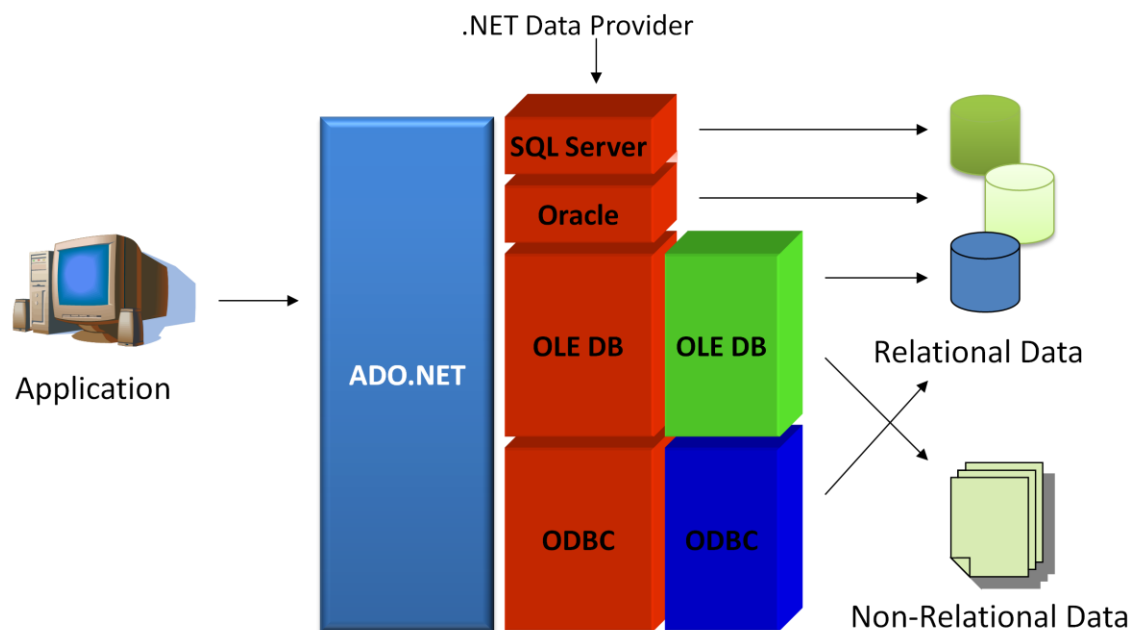


1 Chương 3 Cơ Sở Dữ Liệu với ADO.Net

1.1 Kiến trúc ADO.Net



ADO.NET là một phần của .NET Framework, nó được xem là “bộ thư viện lớp” chịu trách nhiệm xử lý dữ liệu trong ngôn ngữ MS.NET. ADO.NET được thiết kế với dạng dữ liệu “ngắt kết nối”, nghĩa là chúng ta có thể lấy cả một cấu trúc phức tạp của dữ liệu từ database, sau đó ngắt kết nối với database rồi mới thực hiện các thao tác cần thiết. Đây là một sự tiến bộ về mặt thiết kế bởi vì thiết kế ADO trước đây luôn cần duy trì một kết nối trong quá trình thao tác dữ liệu.

Có thể coi ADO.NET là một thế hệ tiếp theo của ADO. ADO.NET kế thừa tất cả những ưu điểm của ADO, đồng thời với ý tưởng thiết kế hoàn toàn mới ADO.NET có một diện mạo khác hẳn so với tiền thân của nó. Một vài đặc điểm nổi bật của ADO.NET mà ADO không có như sau:

ADO.NET được thiết kế hoàn toàn dựa vào XML vì XML là chuẩn trao đổi dữ liệu tiên bộ và tốt nhất trên môi trường Internet hiện nay. ADO.NET được thiết kế hoàn toàn hướng đối tượng : đây là đặc điểm chi phối toàn bộ các sản phẩm Microsoft .NET.

1.2 DataSet

Là thành phần chính cho đặc trưng kết nối không liên tục (ngắt kết nối) của kiến trúc ADO.NET.

DataSet được thiết kế để thích ứng với bất kì nguồn dữ liệu nào.

Nhiệm vụ của DataSet là nhận dữ liệu về từ DataAdapter và xử lý nó.

Lưu trữ dữ liệu của DataBase trong bộ nhớ.

Mọi thao tác thay đổi dữ liệu được thực hiện trên DataSet, không làm ảnh hưởng đến DataBase.

Theo vết các thay đổi trên dữ liệu và có thể cập nhật dữ liệu ngược vào DataBase thông qua SqlDataAdapter.

Gồm các đối tượng : DataTable, DataRelationship, Constraint.

Việc sử dụng DataSet là một tiến bộ lớn của kiến trúc ADO.NET tuy nhiên với các ứng dụng Web việc sử dụng DataSet không được khuyến khích vì đối tượng DataSet được xem là quá lớn, nặng nề khó thích hợp cho đường truyền trên web vốn rất hạn chế.

1.2.1 Các thuộc tính của DataSet

STT	Tên Thuộc Tính	Ý Nghĩa
1	DataSetName	Tên của dataset
2	Relations	
3	Tables	

1.2.2 Các phương thức chính của DataSet

STT	Tên Phương Thức	Ý Nghĩa
1	GetChange	
2	RejectChanges	
3	AcceptChanges	
4	GetXML	
5	ReadXML	
6	WriteXML	

1.3 SQL Server Data Provider

Các lớp chính của SQL Server Data Provider

Tên Lớp	Ý Nghĩa
SqlCommand	Thực thi SQL queries, câu lệnh hoặc lưu trữ thủ tục
SqlConnection	Tạo kết nối tới SQL Server
SqlDataAdapter	Cầu nối trung gian giữa dataset và data source
SqlReader	Cung cấp một data stream tới kết quả
SqlError	Lưu trữ thông tin về lỗi và cảnh cáo (warning)
SqlException	Các ngoại lệ trong trường hợp SQL Server lỗi và cảnh báo
SqlParameter	Tham số biên command
SqlTransaction	Transaction của SQL Server

1.3.1 Lớp đối tượng SqlConnection

Dùng để tạo kết nối đến các CSDL Sql Server

1.3.1.1 Các thuộc tính của SqlConnection

STT	Tên Thuộc Tính	Ý Nghĩa
1	ConnectionString	Chuỗi kết nối database
2	ConnectionTimeout	Thời gian chờ trước khi ngắt kết nối với database
3	Container	Icontainer chứa các Component
4	Database	Tên của database hiện tại sau khi kết nối
5	DataSource	Tên của server, tên file chứa dữ liệu
6	Provider	Tên của OLEDB provider
7	ServerVersion	Version của server
8	Site	Isite của Component
9	State	Trạng thái của liên kết

1.3.1.2 Các phương thức chính của SqlConnection

STT	Tên Phương Thức	Ý Nghĩa
1	SqlConnection	Phương thức khởi tạo
2	Open	Mở kết nối tới database
3	Close	Đóng kết nối với database
4	Dispose	Hủy đối tượng
5	BeginTransaction	Bắt đầu 1 transaction

Cách tạo một kết nối:Cách 1:

- Tạo chuỗi kết nối
- Tạo đối tượng kết nối SqlConnection, truyền tham số chuỗi kết nối vào.

Ví dụ:

```
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection(CnStr);
```

Cách 2:

- Tạo chuỗi kết nối
- Tạo đối tượng kết nối SqlConnection, không truyền tham số
- Trỏ thuộc tính ConnectionString của đối tượng SqlConnection đến chuỗi kết nối.

Ví dụ:

```
string CnStr = "Server=.;Database=AgribankDB;UID=sa;PWD=hoangvy";
SqlConnection Cn = new SqlConnection();
Cn.ConnectionString = CnStr;
Cn.Open();
Cn.Close();|
```

Các tham số của SqlConnection

Tham số	Mô tả
Data Source	Tên máy hoặc IP
Initial Catalog	Tên Database
Integrated Security	Sử dụng SSPI
User ID	Tên User kết nối
Password	Mật khẩu kết nối

Lưu ý:

- Windows Authenticate: có nghĩa là sử dụng quyền hạn của Windows Account để truy cập SQL Server. Khi kết nối ở mode này, thì không cần truyền userID, password.
- Để kết nối bằng Window Authencation, trong chuỗi kết nối cho thêm thuộc tính: "Integrated Security=true" hoặc là: "Integrated Security=SSPI"
- Ví dụ:

Data Source=localhost;Initial Catalog=myDB;Integrated Security=SSPI;

- Sql server account: truyền thông tin account (username, password) vào chuỗi kết nối
- Ví dụ:
string CnStr = "Server=.;Database=TestDB;UID=sa;PWD=sa";

1.3.2 Lớp đối tượng SqlCommand

Đối tượng thực hiện các câu lệnh tương tác truy vấn, rút trích dữ liệu từ database khi đã thiết lập kết nối tới dữ liệu và trả về kết quả.

Kết quả trả về được lưu trữ dưới dạng luồng thông qua 2 đối tượng :

- DataReader
- DataSet thông qua một đối tượng SqlDataAdapter

Các hàm khởi tạo

```
new SqlCommand()
```

```
new SqlCommand(cmdText)
```

```
new SqlCommand(cmdText, connection)
```

```
new SqlCommand(cmdText, connection, transaction)
```

Ví dụ 1:

```
String sql = "Select * from SinhVien"  
SqlCommand cmd = new SqlCommand();  
Cmd.CommandText = sql;
```

Ví dụ 2:

```
String sql = "Select * from SinhVien"  
SqlCommand cmd = new SqlCommand(sql);
```

Ví dụ 3:

```
String CnStr = "Server=.;Database='SinhVienDB';uid=sa;pwd=sa;";  
SqlConnection cn = new SqlConnection(CnStr);  
String sql = "Select * from SinhVien"  
SqlCommand cmd = new SqlCommand(sql,cn);
```

Ví dụ 4:

```
String CnStr = "Server=.;Database='SinhVienDB';uid=sa;pwd=sa;";
SqlConnection cn = new SqlConnection(CnStr);
String sql = "Select * from SinhVien"
SqlCommand cmd = new SqlCommand(sql,cn,null);
```

1.3.2.1 Các thuộc tính của SqlCommand

STT	Tên Thuộc Tính	Ý Nghĩa
1	CommandText	Câu lệnh SQL hay stored procedure kết nối data source
2	CommandTimeout	Thời gian chờ trước khi ngắt kết nối
3	CommandType	Giá trị mô tả hoạt động của CommandText
4	Connection	Thiết lập SqlConnection
5	Container	IContainer chứa Component
6	DesignTimeVisible	Giá trị mô tả đối tượng command xuất hiện trong Designer
7	Parameters	Lấy các tham số
8	Site	ISite của Component
9	Transaction	SqlTransaction khi SqlCommand thực thi
10	UpdatedRowSource	Update DataRow

1.3.2.2 Các phương thức chính của SqlCommand

STT	Tên Phương Thức	Ý Nghĩa
1	SqlCommand	Phương thức khởi tạo
2	ExecuteReader	Gửi CommandText đến Kết nối để tạo ra SqlDataReader
3	ExecuteNonQuery	Trả về số lượng dòng bị ảnh hưởng trên CSDL
4	ExecuteScalar	Trả về 1 giá trị đầu tiên (VD: giá trị tính tổng)
5	ExecuteXMLReader	Trả về 1 XMLReader
6	Dispose	Hủy đối tượng

1.3.3 Lớp đối tượng SqlParameter

Cần thực hiện câu lệnh nhiều lần với các giá trị khác nhau.

Đối tượng tham số truyền vào cho đối tượng SqlCommand

Có các thuộc tính sau :

STT	Tên Thuộc Tính	Ý Nghĩa
1	ParameterName	Tên tham số
2	SqlDbType	Kiểu dữ liệu của tham số tương ứng với kiểu dữ liệu của SqlServer
3	Direction	Input, Output, InputOutput, ReturnValue, ...
4	Size	Kích thước tối đa của dữ liệu
5	Value	Giá trị của tham số (input / Output)

Để truyền tham số cho command:

➤ Khai báo đối tượng command với tham số:

```
SqlCommand cmd = new SqlCommand ( "select * from Sach where SachID =  
@SachID ", conn);
```

➤ Định nghĩa các tham số được dùng ở đối tượng command:

```
SqlParameter param = new SqlParameter();  
param.ParameterName = "@SachID ";  
param.Value = sachID;
```

➤ Thêm mới tham số vào đối tượng command:

```
cmd.Parameters.Add(param);
```

Ví dụ: truyền nhiều tham số

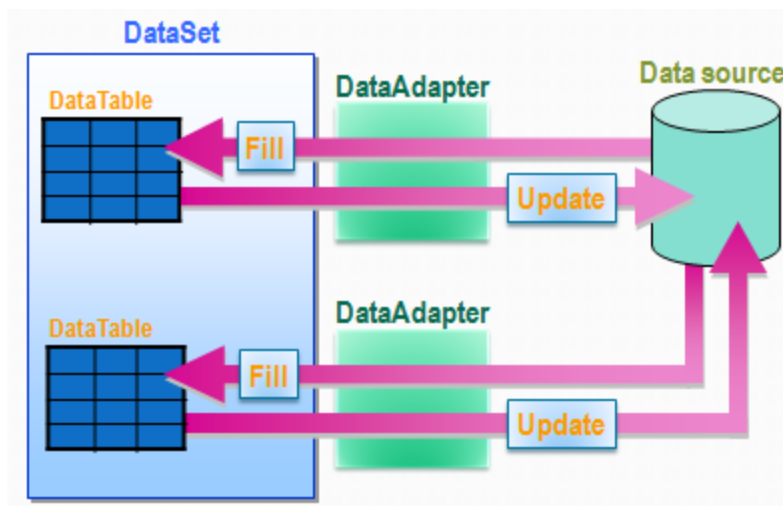
```
String CnStr = "Server=localhost;Database=DBTest;UID=sa;PWD=sa";  
SqlConnection = new SqlConnection(CnStr);  
String cmdText = "INSERT INTO Student(Name,Address, Age) VALUES(@Name, @Address, @Age)";  
SqlCommand cmd = new SqlCommand(cmdText,Cn);  
  
cmd.Parameters.Add("@Name", SqlDbType.NVarChar);  
cmd.Parameters.Add("@Address", SqlDbType.NVarChar);  
cmd.Parameters.Add("@Age", SqlDbType.Int)  
  
cmd.Parameters["@Name"].Value = txtName.Text;  
cmd.Parameters["@Address"].Value = txtAddress.Text;  
cmd.Parameters["@Age"].Value = Convert.ToInt32(txtAge.Text);  
|  
Cn.Open();  
- //Thực thi câu lệnh và đóng kết nối.  
cmd.ExecuteNonQuery();  
Cn.Close();
```

1.3.4 Lớp đối tượng SqlDataAdapter

Đây là đối tượng rất quan trọng của ADO.NET, nó là cầu nối trung gian của database và dataset (dataset là đối tượng ngắt kết nối), bởi vì đối tượng “ngắt kết nối” dataset không thể liên lạc trực tiếp với database nên nó cần một đối tượng trung gian lấy dữ liệu từ database cho nó

Cung cấp các phương thức và thuộc tính để lấy và lưu dữ liệu giữa DataSet và CSDL

Sử dụng DataSet để lưu trữ dữ liệu, đồng thời, cho cập nhật dữ liệu ngược lại vào Database



Các phương thức chính :

STT	Tên Phương Thức	Ý Nghĩa
1	Fill	Lấy dữ liệu từ data source
2	Update	Cập nhật dữ liệu vào data source

1.4 OleDb Data Provider

Các lớp chính trong OleDb Data Provider

Tên Lớp	Ý Nghĩa
OleDbCommand	Thực thi SQL queries, câu lệnh hoặc lưu trữ thủ tục
OleDbConnection	Tạo kết nối tới data source
OleDbDataAdapter	Cầu nối trung gian giữa dataset và data source
OleDbError	Lưu trữ thông tin về lỗi và cảnh cáo (warning)
OleDbException	Các ngoại lệ trong trường hợp SQL Server lỗi và cảnh báo
OleDbParameter	Tham số biên command
OleDbTransaction	Transaction của data source

1.4.1 Lớp đối tượng OleDbConnection

Lớp đối tượng OleDbConnection là một lớp đối tượng dùng để kết nối cơ sở dữ liệu.

1.4.1.1 Các thuộc tính của OleDbConnection

STT	Tên Thuộc Tính	Ý Nghĩa
1	ConnectionString	Chuỗi kết nối database
2	ConnectionTimeout	Thời gian chờ trước khi ngắt kết nối với database
3	Container	Icontainer chứa các Component
4	Database	Tên của database hiện tại sau khi kết nối
5	DataSource	Tên của server, tên file chứa dữ liệu
6	Provider	Tên của OLEDB provider
7	ServerVersion	Version của server
8	Site	Isite của Component
9	State	Trạng thái của liên kết

1.4.1.2 Các phương thức chính của OleDbConnection

STT	Tên Phương Thức	Ý Nghĩa
1	OleDbConnection	Phương thức khởi tạo
2	Open	Mở kết nối tới database
3	Close	Đóng kết nối với database
4	Dispose	Hủy đối tượng

1.4.2 Lớp đối tượng OleDbCommand

Lớp đối tượng OleDbCommand thực hiện các câu truy vấn SQL hoặc stored procedure trên một data source.

1.4.2.1 Các thuộc tính của OleDbCommand

STT	Tên Thuộc Tính	Ý Nghĩa
1	CommandText	Câu lệnh SQL hay stored procedure kết nối data source
2	CommandTimeout	Thời gian chờ trước khi ngắt kết nối

3	CommandType	Giá trị mô tả hoạt động của CommandText
4	Connection	Thiết lập OleDbConnection
5	Container	IContainer chứa Component
6	DesignTimeVisible	Giá trị mô tả đối tượng command xuất hiện trong Designer
7	Parameters	Lấy các tham số
8	Site	ISite của Component
9	Transaction	OleDbTransaction khi OleDbCommand thực thi
10	UpdatedRowSource	Update DataRow

1.4.2.2 Các phương thức của OleDbCommand

STT	Tên Phương Thức	Ý Nghĩa
1	OleDbCommand	Phương thức khởi tạo
2	ExecuteReader	Gửi CommandText đến Kết nối để tạo ra SqlDataReader
3	ExecuteNonQuery	Trả về số lượng dòng bị ảnh hưởng trên CSDL
4	ExecuteScalar	Trả về 1 giá trị đầu tiên (VD: giá trị tính tổng)
3	Dispose	Hủy đối tượng

1.4.3 Lớp đối tượng OleDbParameter

STT	Tên Thuộc Tính	Ý Nghĩa
1	ParameterName	Tên tham số
2	OleDbType	Kiểu dữ liệu của tham số tương ứng với kiểu dữ liệu của data source
3	Direction	Input, Output, InputOutput, ReturnValue, ...
4	Size	Kích thước tối đa của dữ liệu
5	Value	Giá trị của tham số (input / Output)

1.4.4 Lớp đối tượng OleDbDataAdapter

STT	Tên Phương Thức	Ý Nghĩa
1	Fill	Lấy dữ liệu từ data source
2	Update	Cập nhật dữ liệu vào data source

