

LTNet LINQ

TS. TRAN
ANH TUAN

Nội dung

- LINQ là gì
- Cách dùng LINQ
- LINQ to Object
- LINQ to SQL Server



Linq là gì ?

- LINQ (Language Integrated Query)
- Đây là thư viện mở rộng cho các ngôn ngữ lập trình C# và Visual Basic.NET (có thể mở rộng cho các ngôn ngữ khác) cung cấp khả năng truy vấn trực tiếp dữ liệu Object, CSDL và XML.

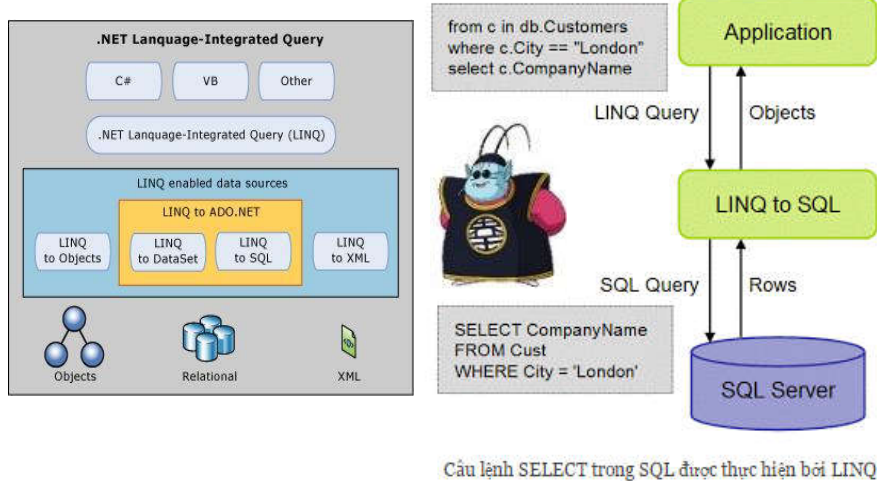


Linq là gì ?

- LINQ là một tập hợp các thành phần mở rộng cho phép viết các câu truy vấn dữ liệu ngay trong một ngôn ngữ lập trình, như C# hoặc VB.NET.
- Khi tạo một đối tượng LINQ thì Visual Studio sẽ tự động sinh ra các lớp có các thành phần tương ứng với CSDL của chúng ta.
- Khi muốn truy vấn, làm việc với CSDL ta chỉ việc gọi và truy xuất các hàm, thủ tục tương ứng của LINQ mà không cần quan tâm đến các câu lệnh SQL thông thường.



Sử dụng Linq



Cách LINQ hoạt động

- Trình biên dịch sẽ tạo mã

```
string[] list = new string[] { "teo", "ty", "bin", "bo"};
```

```
var ret1 = from c in list
            where c.StartsWith("t")
            select c;
```

```
IEnumerable<string> ret2 =
    list.Where(c=>c.StartsWith("t"));
```

Linq áp dụng cho mảng

- Ví dụ

```
int[] nums = new int[] { 1,1,3,2,5,0,9,8,9}
```

```
var result =from n in nums
              where n % 2 == 1
              orderby n
              select n;
```

```
foreach (int i in result)
    Console.WriteLine(i);
```

Output

```
1
1
3
5
9
9
```

Cách LINQ hoạt động

```
string[] list = new string[] { "teo", "ty", "bin", "bo"};
var ret1 = from c in list
            where c.EndsWith("o")
            orderby c
            select new { Id=c.ElementAt(0), Name=c };
var ret2 = list
            .Where(c => c.EndsWith("o"))
            .OrderBy(c => c)
            .Select(c=>new { Id=c.ElementAt(0), Name=c});
foreach (var c in ret1){
    Console.WriteLine(c.Id + " - " +c.Name);
}
```

Cách LINQ hoạt động

- Trì hoãn thực thi

```
var source = new List<string> { "A","B","C"};
var values = from c in source
              select c;

source.Add("D");
foreach (var c in values)
{
    Console.WriteLine(c);
}
```

Out Put:

A
B
C
D



Cách LINQ hoạt động

- Không trì hoãn thực thi

```
var source = new List<string> { "A","B","C"};
var values = (from c in source
              select c).ToList();

source.Add("D");
foreach (var c in values)
{
    Console.WriteLine(c);
}
```

Out Put:

A
B
C



Linq to Object

➤ Examples

```
int []numbers={3,4,5,1,5};
//query syntax
IEnumerable<int> query1 = from num in numbers
                          where num % 2 == 0
                          orderby num
                          select num;

//method syntax
var query2 = numbers
    .Where(n => n % 2 == 0)
    .OrderBy(n => n);
```

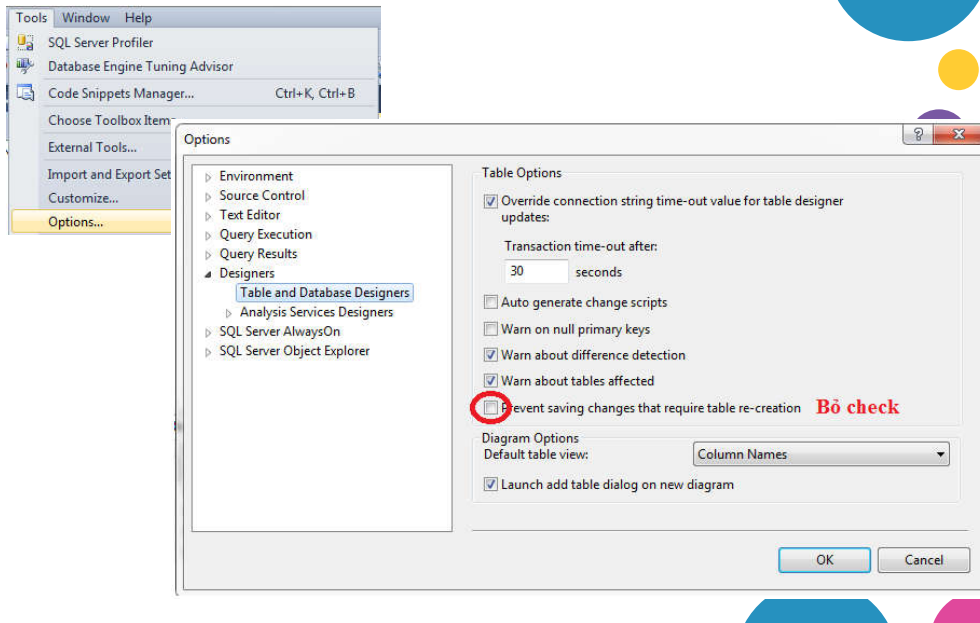


Linq To SQL

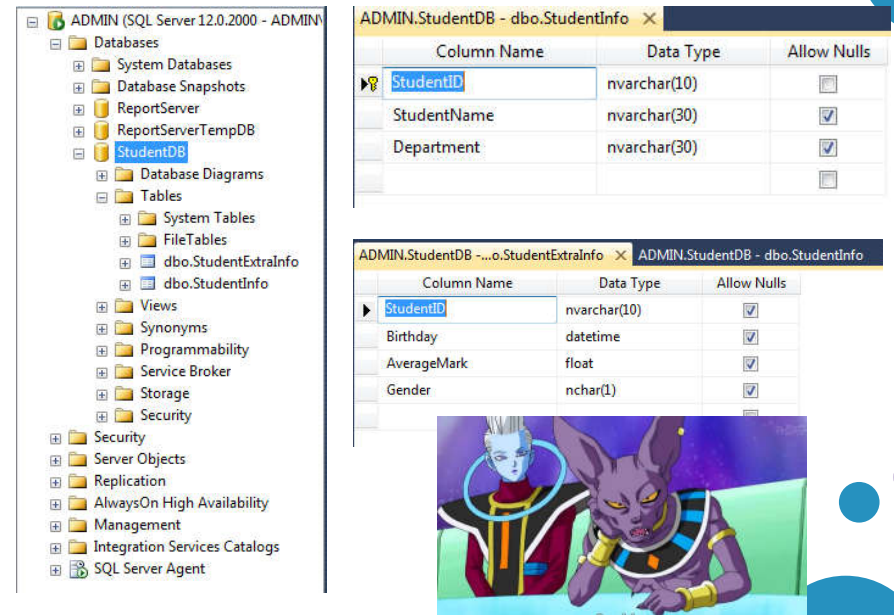
- Tạo Cơ Sở Dữ Liệu bằng SQL
- Bước 1: Mở SQL Server và Login
- Bước 2: New Database > Đặt tên StudentDB
- Bước 3: New Table > Thiết kế table StudentInfo
- Bước 4: New Table > Thiết kế table StudentExtraInfo
- Bước 5: Chọn table StudentInfo > phải chuột chọn "Edit Top 200 rows" để thêm dữ liệu cho bảng StudentInfo
- Bước 6: Chọn table StudentExtraInfo > phải chuột chọn "Edit Top 200 rows" để thêm dữ liệu cho bảng StudentExtraInfo



- Lưu ý: Trong quá trình tạo bảng nếu gặp lỗi không cho lưu thì chọn Tools > Option > vào Designers bỏ check “Prevent saving”



Linq To SQL



Linq To SQL

- Dữ liệu bảng StudentInfo

StudentID	StudentName	Department
ST001	Tran Anh	Mathematics
ST002	Nguyen Van	Mathematics
ST003	Le Binh	Informatics
ST004	Le Dat	Informatics
ST005	Pham Duc	Biologies
NULL	NULL	NULL

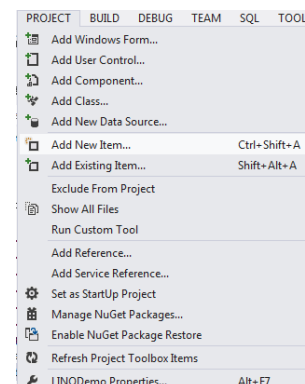


- Dữ liệu bảng StudentMoreInfo

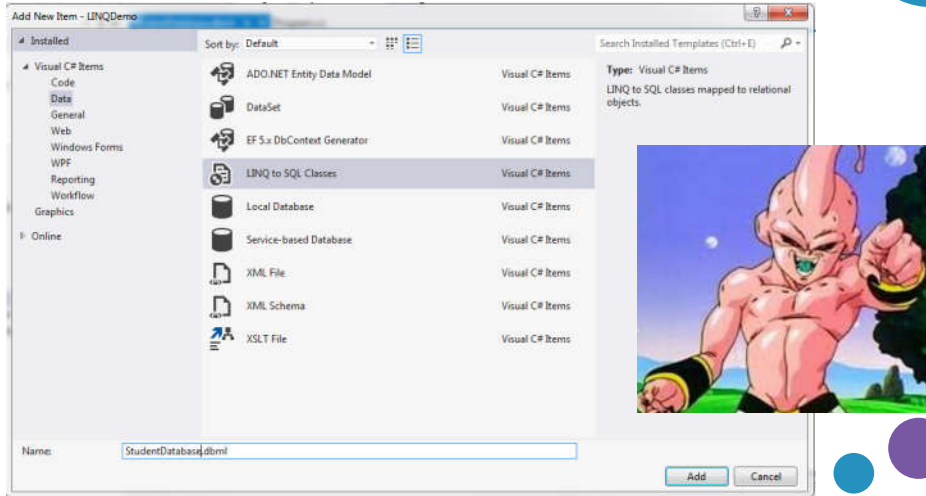
StudentID	Birthday	AverageMark	Gender
ST001	1900-10-26 00:00:00	9.3	M
ST002	1991-07-12 00:00:00	7.5	F
ST003	1990-01-01 00:00:00	8.2	F
ST004	1992-05-18 00:00:00	6.5	M
ST005	1992-11-09 00:00:00	5.5	M
NULL	NULL	NULL	NULL

Linq To SQL

- Tại menu Project > Chọn Add New Item
- Tại hộp thoại Add New Item > Chọn Data > sau đó là LINQ to SQL Classes
- Đặt tên đối tượng kết nối Database là “StudentDatabase.dbml” để dễ quản lý

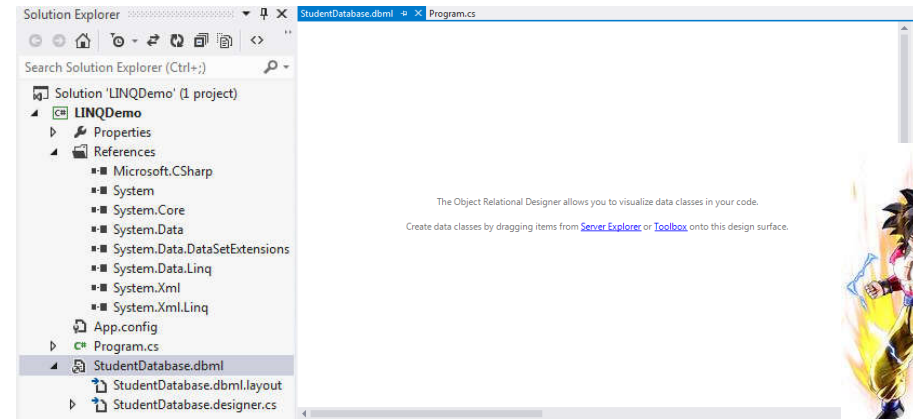


Linq To SQL



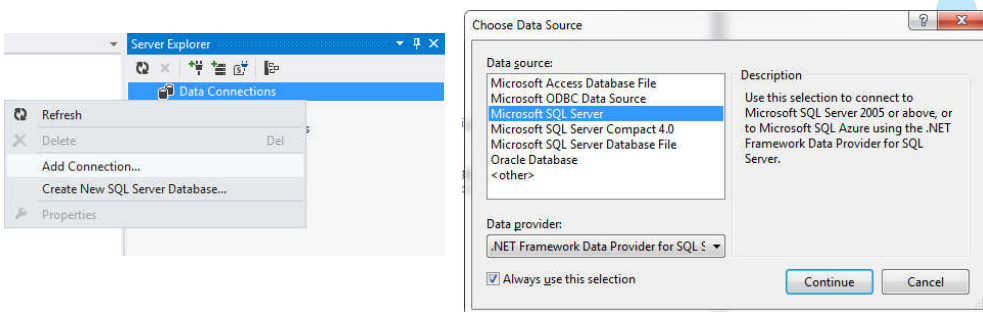
Linq To SQL

- Mở Solution Explorer chọn StudentDatabase (double chuột) rồi chọn Server Explorer



Linq To SQL

- Tại Server Explorer > Chọn phải chuột vào Data Connection > Add Connection
- Phần Choose Datasource chọn SQL Server

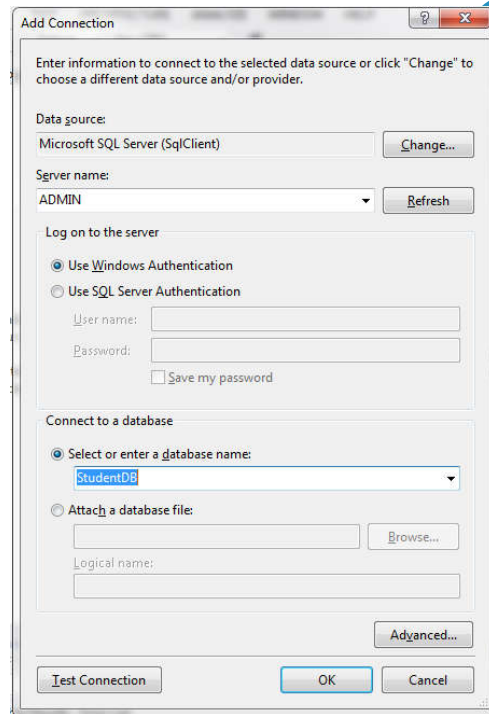
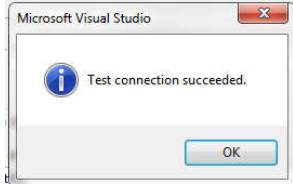


Linq To SQL

- Phần Server Name > Chọn Server Name muốn kết nối
 - Dạng Window Authentication
 - Dạng SQL Authentication (cần username và password)
- Phần "Select or enter a database name" chọn Cơ Sở Dữ Liệu muốn kết nối > Chọn StudentDB
- Nhấn nút "Test Connection" để đảm bảo kết nối thành công

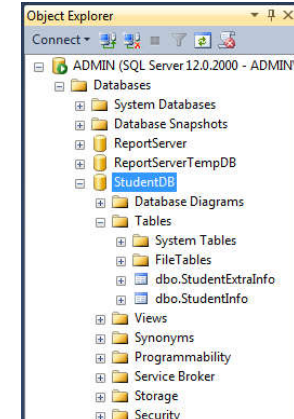
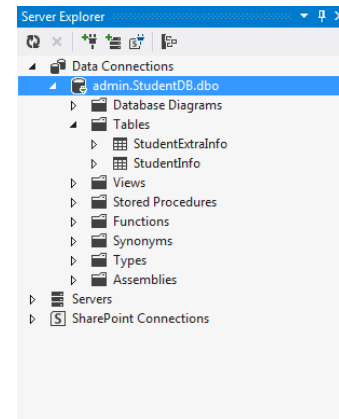


Linq To SQL



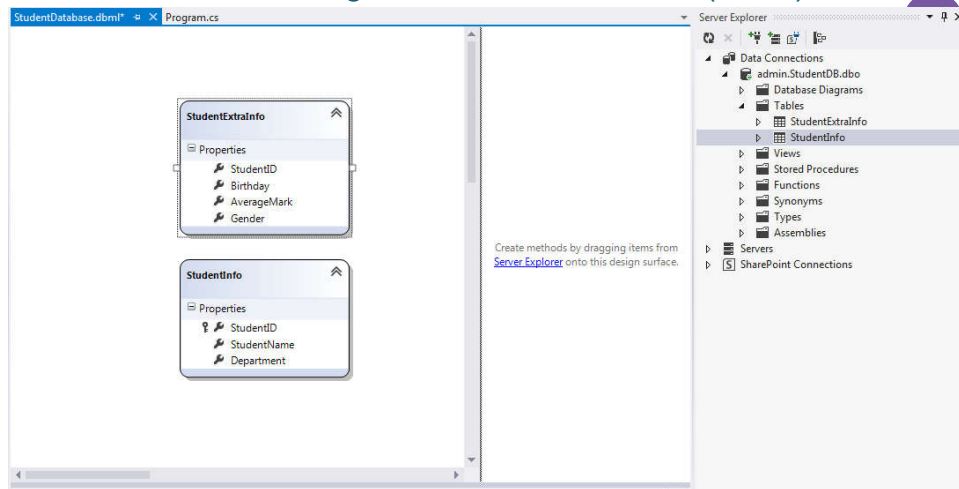
Linq To SQL

- Nếu kết nối thành công thì sẽ hiển thị database StudentDB giống như bên SQL Server



Linq To SQL

- Sau khi có được file ta tiến hành kéo các table cần thiết vào để tự động sinh các thủ tục Linq (Save)



Linq To SQL

- Truy vấn dữ liệu (Query Syntax)
 - Ta phải khai báo 1 biến DataContext để tương tác với Linq
 - Sau đó tiến hành truy vấn trực tiếp trên C#

```
StudentDatabaseDataContext db = new StudentDatabaseDataContext();

var stlist1 = from st in db.StudentInfos
               where st.Department == "Mathematics"
               select st;

// Hoặc chọn một số trường
var stlist2 = from st in db.StudentInfos
               where st.Department == "Mathematics"
               select new
               {
                   st.StudentID,
                   st.StudentName
               };
```



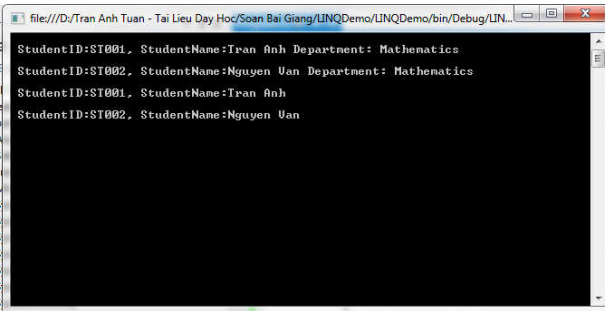
Linq To SQL

- Truy vấn dữ liệu (Query Syntax)

```
foreach (var st in stlist1)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1} Department: {2}",
        st.StudentID, st.StudentName, st.Department);
}

foreach (var st in stlist2)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1}", st.StudentID, st.StudentName);
}

Console.ReadLine();
```

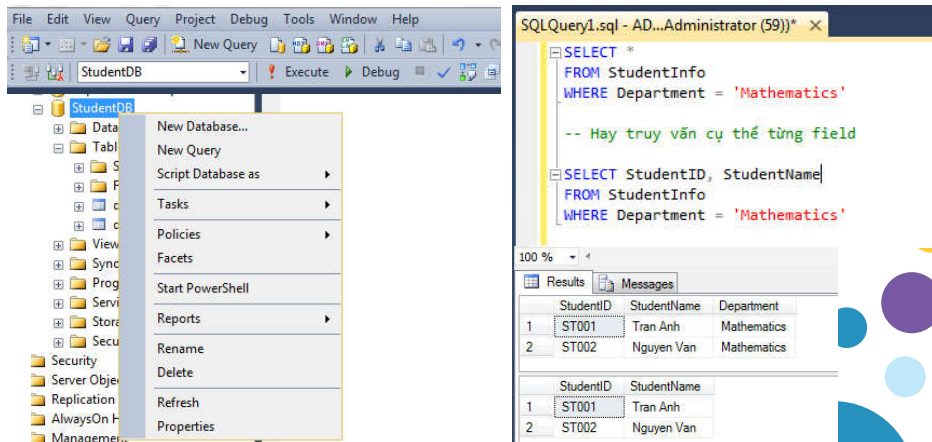


```
StudentID:ST001, StudentName:Tran Anh Department: Mathematics
StudentID:ST002, StudentName:Nguyen Van Department: Mathematics
StudentID:ST001, StudentName:Tran Anh
StudentID:ST002, StudentName:Nguyen Van
```



Linq To SQL

- Câu truy vấn tương ứng trong SQL Server
 - Nhấn phải chuột lên StudentDB > Chọn New Query
 - Nhập câu truy vấn rồi nhấn F5 để thực thi (Hay nút Execute) – Lưu ý phải chọn CSDL StudentDB ở combobox



Linq To SQL

- Truy vấn dữ liệu (Method Syntax)
 - Đây là hình thức truy vấn theo cung cách gọi hàm

```
var stlist3 = db.StudentInfos.Where(st => st.Department == "Mathematics")
    .Select(st => new { st.StudentID, st.StudentName });

foreach (var st in stlist3)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1}", st.StudentID, st.StudentName);
}
```



Linq To SQL

- Truy vấn nhiều bảng (Query Syntax)

```
var stlist4 = from st1 in db.StudentInfos
join st2 in db.StudentExtraInfos on st1.StudentID equals st2.StudentID
select new
{
    MaSV = st1.StudentID,
    TenSV = st1.StudentName,
    Khoa = st1.Department,
    GioiTinh = st2.Gender,
    NgaySinh = st2.Birthday,
    DTB = st2.AverageMark
};

foreach (var st in stlist4)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1}", st.MaSV, st.TenSV);
    Console.WriteLine("\n\t Department:{0}, Gender:{1}", st.Khoa, st.GioiTinh);
    Console.WriteLine("\n\t Birthday:{0}, AverageMark:{1}", st.NgaySinh.ToString(), st.DTB);
}
```



Linq To SQL

- Truy vấn nhiều bảng (Query Syntax)

```
file:///D:/Tran Anh Tuan - Tai Lieu Day Hoc/Soan Bai Giang/LINQDemo/LINQDemo/bin/Debug/LIN...
StudentID:ST002, StudentName:Nguyen Van
Department:Mathematics, Gender:F
Birthday:7/12/1991 12:00:00 AM, AverageMark:7.5
StudentID:ST003, StudentName:Le Binh
Department:Informatics, Gender:F
Birthday:1/1/1990 12:00:00 AM, AverageMark:8.2
StudentID:ST004, StudentName:Le Dat
Department:Informatics, Gender:M
Birthday:5/18/1992 12:00:00 AM, AverageMark:6.5
StudentID:ST005, StudentName:Pham Duc
Department:Biologies, Gender:M
Birthday:11/9/1992 12:00:00 AM, AverageMark:5.5
```

Linq To SQL

- Truy vấn trong SQL Server

```
SELECT
    MaSV = st1.StudentID,
    TenSV = st1.StudentName,
    Khoa = st1.Department,
    GioiTinh = st2.Gender,
    NgaySinh = st2.Birthday,
    DTB = st2.AverageMark
FROM StudentInfo st1
JOIN StudentExtraInfo st2 ON st1.StudentID = st2.StudentID
```

	MaSV	TenSV	Khoa	GioiTinh	NgaySinh	DTB
1	ST001	Tran Anh	Mathematics	M	1900-10-26 00:00:00.000	9.3
2	ST002	Nguyen Van	Mathematics	F	1991-07-12 00:00:00.000	7.5
3	ST003	Le Binh	Informatics	F	1990-01-01 00:00:00.000	8.2
4	ST004	Le Dat	Informatics	M	1992-05-18 00:00:00.000	6.5
5	ST005	Pham Duc	Biologies	M	1992-11-09 00:00:00.000	5.5



Linq To SQL

- Truy vấn nhiều bảng (Method Syntax)

```
var stlist5 = db.StudentInfos.Join(db.StudentExtraInfos, st1 => st1.StudentID, st2 => st2.StudentID,
    (st1, st2) => new
    {
        MaSV = st1.StudentID,
        TenSV = st1.StudentName,
        Khoa = st1.Department,
        GioiTinh = st2.Gender,
        NgaySinh = st2.Birthday,
        DTB = st2.AverageMark
    });

foreach (var st in stlist5)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1}", st.MaSV, st.TenSV);
    Console.WriteLine("\n\t Department:{0}, Gender:{1}", st.Khoa, st.GioiTinh);
    Console.WriteLine("\n\t Birthday:{0}, AverageMark:{1}", st.NgaySinh.ToString(), st.DTB);
}
```



Linq To SQL

- Các biến ở trên trả về có kiểu **IEnumerable**, 1 loại kiểu dữ liệu giống như List. Các thao tác cơ bản với loại biến này:
 - xyz.FirstOrDefault(): Chọn bản ghi đầu tiên hoặc mặc định
 - xyz.Skip(5): Nhảy qua n bản ghi
 - xyz.Take(5): Lấy n bản ghi đầu tiên
 - xyz.ToList(): Chuyển sang kiểu List
 - xyz.Count(): đếm số bản ghi
 - xyz.Select(...), xyz.Where(...), xyz.Join(...): Các câu lệnh truy vấn theo kiểu Method Syntax

Linq To SQL

- Đôi khi bạn không tìm được cú pháp thích hợp hoặc Linq không hỗ trợ loại truy vấn mà vốn có trong SQL thì bạn có thể thực hiện trực tiếp câu lệnh đó thông qua Linq
- Tuy nhiên trước khi thực hiện bạn cần khai báo lớp nhận kết quả truy vấn

```
namespace MainData
{
    public class Student
    {
        public string MaSV {get; set;}
        public string TenSV {get; set;}
        public string Khoa {get; set;}
        public string GioiTinh {get; set;}
        public DateTime NgaySinh {get; set;}
        public double DTB {get; set;}
    }
}
```



Linq To SQL

```
var stlist6 = db.ExecuteQuery<Student>(@"SELECT
    MaSV = st1.StudentID,
    TenSV = st1.StudentName,
    Khoa = st1.Department,
    GioiTinh = st2.Gender,
    NgaySinh = st2.Birthday,
    DTB = st2.AverageMark
FROM StudentInfo st1
JOIN StudentExtraInfo st2 ON st1.StudentID = st2.StudentID
");
```

```
foreach (var st in stlist6)
{
    Console.WriteLine("\n StudentID:{0}, StudentName:{1}", st.MaSV, st.TenSV);
    Console.WriteLine("\n\t Department:{0}, Gender:{1}", st.Khoa, st.GioiTinh);
    Console.WriteLine("\n\t Birthday:{0}, AverageMark:{1}", st.NgaySinh.ToString(), st.DTB);
}
```



Linq To SQL

- Thêm sửa và xóa dữ liệu trong Linq
 - Thêm dữ liệu

```
StudentInfo stnew = new StudentInfo();
stnew.StudentID = "ST006";
stnew.StudentName = "Phan Quy";
stnew.Department = "Biologies";
db.StudentInfos.InsertOnSubmit(stnew);
db.SubmitChanges();
```

	StudentID	StudentName	Department
1	ST001	Tran Anh	Mathematics
2	ST002	Nguyen Van	Mathematics
3	ST003	Le Binh	Informatics
4	ST004	Le Dat	Informatics
5	ST005	Pham Duc	Biologies
6	ST006	Phan Quy	Biologies



Linq To SQL

- Thêm sửa và xóa dữ liệu trong Linq
 - Sửa dữ liệu

```
var ast = (from st in db.StudentInfos
    where st.StudentID == "ST006"
    select st).FirstOrDefault();
ast.Department = "Mathematics";
db.SubmitChanges();
```

	StudentID	StudentName	Department
1	ST001	Tran Anh	Mathematics
2	ST002	Nguyen Van	Mathematics
3	ST003	Le Binh	Informatics
4	ST004	Le Dat	Informatics
5	ST005	Pham Duc	Biologies
6	ST006	Phan Quy	Mathematics



Linq To SQL

- Thêm sửa và xóa dữ liệu trong LinQ
 - Xóa dữ liệu

```
var ast = (from st in db.StudentInfos
           where st.StudentID == "ST006"
           select st).FirstOrDefault();
db.StudentInfos.DeleteOnSubmit(ast);
db.SubmitChanges();
```

	StudentID	StudentName	Department
1	ST001	Tran Anh	Mathematics
2	ST002	Nguyen Van	Mathematics
3	ST003	Le Binh	Informatics
4	ST004	Le Dat	Informatics
5	ST005	Pham Duc	Biologies



Linq To SQL

- Binding
 - *Binding* là cách thuận tiện để thay đổi, cập nhật, thêm mới bản ghi ở CSDL một cách dễ dàng trực quan, sử dụng LinQ.



Linq To SQL

- Thêm , sửa và xóa dữ liệu trong SQL Server

```
-- Thêm dữ liệu
INSERT INTO StudentInfo (StudentID, StudentName, Department)
VALUES ('ST006', 'Pham Quy', 'Biologies');

-- Cập nhật dữ liệu
UPDATE StudentInfo
SET Department = 'Mathematics'
WHERE StudentID = 'ST006'

-- Xóa dữ liệu
DELETE FROM StudentInfo
WHERE StudentID = 'ST006'
```

Messages

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)



Linq To SQL

- Binding
 - Giả sử ta thực hiện Binding ở 1 bảng, có 2 chế độ (kiểu hiển thị) là *GridView* và *Detail*:
 - Chế độ GridView: có 1 gridview để hiển thị các bản ghi trong bảng, tương tự như khi chúng ta chạy câu truy vấn SELECT * trong SQL
 - Detail: các Control cho phép thay đổi giá trị của bản ghi đang được chọn ở Gridview. Khi click vào 1 hàng ở Gridview thì dữ liệu của hàng đó sẽ được điền tự động vào các Control này.



Linq To SQL

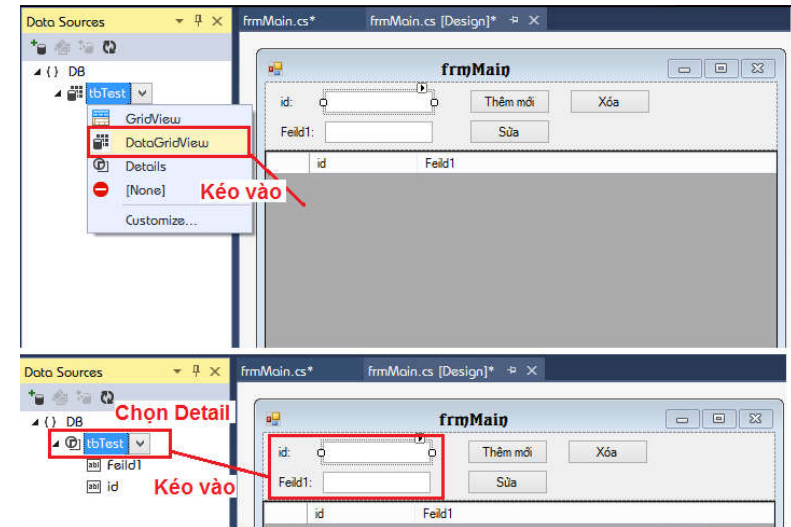
- **Cách dùng Binding:**

- Trong chế độ Design form, mở cửa sổ *Data Sources*, ở đây các bảng trong file DB.dbml (được tạo khi khởi tạo LinQ) sẽ được hiển thị, giúp ta có thể kéo vào trong form của mình
- Với mỗi bảng có các tùy chọn để chọn chế độ xem: GridView, Detail,...



Linq To SQL

- **Cách dùng Binding:**



Linq To SQL

- **Cách dùng Binding:**

- Sau khi kéo như vậy thì nó sinh ra 1 đối tượng là *tbTestBindingSource*, đây chính là đối tượng tương tác với SQL, LinQ.
- Quy định nguồn dữ liệu cho Binding:

```
1 // DataSource là 1 bảng có sẵn
2 tbTestBindingSource.DataSource = db.tbTests;
3
4 // Datasource là một đối tượng IEnumerable lấy từ truy vấn LINQ
5 tbTestBindingSource.DataSource = result;
```

- Bạn có thể kéo 1 BindingNavigator để thực hiện các thao tác thêm, xóa dễ dàng



Linq To SQL

- **Cách dùng Binding:**

- Khi đã có Binding như thế này rồi thì việc thêm sửa xóa cực kỳ đơn giản, không cần code nhiều, sau khi thay đổi trên gridView, detail để lưu tất cả các thay đổi chỉ cần

```
1 tbTestBindingSource.EndEdit();
2 db.SubmitChanges();
```



Linq To SQL

- Cách dùng Binding:

- Các thao tác có thể làm với BindingSource

```
1 tbTestBindingSource.RemoveCurrent(); // Xóa hàng hiện tại
2
3 // Di chuyển trên gridview
4 tbTestBindingSource.MoveFirst();
5 tbTestBindingSource.MoveLast();
6 tbTestBindingSource.MoveNext();
7 tbTestBindingSource.MovePrevious();
8
9 tbTestBindingSource.Position = 5; // Xác định vị trí cho Binding
```

- Sau số thao tác thay đổi trên grid view, bạn muốn xem số hàng thêm mới, sửa xóa??

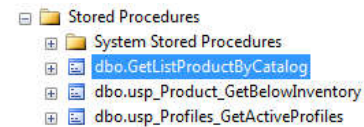
```
1 int insert = db.GetChangeSet().Inserts.Count;
2 int update = db.GetChangeSet().Updates.Count;
3 int delete = db.GetChangeSet().Deletes.Count;
```



Linq To SQL

- Cách dùng Store Procedures

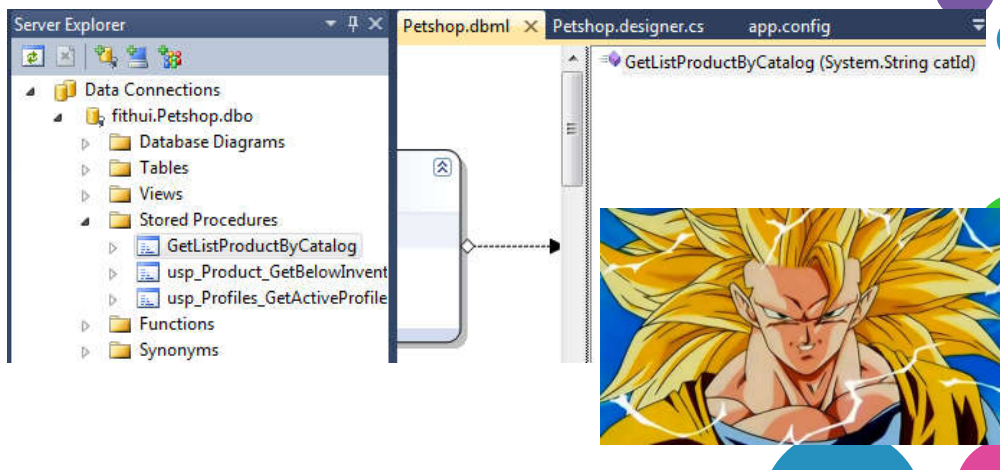
```
USE Petshop
GO
create proc GetListProductByCatalog
@catId varchar(10)
as
select * from Product where CategoryId=@catId
```



Linq To SQL

- Cách dùng Store Procedures

- Kéo thả Store Procedure vào



Linq To SQL

- Cách dùng Store Procedures

```
private void button10_Click(object sender, EventArgs e)
{
    PetshopDataContext db = new PetshopDataContext();
    var products=db.

    ExecuteQuery<>
    GetChangeSet
    GetCommand
    GetHashCode
    GetListProductByCatalog
    GetTable
    GetTable<>
    GetType
    LoadOptions
    ISingleResult<G
    catId>
}
```

```
PetshopDataContext db = new PetshopDataContext();
var products = db.GetListProductByCatalog("birds");
foreach (var pro in products)
{
    Console.WriteLine(pro.Name);
}
```



Linq To SQL

- Cách dùng View & Function

```
CREATE VIEW [Current Product List] AS  
SELECT ProductID,ProductName  
FROM Products  
WHERE Discontinued=No
```

```
CREATE FUNCTION dbo.ConvertInput (@MyValueIn int)  
RETURNS decimal(10,2)  
AS  
BEGIN  
    DECLARE @MyValueOut int;  
    SET @MyValueOut= CAST( @MyValueIn AS decimal(10,2));  
    RETURN(@MyValueOut);  
END;
```



- Cách làm tương tự như Store Procedure khi áp dụng

THANK YOU

