



LINQ

Tổng quan về LinQ



Nội Dung

LINQ

1. Sơ nét về LinQ

- Khái niệm LinQ.
- Tính năng của LinQ.
- Ưu nhược điểm

2. LinQ to SQL

- Khái niệm về LinQ to SQL
- Mô hình ánh xạ .
- Nguyên lí làm việc.
- Thực hiện CRUD với LinQ.

3. LinQ to XML

- Khái niệm về LinQ to XML
- Tạo 1 phần tử và xây dựng cây XML
- XML namespace
- Tính Năng

4. Demo



I.Sơ nét về LinQ

LINQ





Khái niệm LinQ

LINQ

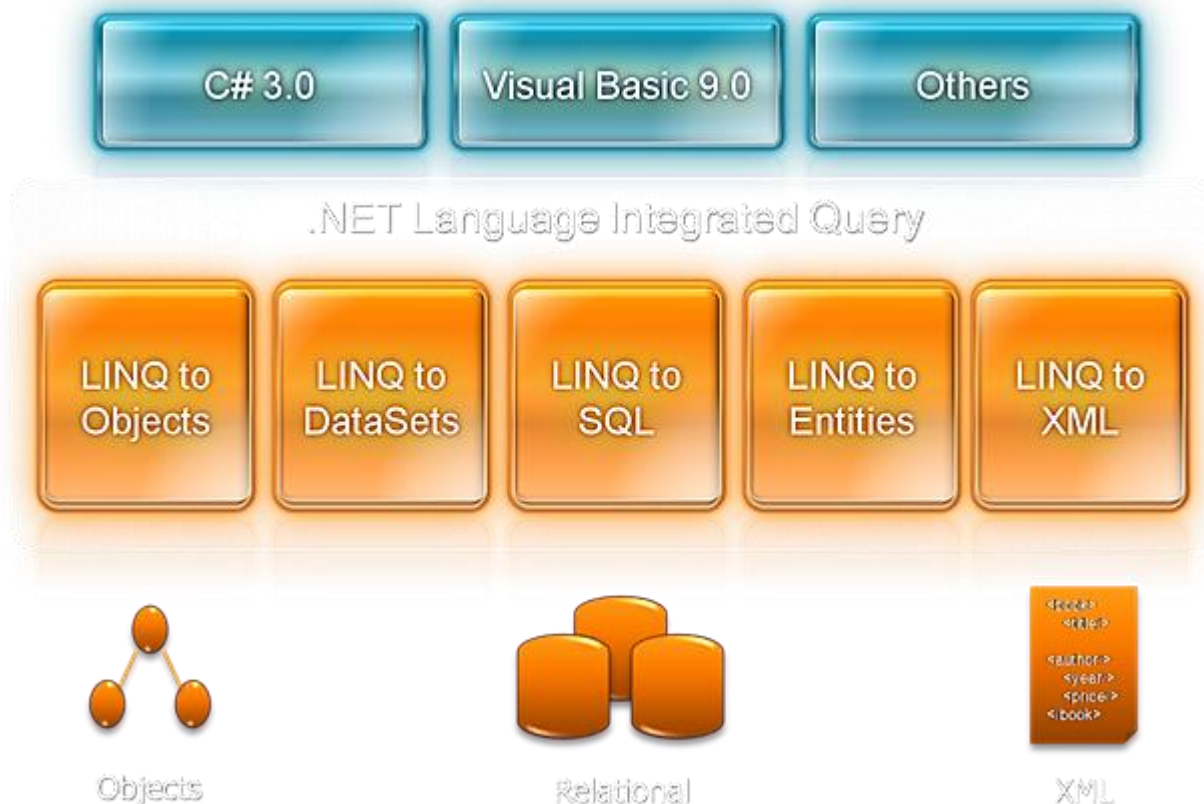
- LinQ được viết tắt từ Language Intergrated Query
- LinQ là ngôn ngữ làm đơn giản hóa việc tương tác dữ liệu thông qua lệnh C# và VB
- Giúp lập trình viên truy vấn csdl với 1 phương thức duy nhất
- Ánh xạ các mô hình dữ liệu sang mô hình hướng đối tượng tương ứng.



Tính năng của LinQ

LINQ

Kiến Trúc LinQ





LINQ

Sơ đồ tính năng





Query Expression

```
class SinhVien
{
    public string masv { get; set; }
    public string ten { get; set; }
    public int tuoi { get; set; }
}

class Program
{
    static void Main(string[] args)
    {
        SinhVien a = new SinhVien { tuoi = 20, ten = "TruongQuangTu", masv = "100131" };
        List<SinhVien> ds = new List<SinhVien>
        {
            new SinhVien(){tuoi = 20, ten = "TruongQuangTu", masv = "100131"},
            new SinhVien(){tuoi = 20, ten = "LeVanTuan", masv = "100133"},
            new SinhVien(){tuoi = 20, ten = "HuynhThiThanhThuy", masv = "100072"},
            new SinhVien(){tuoi = 20, ten = "LeAnhKhoi", masv = "100108"}
        };
        var query = (from p in ds
                     where p.masv == "100131"
                     select p).First();
        Console.WriteLine(query.ten);
    }
}
```

C:\Windows\system32\cmd.exe

TruongQuangTu
Press any key to continue . . .



Lambda Expression

```
int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };

// đếm các số lẻ
int oddNumbers = numbers.Count(n => n % 2 == 1);

// lấy danh sách các số đầu tiên nhỏ hơn 6
var firstNumbersLessThan6 = numbers.TakeWhile(n => n < 6);

// lấy danh sách các số đầu tiên có giá trị lớn hơn vị trí của nó
var firstSmallNumbers = numbers.TakeWhile((n, index) => n >= index);

// lấy các sản phẩm có giá trên 100
products.Where(p => p.UnitPrice > 100);

string[] words = { "cherry", "apple", "blueberry" };
// tổng các ký tự của các phần tử trong mảng
double totalChars = words.Sum(w => w.Length);
```




Ưu ,nhược điểm

LINQ

Ưu điểm:

- Do cú pháp dùng chung → dễ mở rộng (thư viện cho các kiểu dữ liệu khác nhau như XML, SQL, Azure...).
- Giảm thời gian phát triển ứng dụng.
- Vốn dĩ là thuần hướng đối tượng nên dễ quản lí trong ứng dụng so với cách table trong SQL.

Nhược điểm:

- Khả năng truy xuất dữ liệu không cao.
- Chỉ sử dụng được trên .Net framework.



II.LinQ to SQL

LINQ

Customer

Properties

- CustomerID
- CompanyName
- ContactName
- ContactTitle
- Address
- City
- Region
- PostalCode
- Country
- Phone
- Fax

```
var customers =  
    from c in dc.Customers  
    where c.City.StartsWith("A")  
    select c;
```

CustomerID	CompanyName	ContactName
DRACD	Drachenblut Deli...	Sven Ottlie
OLDWO	Old World Delicat...	Rene Phill
RATTC	Rattlesnake Can...	Paula Wils

LINQ

to

SQL



Khái niệm LinQ to SQL

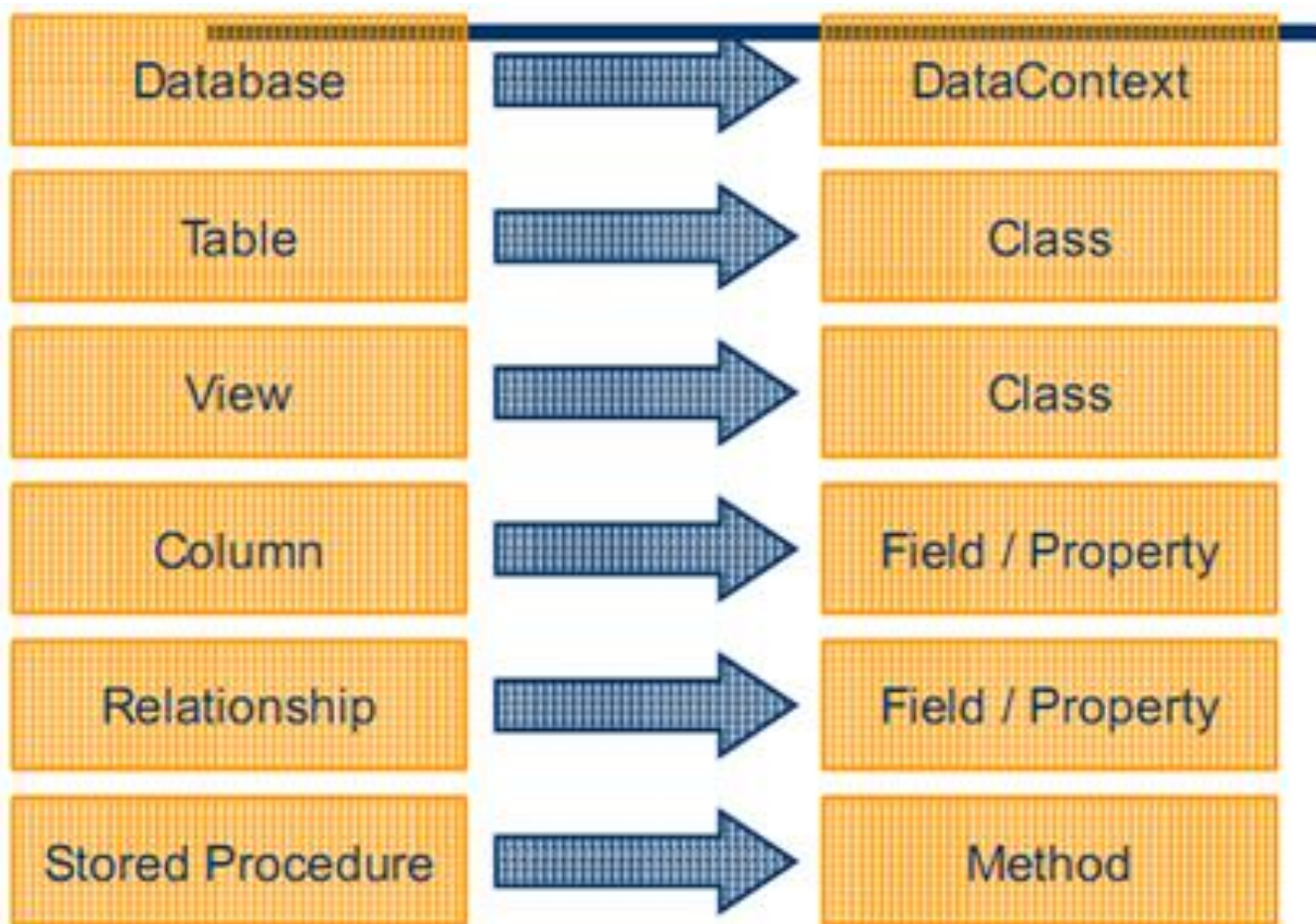
LINQ

LINQ to SQL là một phiên bản hiện thực hóa của O/RM (object relational mapping) có bên trong .NET Framework (.NET 3.5), nó cho phép bạn mô hình hóa một cơ sở dữ liệu dùng các lớp .NET. Sau đó bạn có thể truy vấn cơ sở dữ liệu (CSDL) dùng LINQ, cũng như cập nhật/thêm/xóa dữ liệu từ đó.

LINQ to SQL hỗ trợ đầy đủ transaction, view và các stored procedure (SP). Nó cũng cung cấp một cách dễ dàng để thêm khả năng kiểm tra tính hợp lệ của dữ liệu và các quy tắc vào trong mô hình dữ liệu của bạn.



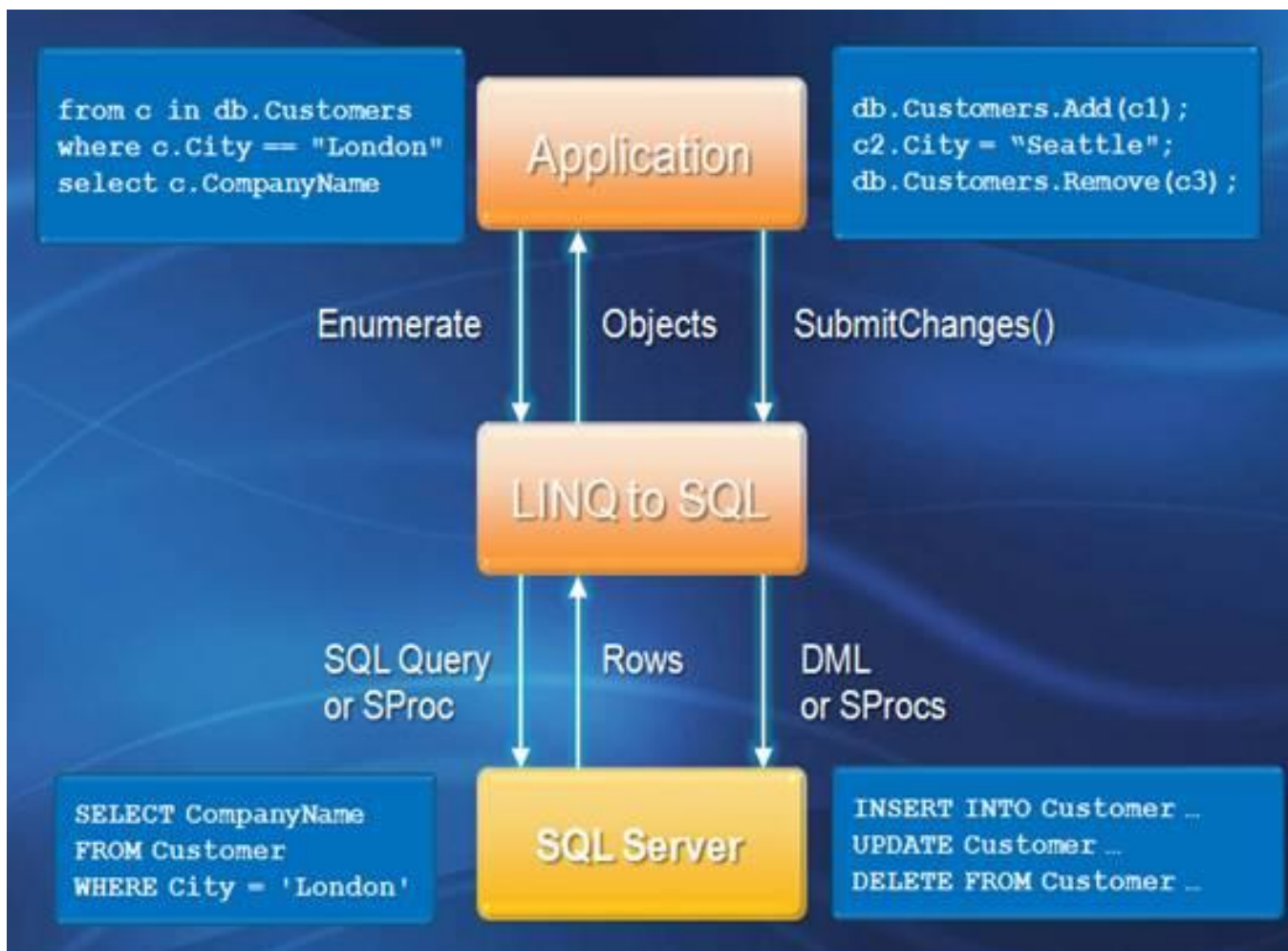
Mô hình ánh xạ





Nguyên lí làm việc

LINQ





Thực hiện CRUD với LinQ

LINQ

1.Create (C)RUD)

- Vì LINQ mọi thao tác đều thông qua đối tượng, do Create ở đây có nghĩa là tạo đối tượng và thêm đối tượng đó vào cơ sở dữ liệu (có thể hiểu là Insert)



2.Read (CRUD)

- Việc đọc dữ liệu chắc hẳn là không thể thiếu đối với ứng dụng cơ sở dữ liệu, đối với lệnh SQL thì nó là Select, nhưng trong LINQ chỉ thông qua đối tượng tương ứng chúng ta sẽ lấy ra được từ 1 đối tượng, cho đến một tập hợp những đối tượng.



3.Update(CRUD)

- Vì LINQ làm việc theo cơ chế tham chiếu (Data Mapping) thông qua đối tượng tham chiếu đến Cơ sở dữ liệu, do đó chúng ta phải lấy ra được đối tượng muốn Update .



4.Delete(CRUD)

- Trong LINQ việc Delete có thể là đơn giản nhất, chúng ta chỉ cần chỉ định rõ tử nào để Delete.



LINQ

LINQ to XML

Visual Studio 2008, Silverlight, and Beyond

Shyam Pather

Principal Development Lead

Microsoft Data Programmability

spather@microsoft.com



Khái niệm LinQ to XML

LINQ

LinQ to XML cung cấp giao diện lập trình XML mà đôn bầy là .NET Language –Integrated Query (LinQ) Framework.

LinQ to XML sử dụng ngôn ngữ mới nhất .NET Framework và được so sánh ,thiết kế lại với giao diện lập trình XML Document Object Model(DOM).



Tạo một phần tử XML

LINQ

Cấu trúc

`XElement(XName name, object content)`

`XElement(XName name)`

`XName`: tên phần tử.

`object`: nội dung của phần tử.

Ví dụ sau tạo phần tử `<Customer>` có nội dung "Adventure Works".

C#

```
XElement n = new XElement("Customer", "Adventure Works");
```

```
Console.WriteLine(n);
```

Xml

```
<Customer>Adventure Works</Customer>
```




Xây dựng một cây XML

LINQ

Sử dụng đối tượng XDocument xây dựng một cây XML. Thông thường ta có thể tạo ra cây XML với nút gốc XElement.

Name	Description
<u>XDocument()()</u>	Khởi tạo một thể hiện mới của lớp XDocument.
<u>XDocument(array<Object>[]()[])</u>	Khởi tạo một thể hiện mới của lớp XDocument với nội dung xác định.
<u>XDocument(XDocument)</u>	Khởi tạo một thể hiện mới của lớp XDocument từ một đối tượng XDocument hiện có.
<u>XDocument(XDeclaration, array<Object>[]()[])</u>	khởi tạo một đối tượng mới của lớp XDocument với XDeclaration quy định và nội dung.



LINQ

C#

```
XDocument doc = new XDocument(  
    new XComment("This is a comment"),  
    new XElement("Root",  
        new XElement("Info5", "info5"),  
        new XElement("Info6", "info6"),  
        new XElement("Info7", "info7"),  
        new XElement("Info8", "info8")  
    )  
);  
Console.WriteLine(doc);
```

Xml

```
<!--This is a comment-->  
<Root>  
    <Child1>data1</Child1>  
    <Child2>data2</Child2>  
    <Child3>data3</Child3>  
    <Child2>data4</Child2>  
</Root>
```



LINQ

Ta có thể sử dụng XElement mà không cần dùng XmlDocument. Lớp XElement là những lớp học chính trong LINQ để giao diện lập trình XML. Trong nhiều trường hợp, ứng dụng của bạn sẽ không yêu cầu bạn tạo một tài liệu. Bằng cách sử dụng lớp XElement, bạn có thể tạo ra một cây XML, thêm cây XML, sửa đổi cây XML, và lưu nó.



XML namespace

LINQ

XML namespace giúp tránh xung đột giữa các bộ phận khác nhau của một tài liệu XML. khi khai báo một namespace, bạn chọn một tên cục bộ sao cho nó là duy nhất.

Những tiền tố làm cho tài liệu XML súc tích và dễ hiểu hơn.



Tạo một namespace trong cây XML

LINQ

```
C#  
  
// Create an XML tree in a namespace, with a specified prefix  
XNamespace aw = "http://www.adventure-works.com";  
XElement root = new XElement(aw + "Root",  
    new XAttribute(XNamespace.Xmlns + "aw", "http://www.adventure-works.com"),  
    new XElement(aw + "Child", "child content")  
);  
Console.WriteLine(root);
```

```
Xml  
  
<aw:Root xmlns:aw="http://www.adventure-works.com">  
    <aw:Child>child content</aw:Child>  
</aw:Root>
```



Những thao tác biến đổi trên cây XML

LINQ

1. Thêm phần tử, thuộc tính và nút vào một cây XML
2. Thay đổi phần tử, thuộc tính và nút của một cây XML
3. Xóa phần tử, thuộc tính và nút từ một cây XML



Tính Năng:

LINQ

Sử dụng LINQ to XML bạn có thể:

- ✓ Load XML từ nhiều file hoặc luồng.
- ✓ Xuất XML ra file hoặc luồng.
- ✓ Truy vấn cây XML bằng những truy vấn LINQ.
- ✓ Thao tác cây XML trong bộ nhớ.
- ✓ Biến đổi cây XML từ dạng này sang dạng khác



Tham Khảo

LINQ

<http://phamnguyen.info/?p=2616>

<http://phamnguyen.info/?p=2621>

<http://elearning.hoasen.edu.vn/course/view.php?id=62>

<http://yinyangit.wordpress.com/2011/07/05/linq-basic%E2%80%93linq-to-xml-architecture-overview/>

<http://msdnvietnam.net/forums/p/1712/5575.aspx>



LINQ

DEMO