**Phần 1: Truy vấn dữ liệu.**  
**Phần 2: Thực thi dữ liệu**  
**Phần 3: Làm viêc với Transaction.**  
**Phần 4: Linq nâng cao.**

Vì mới tìm hiểu về Linq, nên học hỏi vẫn là mục đích cuối cùng của bài viết này. Dó đó mong nhận đươc sự đóng góp từ các bạn để hoàn chỉnh thêm kiến thức.Triết lý muôn thuở “Nếu muốn hiểu rỏ về vấn đề gì. Hãy bắt đầu từ điều đơn giản nhất.”.

Đầu tiên, Lấy mẫu tin đầu tiên trong bảng(Lấy sản phẩm đầu tiên trong bảng Products). Hai câu truy vấn sau đây trả về cùng kết quả.

var products = context.Products.FirstOrDefault();

var products = (from p in context.Products select p).Skip(0).Take(1); //Tương đương SELECT TOP 1 trong SQL

Tương tự trên, nhưng lấy mẫu tin cuối cùng(Sản phẩm cuối cùng trong bảng Products).

var products = context.Products.LastOrDefault(); //LastOrDefault vẫn chưa đươc hỗ trợ trong .NET 3.5

var products = (from p in context.Products orderby p.ProductID descending

select p).Skip(0).Take(1);

Lấy sản phẩm cố định trong bảng Products.

Product products = context.Products.Single(p=>p.ProductID == 22);

Lấy 10 sản phẩm đầu tiên trong bảng Products.

var products = (from p in context.Products select p).Take(10);

//Tương đươngSELECT TOP 10 trong SQL

Lấy sản phẩm thứ 11 đến 20.

var products = (from p in context.Products select p).Skip(10).Take(10);

Lấy tất cả các sản phẩm trong bảng Products.

var products = from p in context.Products select p;

//Tương đương SELECT \* trong SQL

Lấy các cột cố định trong bảng Products.

var products = from p in context.Products select new

{

p.ProductID,

p.ProductName,

p.UnitPrice,

p.UnitsInStock,

p.UnitsOnOrder

};

Thay đổi tiêu đề cột.

var products = from p in context.Products select new

{

MaSanPham = p.ProductID,

TenSanPham =p.ProductName,

DonGia = p.UnitPrice,

SoLuong =p.UnitsInStock,

SoLuongBan = p.UnitsOnOrder

};//Tương đương **Alias columns**

Sắp xếp các cột trong kết quả trả về.

var products = from p in context.Products

orderby p.ProductID descending, p.ProductName//Tương đương ORDER BY trong SQL

select new

{

p.ProductID,

p.ProductName,

p.UnitPrice,

p.UnitsInStock,

p.UnitsOnOrder

};

Loại bỏ các dòng dữ liệu trùng nhau.

var products = (from od in context.Order\_Details join p in context.Products on od.ProductID equals p.ProductID

select new { od.ProductID, p.ProductName}).Distinct();

//Tương đương SELECT DISTINCT trong SQ

Lấy hóa đơn có sản phẩm bán là 7(ProductID = 7)

var customers = from od in context.Order\_Details where od.ProductID == 7

select od;

Lấy sản phẩm có đơn giá trong khoảng 20-30.

var products = from p in context.Products

where p.UnitPrice >= 20 && p.UnitPrice <= 30//Tương đương BETWEEN trong SQL

select new { p.ProductName, p.UnitPrice, p.UnitsInStock, p.UnitsOnOrder };

Lấy tất cả các sản phẩm đã bán ra.

var products = from p in context.Products where (from od in context.Order\_Detailsselect od.ProductID).Contains(p.ProductID)

select new {p.ProductID, p.ProductName, p.UnitPrice, p.UnitsInStock,

p.UnitsOnOrder };//Tương đương từ khóa IN trong SQL

Lấy tất cả các sản phẩm chưa bán ra.

var products = from p in context.Products

where !(from od in context.Order\_Details

select od.ProductID).Contains(p.ProductID)

//Tương đương từ khóa NOT IN trong SQL

select new {

p.ProductID, p.ProductName, p.UnitPrice, p.UnitsInStock, p.UnitsOnOrder };

Lấy danh sách khách hàng có tên bắt đầu là “Sa”.

var customers = from c in context.Customers

where c.CompanyName.StartsWith("Sa")//Tương đương LIKE 'Sa%' trong SQL

select new { c.CustomerID, c.CompanyName, c.ContactName, c.City, c.Country };

Lấy danh sách khách hàng có tên kết thúc bằng “es”.

var customers = from c in context.Customers

where c.CompanyName.EndsWith("es")//Tương đương LIKE '%es' trong SQL

select new { c.CustomerID, c.CompanyName, c.ContactName, c.City, c.Country };

Lấy danh sách khách hàng có chứa chuỗi “sa” trong CompanyName.

var customers = from c in context.Customers

where c.CompanyName.Contains("Sa")//Tương đương LIKE '%Sa%' trong SQL

select new { c.CustomerID, c.CompanyName, c.ContactName, c.City, c.Country };

Lấy danh sách khách hàng tại Anh và Pháp.

var customers = (from c in context.Customers

where c.Country == "UK"

select new

{

c.CustomerID,

c.CompanyName,

c.ContactName,

c.City, c.Country

}).Union(//Tương đương PHÉP HỢP UNION trong SQL

from c in context.Customers

where c.Country == "France"

select new {

c.CustomerID,

c.CompanyName,

c.ContactName,

c.City,

c.Country } );

Lấy danh sách khách hàng có hóa đơn.

var customers = from c in context.Customers

join o in context.Orders on c.CustomerID equals o.CustomerID

//Tương đương INNER JOIN trong SQL

orderby c.CustomerID, o.OrderID

select new {

c.CustomerID,

o.OrderID,

OrderDate = o.OrderDate.Value.ToShortDateString(),

c.CompanyName,

c.City,

c.Country };

Lấy hóa đơn của khách hàng, bao gồm khách hàng không có hóa đơn.

var customers = from c in context.Customers

join o in context.Orders

on c.CustomerID equals o.CustomerID into OD

from o in OD.DefaultIfEmpty()//Tương đương OUTER JOIN trong SQL

select new

{

CustomerID = c.CustomerID,

CompanyName = c.CompanyName,

OrderID = o == null ? 0 : o.OrderID

};

Đếm số hóa đơn đối với mỗi khách hàng

var customers = from c in context.Customers

select new { c.CustomerID, c.CompanyName, c.Orders.Count };

Đếm số hóa đơn đối với khách hàng có lớn hơn 5 hóa đơn.

var customers = from c in context.Customers

where c.Orders.Count >5

select new {c.CustomerID, c.CompanyName, c.Orders.Count };

Đếm số hàng hóa trên mỗi hóa đơn và tính đơn giá bình quân của hóa đơn.

var orders = from od in context.Order\_Details

group od by od.OrderID into OD//Tương đương GROUP BY trong SQL

orderby OD.Key

select new

{

OrderID = OD.Key,

ProductID = OD.Count(),

UnitPrice = OD.Average(m => m.UnitPrice)

};

Đếm số hàng hóa trên mỗi hóa đơn và tính giá trị hóa đơn.

var orders = from od in context.Order\_Details

group od by od.OrderID into OD

orderby OD.Key

select new

{

OrderID = OD.Key,

Product = OD.Count(),

UnitPrice = OD.Sum(m => m.UnitPrice)

};

Lấy sản phẩm có giá bán cao nhất và thấp nhất trong mỗi hóa đơn.

var orders = from od in context.Order\_Details

group od by od.OrderID into OD

orderby OD.Key

select new

{

OrderID = OD.Key,

MinPrice = OD.Min(m => m.UnitPrice),

MaxPrice = OD.Max(m => m.UnitPrice)

};

Lấy giá bán cao nhất và thấp nhất của mỏi sản phẩm.

var orders = from od in context.Order\_Details

join p in context.Products on od.ProductID equals p.ProductID

group od by new { p.ProductID, p.ProductName } into OD

orderby OD.Key.ProductID

select new

{

ProductID = OD.Key.ProductID,

ProductName = OD.Key.ProductName,

MaxPrice = OD.Max(m => m.UnitPrice),

MinPrice = OD.Min(m => m.UnitPrice)

};

Lưu ý: Theo mình thấy Linq To SQL và Linq trong ADO.NET Entity Data Model có khác nhau.