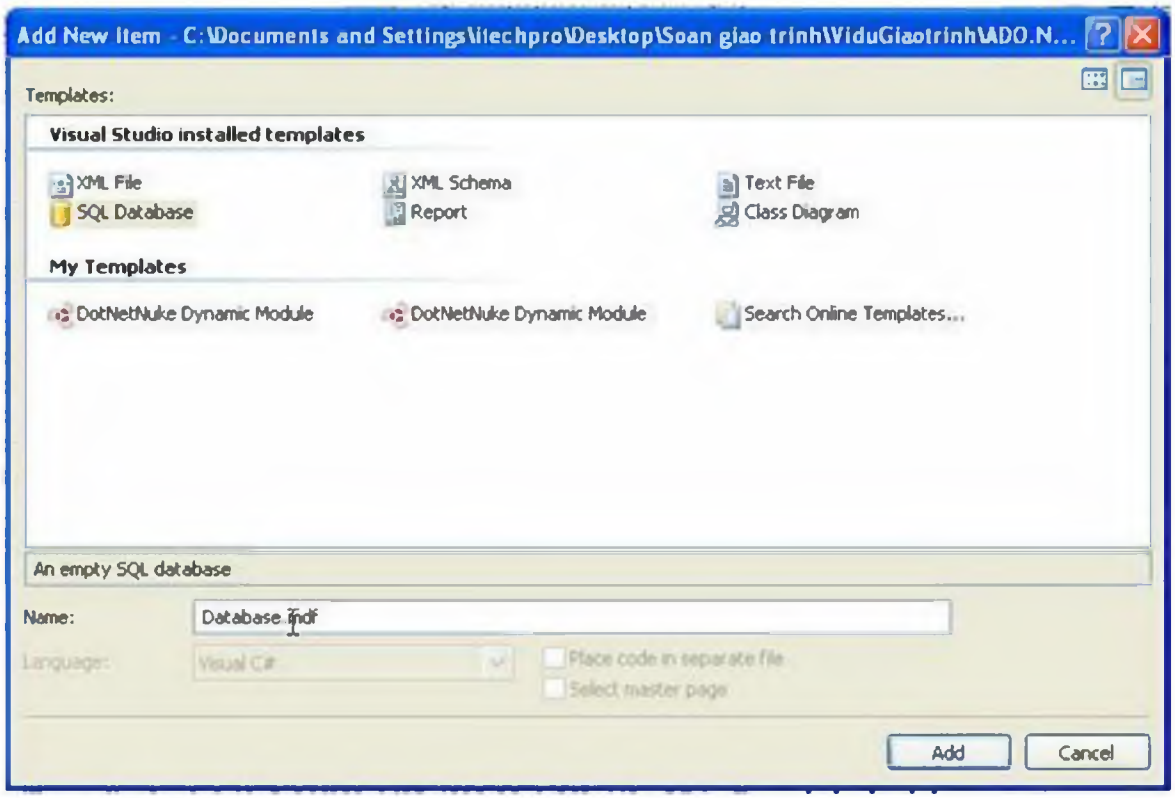
Kiến trúc ADO.NET có thể chia làm 2 phần chính:

- ***Managed Provider Component: bao gồm các đối tượng như DataAdapter, DataReader,...*** giữ nhiệm vụ làm việc trực tiếp với dữ liệu như database, file,...
- ***Content Component: bao gồm các đối tượng như DataSet, DataTable,...*** đại diện cho dữ liệu thực sự cần làm việc. DataReader là đối tượng mới, giúp truy cập dữ liệu nhanh chóng nhưng forward-only và read-only giống như ADO RecordSet sử dụng Server cursor, OpenFowardOnly và LockReadOnly.
  DataSet cũng là một đối tượng mới, không chỉ là dữ liệu, DataSet có thể coi là một bản sao gọn nhẹ của CSDL trong bộ nhớ với nhiều bảng và các mối quan hệ.
  DataAdapter là đối tượng kết nối giữa DataSet và CSDL, nó bao gồm 2 đối tượng Connection và Command để cung cấp dữ liệu cho DataSet cũng như cập nhật dữ liệu từ DataSet xuống CSDL.

Trước khi đi vào học cụ thể các đối tượng của ADO.NET chúng ta cùng xem qua một ví dụ HelloWorld với ADO.NET qua đó bạn sẽ thấy được công việc cần thực hiện khi thao tác với database(ở ví dụ này mình dùng với SQLExpress).

Để làm ví dụ này bạn thực hiện theo các bước sau:

- bước 1. Nhấn chuột phải vào thư mục App_Data chọn new Item, Cửa sổ Add New Item hiện ra bạn chọn SqlDatabase như hình 1 sau



Hình 2

- bước 2. Nhập tên Database vào hộp Name sau đó nhấn Add.

Trong Solution Explorer sẽ thêm vào Database trong thư mục App_Data.

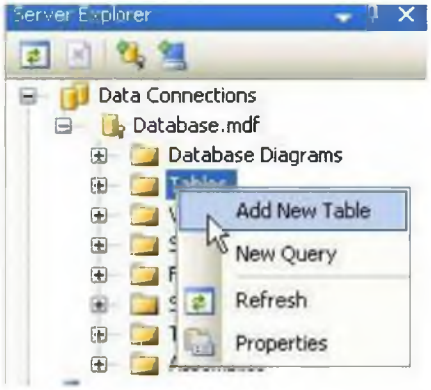
Hình 3

- bước 3. tạo bảng dữ liệu cho Database.mdf

- bước 3.1 bạn click đúp chuột vào Datatabase.mdf -> Server Explorer hiện ra như sau:

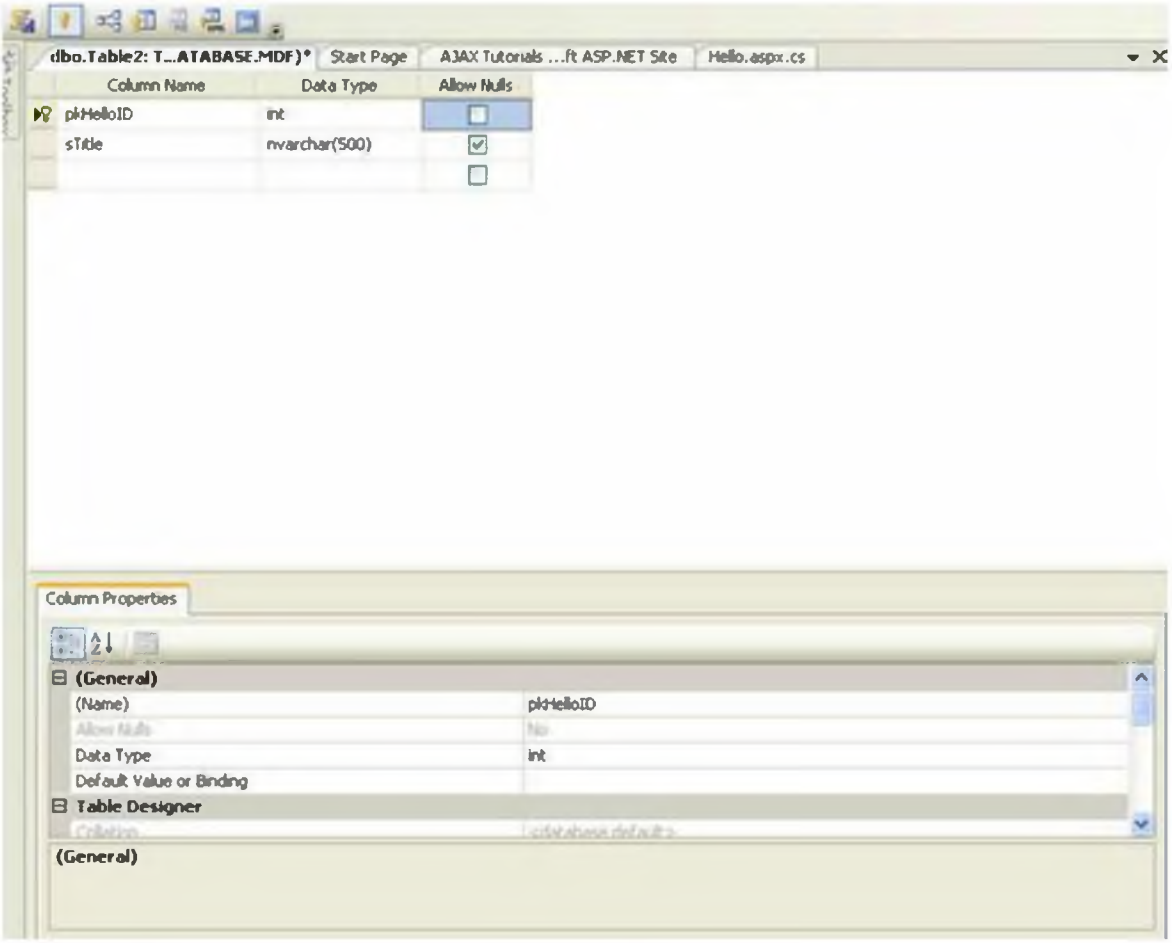
Hình 4

- bước 3.2 bạn nhấn chuột phải vào Tables và chọn Add New Table



Hình 5

Trong màn hình của VS sẽ hiện ra như hình 6 và bạn thao tác tạo các trường dữ liệu như thao tác với Access hay MSSQL 2000/2005



Hình 6

- bước 4.Viết code cho Hello.aspx.cs

```
using System;

using System.Data;

using System.Data.SqlClient;


public partial class _Default : System.Web.UI.Page

{

    protected void Page_Load(object sender, EventArgs e)
```

```
    {

        //chuỗi kết nối đến nguồn dữ liệu

        string driver = "Data
Source=(local)\\SQLEXPRESS;AttachDbFilename=|DataDirectory|Database.mdf;Integ
rated Security=True;User Instance=True";

        //đối tượng kết nối tới cơ sở dữ liệu

        SqlConnection sqlconn = new SqlConnection(driver);

        //Command điều khiển truy vấn sql

        SqlCommand sqlcom = sqlconn.CreateCommand();

        sqlcom.CommandText = "select sTitle from tblHello where pkHelloID=1";

        //mở kết nối dữ liệu

        sqlconn.Open();

        //lấy về chuỗi giá trị trong cơ sở dữ liệu

        string result = (string)sqlcom.ExecuteScalar();

        //đóng kết nối

        sqlconn.Close();

        //in giá trị ra màn hình

        Response.Write(result);

    }

}
```

Cơ bản các bước thực hiện với database

- bước 1: Tại kết nối
- bước 2: Tạo lệnh điều khiển truy vấn SQL
- bước 3:Mở kết nối dữ liệu
- bước 4: thực thi lệnh
- bước 5: đóng kết nối

- bước 6: in kết quả

II. Đối tượng Connection

Kết nối cơ sở dữ liệu SQLServer

Bạn cần nhập khẩu lớp SqlClient

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

Khai báo và khởi tạo:

```
SqlConnection sqlcon;

string driver="server=localhost; UID=sa; PWD=; database=name_database";

sqlcon=new Sqlconnection();

sqlcon.ConnectionString=driver;
```

Driver là chuỗi kết nối đến cơ sở dữ liệu trong trường hợp này mình kết nối với sqlserver 2000/2005

Kết nối với cơ sở dữ liệu Access

Bạn cận nhập khẩu lớp OleDb

```
using System.Data.OleDb;

OleDbConnection oleconn;

string driver = "Provider=Microsoft.jet.OLEDB.4.0; Data
Source=duongdan_tendata";

oleconn = new OleDbConnection();

oleconn.ConnectionString = driver;
```

1. Thuộc tính:

ConnectString: chứa đựng chuỗi kết nối tới cơ sở dữ liệu

Database: Chứa đựng tên cơ sở dữ liệu trong chuỗi kết nối ConnectString ở trên và bạn có thể thay đổi cơ sở dữ liệu trong lúc thực thi bằng phương thức ChangeDataBase:

Sqlconn.ChangeDatabase("name_database_thaydoi");

Server: tên máy chủ bạn trỏ tới

Connect Timeout: số thời gian(tính bằng giây) chờ kết nối dữ liệu mặc đình là 15giây, nếu trong khoảng thời gian này mà vẫn chưa kết nối xong một lỗi Connect Timeout được đưa ra.

State: trả về trạng thái của đối tượng SqlConnection: bạn có thể kiểm tra trạng thái của State như sau

```
Response.Write(sqlconn.State)
```

2. Phương thức của đối tượng Connection

Các phương thức của đối tượng Connection

Open: cho phép mở dữ liệu với các thuộc tính đã khai báo trong ConnectString

Close: Đóng cơ sở dữ liệu đang mở

CreateCommand: phương thức cho phép gán hay trả về một đối tượng Command ứng với đối tượng Connection, như ví dụ HelloWorld

```
SqlConnection sqlconn = new SqlConnection(driver);

SqlCommand sqlcom = sqlconn.CreateCommand();

sqlcom.CommandText = "select sTitle from tblHello where pkHelloID=1";
```

BeginTransaction: Phương thức này khai báo bắt đầu một chuyển tác, để kết thúc chuyển tác bạn dùng Table Commit

Rollback: trong trương hợp có lỗi trong quá trình thực thi bạn có thể sử dụng phương thức Rollback để huỷ bỏ các chuyển tác đã thực hiện.

Dispose: dùng để huỷ bỏ hay giải phóng đối tượng Connection đang sử dụng

III. Đối tượng SqlCommand

Khai báo và khởi tạo đối tượng

Cách 1:

```
SqlCommand sqlcom;

sqlcom=new SqlCommand(ssql,sqlconn)
```

cách 2:

```
SqlCommand sqlcom = new SqlCommand();

sqlcom.Connection = sqlconn;

sqlcom.CommandType = CommandType.Text;

sqlcom.CommandText = "select sTitle from tblHello where pkHelloID=1";
```

Phương thức

- ExcuteReader: dùng để thực thi đọc cơ sở dữ liệu từ bảng cơ sở dữ liệu

- ExcuteNoneQuery: Dùng để thực thi các phát biểu T-Sql như: Insert, Update, Delete, Create,...

- ExcuteScalar: trả về từ phát biết SQL dạng Select chỉ có một cột một hàng.

IV. Đối tượng SqlDataReader

đối tượng này được net cung cấp để đọc dữ liệu từ bảng cơ sở dữ liệu, nó là đối tượng chỉ phục vụ thao tác đọc dữ liệu(Read only). Trong khi truy xuất dữ liệu nó sẽ giữ kết nối liên tục với database(hướng kết nối)

Khai báo và khởi tạo đối tượng

```
SqlDataReader sqlreader;

sqlreader = sqlcom.ExecuteReader();
```

V. Đối tượng DataAdapter

OleDataAdapter được xem như bộ đọc dữ liệu từ cơ sở dliệu nguồn và điền chúng vào đối tượng DataSet hay DataTable

Khai báo, khởi tạo và giải phóng đối tượng.

```
string ssql;
```

Khai báo đối tượng

```
Dim sqlcom As SqlCommand

Dim sqlconn As SqlConnection

Dim sqladapter As SqlDataAdapter

sqlconn.Open();
```

cách 1.

```
sqladapter = New OleDbDataAdapter(ssql, sqlconn)


sqlcom = New SqlCommand(ssql, sqlconn)
```

cách 2.

```
sqladapter = new SqlDataAdapter(sqlcom);
```

Giải phóng đối tượng

```
sqladapter.Dispose();
```

Thuộc tính:

Các thuộc tính bao gồm SelectCommand, InsertCommand, UpdateCommand, DeleteCommand: thực hiện các thao tác select, insert, update, delete dữ liệu

Phương thức:

Fill: Phương thức thực thi câu lệnh select trong sql rồi điền kết quả cho DataSet hoặc Datatable.

Update: gọi lệnh câph nhật các thay đổi vào dữ liệu lên các dữ liệu nguồn

Điền dữ liệu từ Adapter vào DataSet

Dataset là một thùng chứa dữ liệu không kết nối

```
public static DataSet Filldataset(string ssql)
```

```
{
    DataSet dataset = new DataSet();
    opendata();
    try
    {
        sqladapter = new SqlDataAdapter(ssql, sqlconn);
        sqladapter.Fill(dataset);
        sqladapter.Dispose();
    }
    catch (Exception exp)
    {
        closedata();
        System.Web.HttpContext.Current.Response.Write(exp.ToString())
;
    }
    closedata();
    return dataset;
}
```

Điền dữ liệu vào DataTable

```
public static DataTable FillDatatable(string ssql)
{
    opendata();
    DataTable datatable = new DataTable();
    try
    {
```

```
            sqladapter = new SqlDataAdapter(ssql, sqlconn);
            sqladapter.Fill(datatable);
            sqladapter.Dispose();
        }
        finally
        {
            closedata();
        }
        closedata();
        return datatable;
    }
```

VI. Đối tượng Dataset và DataTable

Là thành phần chính của kiến trúc không kết nối cơ sở dữ liệu, được dùng để nắm giữ dữ liệu của mọi cơ sở dữ liệu và cho phép thay đổi dữ liệu bên trong đối tượng này để sau đó cập nhật trở lại cơ sở dữ liệu nguồn bằng phương thức Update của đối tượng DataAdapter

Khởi tạo

```
DataSet dataset = new DataSet();
DataSet dataset = new DataSet("Mydataset");
```

Thuộc tính Tables, dataset được dùng để chứa danh sách các đối tượng DataTable

Ví dụ:

```
private void button1_Click(object sender, EventArgs e)
{
    string strQuery = "select * from tblEmployees";
```

```
            DataSet dataSet = new DataSet("Employees");
            try
            {
                SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(strQuery, Connection.sqlConnection);
                sqlDataAdapter.Fill(dataSet);

                sqlDataAdapter.Dispose();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message);
            }
            this.dataGridView1.DataSource = dataSet.Tables[0];

            //lay ve ten cua doi tuong dataset
            label1.Text ="DataSetName: " + dataSet.DataSetName ;
        }
        private void button2_Click(object sender, EventArgs e)
        {

//khai bao phat bieu sql 1
            string strQuery = "select * from tblEmployees";
            DataSet dataSet = new DataSet("Employees");
            try
```

```
            {
                SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(strQuery, Connection.sqlConnection);
                sqlDataAdapter.Fill(dataSet);

//khai bao phat bieu sql 2
                strQuery = "select * from tblContracts";
                sqlDataAdapter = new
                    SqlDataAdapter(strQuery, Connection.sqlConnection);
                DataTable dataTable = new DataTable();
                sqlDataAdapter.Fill(dataTable);
                dataSet.Tables.Add(dataTable);

                sqlDataAdapter.Dispose();
                string dataTableName="";
                foreach(DataTable dt in dataSet.Tables)
                {
                    dataTableName += dt.TableName + " ";
                }
                label1.Text = "Number of tables: " + dataTableName ;
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message);
            }
```

```
        this.dataGridView1.DataSource = dataSet.Tables[1] ;

    }
```

Phuong thuc Add, Remove

DataSet dataset=new DataSet();

DataTable datatable=new DataTable("datatablename");

dataset.Tables.Add(datatable);

dataset.Tables.Remove(datatable);

xoa voi datatable duoc dat tên

dataset.Tables.Remove(datatablename);

dataset.Tables.RemoveAt(0);

phương thức Clear loại bỏ tất cả các đối tượng trong DataTable

dataset.Tables.Clear();

De dem so dong du lieu trong bang ta co the thuc hien

int sodong=dataset.Tables[0].Rows.Count;

2. Đối tượng DataTable

```
    private void button1_Click(object sender, EventArgs e)
    {
        string strQuery = "select top 10 * from tblEmployees";
//khoi tao doi tuong DataTable
        dataTable = new DataTable("Employees");
```

```
            try
            {
                SqlDataAdapter sqlDataAdapter = new
                    SqlDataAdapter(strQuery, Connection.sqlConnection);
//dien du lieu vao datatable
                sqlDataAdapter.Fill(dataTable);



                sqlDataAdapter.Dispose();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: " + ex.Message);
            }
//gan du lieu va dataGrid voi thuoc tinh DataSource
            this.dataGridView1.DataSource = dataTable;
            label1.Text= dataTable.TableName ;
        }



// thuoc tinh DataRow tra ve cac mau tin dang chua trong doi tuong DataTable
        private void button2_Click(object sender, EventArgs e)
        {
            if (dataTable != null)
            {
                string name = "";
```

```
            foreach (DataRow dataRow in dataTable.Rows)
            {
                name += Convert.ToString(dataRow[1]) + "\n";
            }
            label1.Text = name;
        }
    }

//thuoc tinh Columns tra ve tap doi tuong DataColumn bao gom danh sach cot du
lieu cua bang chua trong doi tuong DataTable
    private void button3_Click(object sender, EventArgs e)
    {
        if (dataTable != null)
        {
            string name = "";
            foreach (DataColumn dataColumn in dataTable.Columns)
            {
                name += Convert.ToString(dataColumn.ColumnName) + "\n";
            }
            label1.Text = name;
        }
    }
```

Ví dụ chúng ta sẽ ứng dụng 3 dối tượng trên vào việc, cập nhật và hiển thị dữ liệu cho bảng sản phẩm

Bước 1: tạo bảng cơ sở dữ liệu

Ví dụ chúng ta có một bảng dữ liệu tblIntrodure gồm các trường:

pkIntrodureID	(int)
sTitle	(nvarchar(300)
sSummary	(nText)
iContent	(nText)
iPosition	(int)

Bước 2: tạo thủ tục StoreProcedure

ta tạo ra 3 thủ tục sql cho bảng giới thiệu của ta như sau

spIntrodure_insert - Thủ tục thêm mới dữ liệu

```
Create PROCEDURE spIntrodure_insert
      @sTitle nvarchar(100),
      @sSummary ntext,
      @sContent ntext,
      @iPosition int
AS
      insert into tblIntrodure(sTitle, sSummary, sContent, iPosition)
            values(@sTitle, @sSummary, @sContent, @iPosition)
      GO
```

spIntrodure_edit - Thủ tục sửa dữ liệu

```
Create PROCEDURE spIntrodure_edit
      @pkIntrodureID int,
      @sTitle nvarchar(100),
      @sSummary ntext,
```

```
	@sContent ntext,

	@iPosition int

AS

	update tblIntrodure set

		sTitle=@sTitle, sSummary=@sSummary, sContent=@sContent,
iPosition=@iPosition

	where pkIntrodureID=@pkIntrodureID

	GO
```

spIntrodure_deletebyID - Thủ tục xoá dữ liệu

```
Create PROCEDURE spIntrodure_deletebyID

	@pkIntrodureID int

AS

	delete from tblIntrodure where pkIntrodureID=@pkIntrodureID

	GO
```

Chú ý: trên là cách tạo 3 thủ tục theo cú pháp của MSSQL nếu bạn tạo thủ tục SQL trong VS thì từ khoá Create sẽ chuyển thành Alter và GO chuyển thành Return

Bước 3: Tạo các lớp(nằm trong thư mục App_Code)

IntrodureInfo.cs

```
using System;


namespace iTechPro.Modules.Introdure

{

    public class IntrodureInfo

    {
```

```
int _pkIntrodureID;
public int pkIntrodureID
{
    get { return _pkIntrodureID; }
    set { _pkIntrodureID = value; }
}

string _sTitle;
public string sTitle
{
    get { return _sTitle; }
    set { _sTitle = value; }
}

string _sImage;
public string sImage
{
    get { return _sImage; }
    set { _sImage = value; }
}

string _sSumary;
public string sSumary
{
    get { return _sSumary; }
```

```
            set { _sSumary = value; }
        }

        string _sComment;
        public string sComment
        {
            get { return _sComment; }
            set { _sComment = value; }
        }

        int _iPosition;
        public int iPosition
        {
            get { return _iPosition; }
            set { _iPosition = value; }
        }
    }
}
```

IntrodureDB.cs (chứa tất cả phương thức xử lý và lấy dữ liệu cho bảng tblIntrodure)

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

using Website.Library;
```

```
namespace Website.Modules.Introdure
{
    public class IntrodureDB : ExcuteDataHelper
    {

        public IntrodureDB()
        {
            //
            // TODO: Add constructor logic here
            //
        }

        public static void Delete(string _pkIntrodureID)
        {
            string[] parameters = new string[] { "@pkIntrodureID"};
            string[] values = new string[] { _pkIntrodureID};
            executeData("spIntrodure_deletebyID", parameters, values);

        }

        public static void Insert(IntrodureInfo _introdure)
        {
            string[] parameters = new string[7] { "@sTitle", "@sImage",
"@sSumary", "@sComment", "@sPage", "@sLang", "@iPosition" };
```

```
            string[] values = new string[7] { _introdure.sTitle,
_introdure.sImage, _introdure.sSumary, _introdure.sComment, _introdure.sPage,
_introdure.sLang, _introdure.iPosition.ToString() };

            executeData("spIntrodure_insert", parameters, values);

        }



        public static void Update(IntrodureInfo _introdure)

        {

            string[] parameters = new string[7] { "@pkIntrodureID"
,"@sTitle", "@sImage", "@sSumary", "@sComment", "@sPage", "@iPosition" };

            string[] values = new string[7] {
_introdure.pkIntrodureID.ToString(), _introdure.sTitle, _introdure.sImage,
_introdure.sSumary, _introdure.sComment, _introdure.sPage,
_introdure.iPosition.ToString() };

            executeData("spIntrodure_edit", parameters, values);

        }



        public static void UpdateIndex(string _pkIntrodureID, string _giatri)

        {

            string ssql = "update tblIntrodure set iPosition=" + _giatri + "
where pkIntrodureID=" + _pkIntrodureID;

            executeData(ssql);

        }



        public static IntrodureInfo Getinfo(string _pkIntrodureID)

        {

            DataTable mydata =
iTechProData.FillDatatable("spIntrodure_selectbyID", "@pkIntrodureID",
_pkIntrodureID);

            IntrodureInfo _introdure = new IntrodureInfo(); ;

            _introdure.sTitle = mydata.Rows[0]["sTitle"].ToString();
```

```
            _introdure.sImage = mydata.Rows[0]["sImage"].ToString();

            _introdure.sSumary = mydata.Rows[0]["sSumary"].ToString();

            _introdure.sComment = mydata.Rows[0]["sComment"].ToString();

            _introdure.sPage = mydata.Rows[0]["sPage"].ToString();

            _introdure.sLang = mydata.Rows[0]["sLang"].ToString();

            _introdure.iPosition =
int.Parse(mydata.Rows[0]["iPosition"].ToString());

            return _introdure;

        }

    }

}
```

Tại lớp IntrodureDB này chúng ta sẽ kế thừa các phương thức thực thi dữ liệu từ lớp ExcuteDataHelper.cs

Lớp ExcuteDataHelper.cs

```
using System;

using System.Data;

using System.Data.SqlClient;




namespace Website.Library

{

    public class ExcuteDataHelper : iTechProData

    {

        //phuong thuc thuc thi du lieu(them moi, chinh sua, xoa) khi dua vao
mot tham so sql

        #region executeData(string sql)"Thực thi dữ liệu"

        public static void executeData(string sql)
```

```
        {
            opendata();
            sqlcom = new SqlCommand(sql, sqlconn);
            try
            {
                sqlcom.ExecuteNonQuery();
                closedata();
            }
            catch (Exception exp)
            {
                closedata();
                HttpContext.Current.Response.Write(sql + "<br/>");
                HttpContext.Current.Response.Write("Có lỗi trong quá trình
thực thi " + exp.ToString());
            }
        }
        #endregion

        //phuong thuc thuc thi du lieu voi tham so dua vao
        #region executeData(string store, string[] Parameter, string[]
Values)
        public static void executeData(string store, string[] Parameter,
string[] Values)
        {
            opendata();
            sqlcom = new SqlCommand();
            sqlcom.CommandText = store;
```

```
            sqlcom.Connection = sqlconn;
            sqlcom.CommandType = CommandType.StoredProcedure;
            for (int i = 0; i < Parameter.Length; i++)
            {
                sqlcom.Parameters.AddWithValue(Parameter[i], Values[i]);
            }
            try
            {
                sqlcom.ExecuteNonQuery();
                closedata();
            }
            catch (DataException exp)
            {
                sqlconn.Close();
                HttpContext.Current.Response.Write(exp.ToString());
            }
        }
        #endregion
    }
}
```

Trong lớp này chúng ta có 2 phương thức thực thi dữ liệu có thể là thêm mới, chỉnh sửa hay xoá dữ liệu void executeData(string sql) cho phép bạn thực thi dữ liệu với một chuỗi sql đưa vào còn executeData(string store, string[] Parameter, string[] Values) sẽ thực thi dữ liệu với hàm thủ tục từ SQL truyền vào với hai mảng giá trị và tham số và lơp này này thừa kế từ lớp dẫn xuất iTechProData.cs

Lớp iTechProData.cs

```
using System;
```

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


namespace Website.Library
{
    public class WebsiteData
    {
        #region khai bao bien

        protected string ssql;
        protected static SqlConnection sqlconn;
        protected static SqlCommand sqlcom;
        protected static SqlDataAdapter sqladapter;
        protected static DataSet mydata;
        protected static SqlDataReader sqlreader;

        #endregion

        //phuong thuc mo du lieu
        #region opendata() "Mở dữ liệu"
        public static void opendata()
        {
            //đọc chuỗi kết nối từ trong file web.config
```

```
            System.Configuration.AppSettingsReader settingsReader = new
AppSettingsReader();

            string driver = (string)settingsReader.GetValue("hcubiudata",
typeof(String));



            try

            {

                sqlconn = new SqlConnection(driver);

                if (sqlconn.State != ConnectionState.Open)

                {

                    sqlconn.Open();

                }

            }

            catch (Exception exp)

            {

                HttpContext.Current.Response.Write("Lỗi mở dữ liệu" +
exp.ToString());

            }

        }

        #endregion



        //phuong thuc dong du lieu

        #region closedata() "Đóng dữ liệu"

        public static void closedata()

        {

            if (sqlconn.State != ConnectionState.Closed)

            {
```

```
                sqlconn.Close();
                sqlconn.Dispose();
            }
        }
        #endregion

        // điền dữ liệu vào DataTable từ một thủ tục trong Database
        public static DataTable FillDatatable(string store,string _thamso,
string _giatri)
        {
            opendata();
            DataTable datatable = new DataTable();
            sqlcom = new SqlCommand();
            sqlcom.CommandText = store;
            sqlcom.Connection = sqlconn;
            sqlcom.Parameters.AddWithValue(_thamso, _giatri);
            sqlcom.CommandType = CommandType.StoredProcedure;
            try
            {
                sqladapter = new SqlDataAdapter(sqlcom);
                sqladapter.Fill(datatable);
                sqladapter.Dispose();
            }
            finally
            {
                closedata();
```

```
            }

            return datatable;

        }

    }



}
```

Trong lớp trên bạn thấy có 2 đối tượng data mới đó là DataAdapter và DataTable chúng ta sẽ học kỹ hơn trong phần sau trong ví dụ này các bạn chỉ cần hiểu qua là DataAdapter là bộ đọc dữ liệu từ nguồn dữ liệu, và DataTable là đối tượng lưu trữ dữ liệu không kết nối, nó như một bảng tạm để chứa dữ liệu và nó ko cần biết dữ liệu đó từ nguồn nào.

Bước 4: Tạo giao diện sử dụng

Code: adminIntrodure.aspx

```
<%@ Page Language="C#" MasterPageFile="~/admin.master" AutoEventWireup="true"
CodeFile="adminIntrodure.aspx.cs" Inherits="Desktop_Introdure_adminIntrodure"
Title="Admin - Introdure" %>



<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<!--Trinh bay du lieu-->

 <table cellpadding="0" cellspacing="0" width="100%" style="padding-
right:3px; height:390px">



  <tr>

   <td style="padding:15px 15px 15px 15px" valign="top">



<table width="100%" cellpadding="0" cellspacing="0">

    <tr>

     <td align="left" class="hcubiufontlarger">Giới thiệu</td>
```

```
    </tr>

    <tr><td style="height:15px;"></td></tr>




   <!--start them moi-->

   <tr>

    <td align="left">

        <asp:Panel ID="panelupdate" Width="100%" runat="Server"
Visible="false">

            <table width="100%" style="padding-left:20px;">

             <tr>

              <td colspan="2"><b>Cập nhật thông tin giới thiệu</b></td>

             </tr>

             <tr><td style="width: 78px; height:15px;"></td></tr>

             <tr>

                <td align="left" style="width: 78px">Tiêu đề</td>

                <td align="left"><input type="text" name="txtTitle"
id="txtTitle" runat="server" style="width: 329px" /></td>

             </tr>

             <tr>

                <td valign="middle" align="left" style="width: 78px">Tóm
tắt</td>

                <td align="left">



                    <asp:TextBox ID="txtTomtat" runat="server"
TextMode="MultiLine"></asp:TextBox>



                </td>
```

```
            </tr>

            <tr>

               <td align="left" style="height: 88px; width: 78px;">Nội
dung</td>

               <td align="left" style="height: 88px">



                 <asp:TextBox ID="txtNoidung" runat="server"
TextMode="MultiLine" />



               </td>

            </tr>

            <tr>

             <td align="left">Vị trí</td>

             <td align="left">

               <asp:TextBox ID="txtvitri" runat="server"
Text="1"></asp:TextBox>

                 <asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="txtvitri"

                     ErrorMessage="Vị trí phải là kiểu số"
MaximumValue="100" MinimumValue="0" Type="Integer"></asp:RangeValidator></td>

            </tr>

            <tr><td style="width: 78px; height:15px;"></td></tr>

            <tr>

             <td colspan="2" align="left">

              <asp:Button ID="btnaccept" runat="server" Text="Ghi"
Width="100px" OnClick="btnaccept_Click"/>

              <asp:Button ID="btcancel" runat="server" Text="Bỏ qua"
Width="100px" OnClick="btcancel_Click" />

                 <asp:Label ID="lblidintro" runat="server" Text=""
Visible="false"></asp:Label></td>
```

```
            </tr>
          </table>
         </asp:Panel>

<!--End them moi-->

    </td>
   </tr>

   <tr>
    <td  style="height:5px;"></td>
   </tr>

   <asp:Panel ID="panelview" runat="server">

   <tr>
    <td align="left" style="padding-bottom:3px;"><asp:LinkButton
ID="btnaddnew" CssClass="linkbutton" runat="server" Text="Thêm mới"
OnClick="btnaddnew_Click" /></td>
   </tr>

   <tr>
    <td valign="top" align="left">
    <asp:DataGrid id="gridintro" runat="server"
        BorderColor="black"
        Width="100%"
        BorderWidth="1"
```

```
        CellPadding="3"

        Font-Size="10pt"

        HeaderStyle-BackColor="#aaaadd"

        OnItemCommand="gridintro_OnItemCommand"

        AutoGenerateColumns="false">

        <HeaderStyle BackColor="#AAAADD"></HeaderStyle>

        <Columns>

          <asp:TemplateColumn HeaderStyle-HorizontalAlign="Center" ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80px" HeaderText="STT">

            <ItemTemplate>

              <%#Container.ItemIndex +1 %>

             </ItemTemplate>

          </asp:TemplateColumn>

          <asp:BoundColumn HeaderStyle-HorizontalAlign="Left" ItemStyle-
HorizontalAlign="Left" DataField="sTitle" ReadOnly="true" HeaderText="Tiêu
đề"></asp:BoundColumn>



          <asp:TemplateColumn HeaderText="Vị trí" ItemStyle-
HorizontalAlign="Center" HeaderStyle-HorizontalAlign="Center" HeaderStyle-
Width="100px" ItemStyle-Width="100px" ItemStyle-Height="24px" >

           <ItemTemplate>

            <asp:TextBox ID="txtVitri" Width="39px" runat="server"
Text='<%#Eval("iPosition") %>' />

           </ItemTemplate>

          </asp:TemplateColumn>



          <asp:TemplateColumn HeaderText="Chinh sửa" ItemStyle-
HorizontalAlign="Center" HeaderStyle-Width="80px" HeaderStyle-
HorizontalAlign="Center" ItemStyle-Width="100px" ItemStyle-Height="24px" >

           <ItemTemplate>
```

```
            <asp:LinkButton ID ="Edit" CommandArgument
='<%#DataBinder.Eval(Container,"DataItem.pkIntrodureID")%>' runat ="server"
CommandName="Edit" Text ="Edit"></asp:LinkButton>

           </ItemTemplate>

          </asp:TemplateColumn>



          <asp:TemplateColumn HeaderText="Xóa" HeaderStyle-
HorizontalAlign="Center" HeaderStyle-Width="80px" ItemStyle-
HorizontalAlign="Center" ItemStyle-Width="100px" ItemStyle-Height="24px" >

           <ItemTemplate>

            <asp:LinkButton ID ="Delete" CommandArgument
='<%#DataBinder.Eval(Container,"DataItem.pkIntrodureID")%>' runat ="server"
CommandName="Delete" Text ="Delete"></asp:LinkButton>

           </ItemTemplate>

          </asp:TemplateColumn>



         </Columns>

     </asp:DataGrid>

    </td>

  </tr>

  <tr>

   <td align="right" style="padding-top:3px;">

    <asp:Label ID="lblthongbao" runat="server"></asp:Label>

    <asp:LinkButton ID="lbncapnhatvitri" CssClass="linkbutton" runat="server"
Text="Cập nhật vị trí" OnClick="lbncapnhatvitri_Click" />

   </td>

  </tr>



  </asp:Panel>

 </table>
```

```
   </td>
  </tr>
  <tr><td style="height:30px;"></td></tr>
 </table>

</asp:Content>
```

Code adminIntrodure.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using Website.Library;
using Website.Modules.Introdure;

public partial class Desktop_Introdure_adminIntrodure : System.Web.UI.Page
{
    string ssql;
    void Loaddatagrid()
```

```
    {
        ssql = "select pkIntrodureID,sTitle,iPosition from tblIntrodure";
        DatagridHelper.fill_datagrid(gridintro, ssql, "pkIntrodureID");
        foreach (DataGridItem item in this.gridintro.Items)
        {
            LinkButton lbn =
(LinkButton)this.gridintro.Items[item.ItemIndex].FindControl("Delete");
            lbn.Attributes.Add("onclick", "javascript:return confirm('Bạn có
chắc chắn xoá mục giới thiệu này')");
        }
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Loaddatagrid();
        }
    }

    private IntrodureInfo Getcontent()
    {
        IntrodureInfo intro = new IntrodureInfo();
        try
        {
            intro.pkIntrodureID = int.Parse(lblidintro.Text);
        }
```

```
        catch
        {
        }
        intro.sTitle = txtTitle.Value;
        intro.sSumary = txtTomtat.Text;
        intro.sContent = txtNoidung.Text;
        intro.iPosition = int.Parse(txtvitri.Text);
        return intro;
    }

    protected void btnaddnew_Click(object sender, EventArgs e)
    {
        panelupdate.Visible = true;
        panelview.Visible = false;
        txtNoidung.Text = "";
        txtTitle.Value = "";
        this.txtTomtat.Text = "";
        txtvitri.Text = "1";
        btnaccept.Text = "Ghi";
    }

    protected void gridintro_OnItemCommand(object sender,
DataGridCommandEventArgs e)
    {
        lblidintro.Text = e.CommandArgument.ToString();
        if (e.CommandName == "Edit")
```

```
        {
            IntrodureInfo introdure = IntrodureDB.Getinfo(lblidintro.Text);
            txtTitle.Value = introdure.sTitle;
            txtTomtat.Text = introdure.sSumary;
            txtvitri.Text = introdure.iPosition.ToString();
            txtNoidung.Text = introdure.sContent;
            btnaccept.Text = "Cập nhật";
            panelupdate.Visible = true;
            panelview.Visible = false;
        }
        else
        {
            IntrodureDB.Delete(lblidintro.Text);
            Loaddatagrid();
        }
    }

    protected void btnaccept_Click(object sender, EventArgs e)
    {
        IntrodureInfo introdure = Getcontent();
        if (btnaccept.Text == "Ghi")
        {
            IntrodureDB.Insert(introdure);
        }
        else
        {
```

```
            IntrodureDB.Update(introdure);
        }
        panelupdate.Visible = false;
        panelview.Visible = true;
        Loaddatagrid();
       }

    protected void btcancel_Click(object sender, EventArgs e)
    {
        panelview.Visible = true;
        panelupdate.Visible = false;
        Loaddatagrid();
    }

    protected void lbncapnhatvitri_Click(object sender, EventArgs e)
    {
        foreach (DataGridItem item in gridintro.Items)
        {
            TextBox txt =
(TextBox)this.gridintro.Items[item.ItemIndex].FindControl("txtVitri");
            IntrodureDB.UpdateIndex(gridintro.DataKeys[item.ItemIndex].ToStri
ng(), txt.Text);
        }
    }
}
```

Trong đoạn mã trên có sử dụng DataGrid bạn sẽ được học nó kỹ hơn trong phần sau, bây giờ bạn cứ coi nó như là một công cụ để hiển thị dữ liệu.