

PHÁT TRIỂN ỨNG DỤNG CSDL 1

Tháng 10/2015

LẬP TRÌNH TRUY XUẤT CƠ SỞ DỮ LIỆU VỚI ADO.NET VÀ C#

Tóm tắt nội dung bài thực hành:

Xây dựng một ứng dụng (c#)
console kết nối CSDL

1. MỤC TIÊU

- Xây dựng ứng dụng kết nối CSDL cơ bản với các chức năng:
 - o Kết nối CSDL
 - o Ngắt kết nối CSDL
 - o Thêm dữ liệu
 - o Đọc dữ liệu
 - o Xóa dữ liệu
 - o Cập nhật dữ liệu
 - o Tìm kiếm dữ liệu

2. HƯỚNG DẪN CỤ THỂ

- Thư viện sử dụng:

- o `using System.Data;`
- o `using System.Data.SqlClient;`

- Chuỗi kết nối: là chuỗi thiết lập cấu hình kết nối CSDL

- o Sử dụng tài khoản window

```
static string ConnectionString = @"Server=myServerAddress; Database=myDataBase;  
Trusted_Connection=True;";
```

- o Sử dụng tài khoản SQL Server

```
static string ConnectionString = @"Server=myServerAddress; Database=myDataBase;  
User Id=myUsername; Password=myPassword;";
```

- o Tham khảo: <http://www.connectionstrings.com/sql-server/>

- Kết nối CSDL: là phương thức để mở kết nối CSDL

```
SqlConnection Connection { get; set; }
```

```
/// <summary>  
/// Start: Open connect  
/// </summary>  
public void Connect()  
{  
    try  
    {  
        if(Connection == null)  
            Connection = new SqlConnection(ConnectionString);  
  
        if(Connection.State != ConnectionState.Closed)  
            Connection.Close();  
  
        Connection.Open();  
    }  
    catch (SqlException ex)  
    {  
        throw ex;  
    }  
}
```

- Ngắt kết nối CSDL: là phương thức để ngắt kết nối CSDL

```

/// <summary>
/// End: Close connect
/// </summary>
public void Disconnect()
{
    if(Connection!=null && Connection.State == ConnectionState.Open)
        Connection.Close();
}

```

- **Thêm, Xóa, Sửa dữ liệu:** là phương thức thực hiện các câu lệnh Insert, Delete, Update

```

/// <summary>
/// Insert, Update, Delete
/// </summary>
/// <returns>number row effects</returns>
public int ExecuteNonQuery(CommandType cmdType, string strSql)
{
    try
    {
        SqlCommand command = Connection.CreateCommand();
        command.CommandText = strSql;
        command.CommandType = cmdType;

        int nRow = command.ExecuteNonQuery();
        return nRow;
    }
    catch (SqlException ex)
    {
        throw ex;
    }
}

```

Không dùng
tham số

```

/// <summary>
/// Insert, Update, Delete
/// </summary>
/// <returns>number row effects</returns>
public int ExecuteNonQuery(CommandType cmdType, string strSql,
                           params SqlParameter [] parameters)
{
    try
    {
        SqlCommand command = Connection.CreateCommand();
        command.CommandText = strSql;
        command.CommandType = cmdType;
        if (parameters!= null && parameters.Length > 0)
            command.Parameters.AddRange(parameters);

        int nRow = command.ExecuteNonQuery();
        return nRow;
    }
    catch (SqlException ex)
    {
        throw ex;
    }
}

```

Dùng tham
số

- **Tìm kiếm dữ liệu**: là phương thức trả ra kết quả của câu lệnh Select

Cách 1:

Bước 1: Tạo đối tượng kết nối đến CSDL & mở kết nối

Bước 2: Xây dựng câu truy vấn SQL để thực hiện thao tác mong muốn (Thêm, xóa, sửa, cập nhật)

Bước 3: Tạo đối tượng command để thực thi câu truy vấn

Bước 4: Thực thi câu truy vấn và xử lý kết quả trả về

Bước 5: Đóng kết nối

```
/// <summary>
/// select
/// </summary>
/// <returns>sqlDataReader</returns>
public SqlDataReader GetReader(CommandType cmdType, string strSql)
{
    try
    {
        SqlCommand command = Connection.CreateCommand();
        command.CommandText = strSql;
        command.CommandType= cmdType;

        return command.ExecuteReader();
    }
    catch (SqlException ex)
    {
        throw ex;
    }
}
```

Cách 2:

Bước 1: Tạo đối tượng kết nối đến CSDL

Bước 2: Chuẩn bị câu truy vấn

Bước 3: Tạo đối tượng SqlDataAdapter và tạo đối tượng DataTable để chứa dữ liệu

Bước 4: Thực hiện Fill dữ liệu lên DataTable

Bước 5: Sử dụng dữ liệu ở DataTable để làm nguồn dữ liệu cho: ComboBox, DataGridView, ...

```

/// <summary>
/// select
/// </summary>
/// <returns>DataTable</returns>
public DataTable Select(CommandType cmdType, string strSql)
{
    try
    {
        SqlCommand command = Connection.CreateCommand();
        command.CommandText = strSql;
        command.CommandType= cmdType;

        SqlDataAdapter da = new SqlDataAdapter(command);
        DataTable dt = new DataTable();
        da.Fill(dt);

        return dt;
    }
    catch (SqlException ex)
    {
        throw ex;
    }
}

```

- Demo

class Program

```
{
```

```

static void DemoReader()
{
    string strSql = "Select magd, thoigiangd From giaodich";

    Provider provider = new Provider();
    provider.Connect();

    SqlDataReader reader = provider.GetReader(CommandType.Text, strSql);

    while (reader.Read())
    {
        Console.WriteLine("Id = {0}, Name = {1}", reader[0], reader[1]);
    }

    provider.Disconnect();
    Console.Read();
}

```

```

static void DemoAdapter()
{
    string strSql = "Select magd, thoigiangd From giaodich Where magd >"+ soGiaoDich ;

    Provider provider = new Provider();
    provider.Connect();

    DataTable dt = provider.Select(CommandType.Text, strSql);
    foreach (DataRow row in dt.Rows)
    {
        Console.WriteLine("Id = {0}, = {1}", row["magd"], row["thoigiangd"]);
    }

    provider.Disconnect();
}

```

```

static void DemoInsert()
{
    string strSql = "insert giaodich values(@magd,@thoigian)" ;

    Provider provider = new Provider();
    provider.Connect();

    provider.ExecuteNonQuery(CommandType.Text, strSql,
        new SqlParameter { ParameterName = "@Magd", Value = soGiaoDich },
        new SqlParameter { ParameterName = "@thoigian", Value = thoigian });

    provider.Disconnect();
}

```

```

static void DemoUpdate()
{
    string strSql = "update giaodich set thoigiangd=@thoigian where magd=@magd";

    Provider provider = new Provider();
    provider.Connect();

    provider.ExecuteNonQuery(CommandType.Text, strSql,
        new SqlParameter { ParameterName = "@Magd", Value = soGiaoDich },
        new SqlParameter { ParameterName = "@thoigian", Value = thoigian });

    provider.Disconnect();
}

```

```

static void DemoDelete()
{
    string strSql = "delete giaodich where magd=@magd";

    Provider provider = new Provider();
    provider.Connect();

    provider.ExecuteNonQuery(CommandType.Text, strSql,
        new SqlParameter { ParameterName = "@Magd", Value = soGiaoDich });

    provider.Disconnect();
}

```

```

static void DemoGetData()
{
    string strSql = "sp_test";

    Provider provider = new Provider();
    provider.Connect();
    SqlParameter p = new SqlParameter("@j", SqlDbType.VarChar, 10);
    p.Direction = ParameterDirection.Output;

    provider.ExecuteNonQuery(CommandType.StoredProcedure, strSql,
        new SqlParameter { ParameterName = "@Magd", Value = soGiaoDich }, p);

    Console.WriteLine(p.Value);

    provider.Disconnect();
}

```

Gọi **Stored-Procedure**
& nhận giá trị **OutPut**

```
static void Main(string[] args)
{
    DemoReader();
    DemoAdapter();
}
```

```
}
```