

# Chương 6: Lập trình LINQ

## 1. Tổng quan về LINQ

### 1.1 Giới thiệu LINQ

#### 1. Collections

```
List<string> dsChuoai = new List<string>
{
    "Thiện", "cần", "ở", "tại", "lòng", "ta", "chữ",
    "Tâm", "kia", "mới", "bằng", "ba", "chữ", "Tài"
};
```

#### 2. Dữ liệu XML

```
<?xml version="1.0" standalone="yes"?>
<Mon_hoc>
  <MonHoc>
    <Ma_mon>C0</Ma_mon>
    <Ten_mon>Cơ lý thuyết</Ten_mon>
  </MonHoc>
  <MonHoc>
    <Ma_mon>CS</Ma_mon>
    <Ten_mon>Cơ sở dữ liệu</Ten_mon>
  </MonHoc>
  <MonHoc>
    <Ma_mon>CT</Ma_mon>
    <Ten_mon>Cấu trúc dữ liệu</Ten_mon>
  </MonHoc>
  <MonHoc>
    <Ma_mon>RR</Ma_mon>
    <Ten_mon>Toán rời rạc</Ten_mon>
  </MonHoc>
  <MonHoc>
    <Ma_mon>TC</Ma_mon>
    <Ten_mon>Toán cao cấp</Ten_mon>
  </MonHoc>
  <MonHoc>
    <Ma_mon>TT</Ma_mon>
    <Ten_mon>Trí tuệ nhân tạo</Ten_mon>
  </MonHoc>
</Mon_hoc>
```

#### 3. Dữ liệu trong CSDL quan hệ

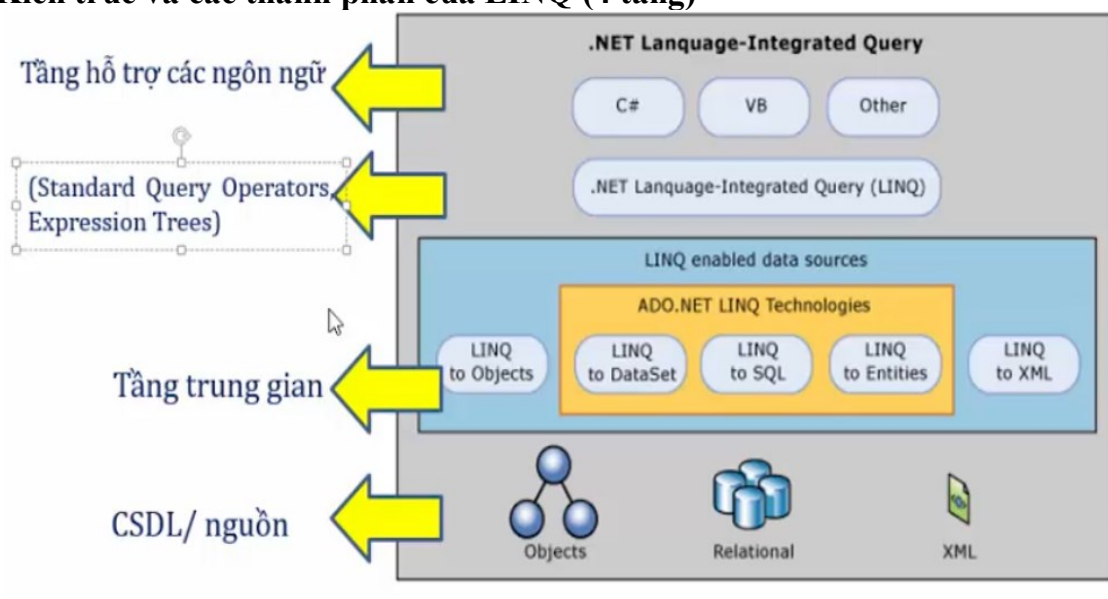
Ma_mon	Ten_mon
CO	Cơ lý thuyết
CS	Cơ sở dữ liệu
CT	Cấu trúc dữ liệu
LT	Lượng tử
RR	Toán rời rạc
TC	Toán cao cấp
TT	Trí tuệ nhân tạo

Sẽ giúp chúng ta giải quyết vấn đề này

**Code truy xuất dữ liệu sẽ khác nhau đối với từng loại dữ liệu → Không thống nhất code**

- Language Integrated Query (LINQ) là ngôn ngữ truy vấn hợp nhất trên các loại dữ liệu khác nhau.
- Với LINQ, bạn có thể truy vấn nhiều nguồn dữ liệu khác nhau trong C#: đối tượng (object), cơ sở dữ liệu SQL, tài liệu XML, mô hình dữ liệu thực thể (entity data model).
- Đưa ra khả năng lập trình mới trong .NET - Giải pháp lập trình hợp nhất

### 1.2 Kiến trúc và các thành phần của LINQ (4 tầng)



### 1.3 Các thư viện hỗ trợ LINQ

- **System.Linq**: hỗ trợ sử dụng các Objects
- **System.Data.Linq**: hỗ trợ sử dụng các CSDL quan hệ
- **System.Data.Objects**: hỗ trợ sử dụng các Entities
- **System.XML.Linq**: hỗ trợ sử dụng XML

### 1.4 Truy vấn LINQ theo biểu thức

- **Cú pháp:**

```
var Tên_biến = from id in source
                [ where condition
                  orderby ordering, ... ]
                select id/expr
```

- **Trong đó:**
  - + Từ khóa **var** để chỉ định *Tên\_biến* có kiểu bất kỳ
  - + **source** có thể là: mảng, tập hợp, bảng, xml...
  - + **id** là định danh cho **source**
- **Ba thành phần hoạt động của truy vấn:**
  - Nhập nguồn dữ liệu
  - Tạo truy vấn
  - Thi hành truy vấn

- **Ví dụ:**

// 1. Data source.

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

// 2. Query creation.

```
var numQuery = from num in numbers
                where (num % 2) == 0
                orderby num descending
                select num;
```

// 3. Query execution.

```
foreach (int num in numQuery)
{ Console.WriteLine("{0} ", num); }
```

### 1.5 Truy vấn LINQ theo phương thức mở rộng

#### a) Biểu thức Lambda – Lambda Expression

- **Cú pháp:**

*Tham\_số\_đầu\_vào* => *Biểu\_thức hoặc Khối\_lệnh*

- **Trong đó:**

- **=>**: Gọi là toán tử **Lambda**
- *Tham\_số\_đầu\_vào*: Có thể có 1 hoặc nhiều tham số (có kiểu tường minh hoặc không), cách nhau bằng dấu phẩy

- **Ví dụ:**

```
x=>x*x+2; x=> {int y=3; x=y*y+4; }
```

```
(x,y)=>x+y; (x,y)=> x==y;
```

## b) Cú pháp LINQ theo phương thức mở rộng

- Cú pháp:

```
var Tên_biến = Source . Tên_pt_mở_rộng(Biểu_thức_Lambda)
```

- Trong đó:

- + **source** có thể là: mảng, tập hợp, bảng, xml...

- + **Tên\_phương\_thức\_mở\_rộng**: có thể là: *Select, Where, Orderby,...*

- Ví dụ:

**// 1. Data source.**

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

**// 2. Query creation.**

```
var numQuery = numbers.Where (num=>(num % 2)== 0)
                        .OrderBy(num => num)
                        //.OrderByDescending(num => num)
                        .Select (num=>num);
```

**// 3. Query execution.**

```
foreach (int num in numQuery)
    { Console.WriteLine("{0} ", num); }
Console.ReadLine();
```

### Danh sách các phương thức mở rộng

#### + Phương thức truy vấn cơ bản

Phương thức	Mô tả	Ví dụ
.Where(e=>điều kiện)	Lọc	Students.Where(s=>s.Marks > 9)
.GroupBy(e=>biểu thức)	Nhóm	Students.GroupBy(s=>s.Clazz)
.OrderBy(e=>biểu thức) .OrderByDescending(e=>biểu thức)	Sắp xếp	Students.OrderBy(s=>s.Name)
.Select(e=>đối tượng)	Chọn	Students.Select(s=>new{s.Name, s.Marks})
.Distinct()	Giữ 1 của các đối tượng giống nhau	Numbers.Distinct()

### + Phương thức truy vấn phân trang

Phương thức	Mô tả	Ví dụ
.Take(số lượng)	Lấy các phần tử đầu	Students.Take(5)
.Skip(số lượng)	Bỏ qua các phần tử đầu	Students.Skip(3).Take(6)
.TakeWhile(e=>đ.kiện)	Lấy các phần tử đầu thỏa điều kiện	Students.TakeWhile(s=>s.Marks < 4)
.SkipWhile(e=>đ.kiện)	Bỏ qua các phần tử đầu thỏa điều kiện	Students.SkipWhile(s=>s.Marks < 0)

### + Phương thức truy vấn 1 thực thể

Phương thức	Mô tả	Ví dụ
.Single(e=>đ.kiện)	Lấy 1 phần tử thỏa điều kiện. Ngoại lệ nếu không tìm thấy hoặc nhiều hơn một.	Students.Single(s=>s.Id=="Hoa")
.First()	Lấy phần tử đầu	Students.First()
.Last()	Lấy phần tử cuối	Students.Last()

### + Phương thức truy vấn 1 thống kê

Phương thức	Mô tả	Ví dụ
.Sum(e=>biểu thức số học)	Tính tổng	Students.Sum(s=>s.Marks)
.Count(e=>biểu thức số học)	Đếm số lượng	Students.Count(s=>s.Id)
.Min(e=>biểu thức số học)	Giá trị nhỏ nhất	Students.Min(s=>s.Marks)
.Max(e=>biểu thức số học)	Giá trị lớn nhất	Students.Max(s=>s.Marks)
.Average(e=>biểu thức số học)	Giá trị trung bình	Students.Average(s=>s.Marks)

## 2. LINQ TO SQL

LINQ to SQL là một phiên bản hiện thực hóa của O/RM (object relational mapping) có bên trong .NET Framework 3.5, cho phép 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.

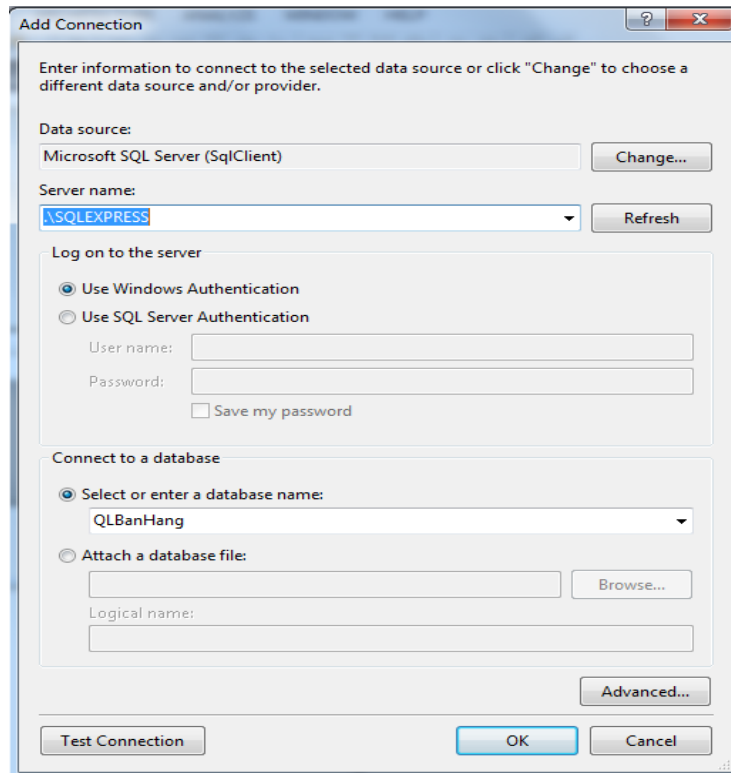
## 2.1 Kết nối đến CSDL

Cho CSDL **QLBanHang** gồm 2 bảng:

**SanPham**(MaSP, TenSP, DonGia, MaLoai)

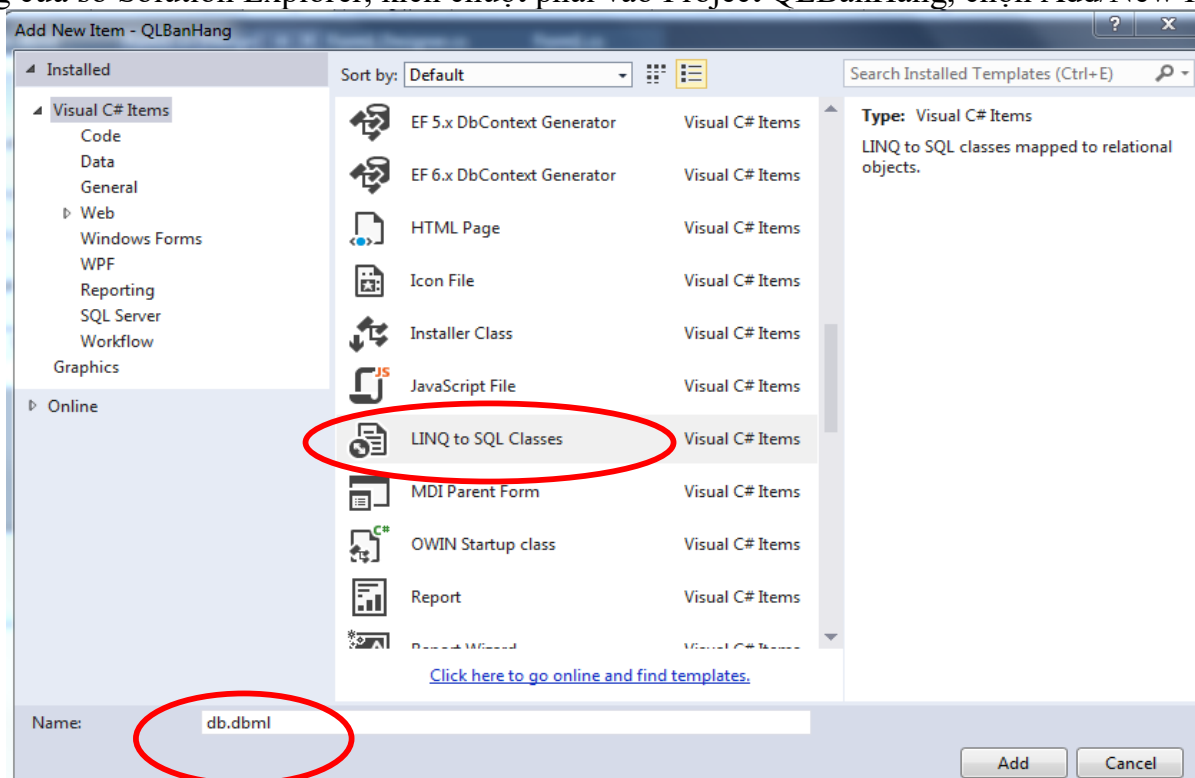
**LoaiSP**(MaLoai, TenLoai)

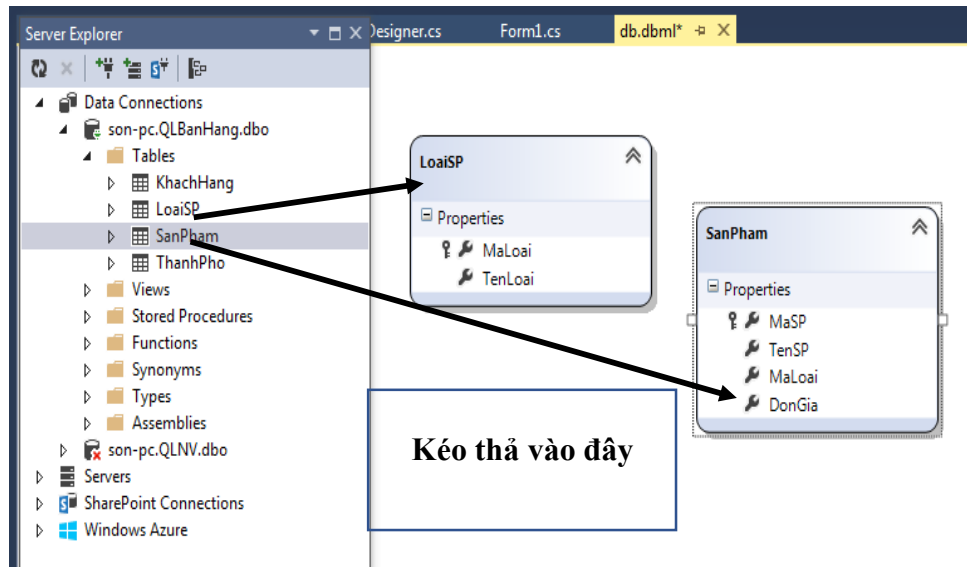
- View -> Server Explorer
- Kích phải chuột lên DataConnection -> Add Connection



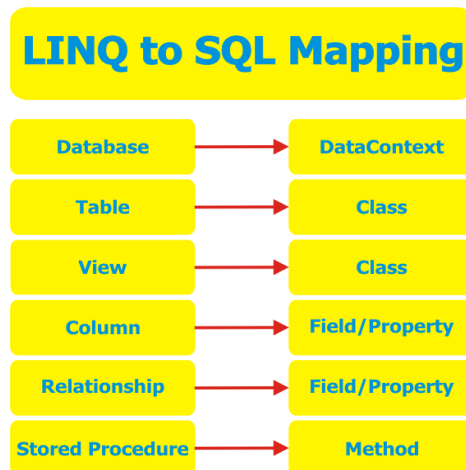
## 2.2 Tạo ra mô hình dữ liệu LINQ TO SQL

Trong cửa sổ Solution Explorer, kích chuột phải vào Project QLBanHang, chọn Add/New Item



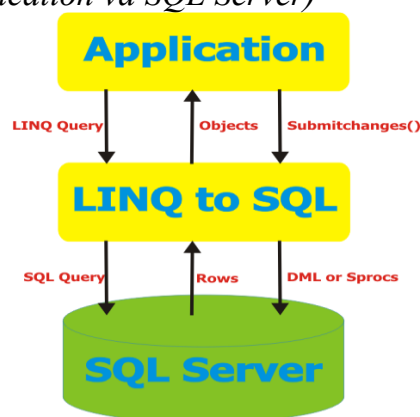


## 2.3 Mô hình ánh xạ của LINQ TO SQL (ánh xạ CSDL theo hướng đối tượng)



## 2.4 Tầng kiến trúc của LINQ TO SQL

(là cầu nối giao tiếp giữa Application và SQL Server)



## 4.5 Các ví dụ sử dụng toán tử/ phương thức trong LINQ TO SQL

### SELECT

VD 1: Hiển thị DL bảng SanPham

```
dbDataContext db = new dbDataContext();
```

```
Cách 1: var sp = from p in db.SanPhams
               select p;
```

Cách 2: `var sp = db.SanPhams  
.Select(x => x);`

VD 2: Hiển thị DL bảng SanPham (Lựa chọn cột hiển thị)

`dbDataContext db = new dbDataContext();`

Cách 1: `var sp = from p in db.SanPhams  
select new { p.MaSP, p.TenSP, p.MaLoai};`

Cách 2: `var sp = db.SanPhams  
.Select(p => new {p.MaSP, p.TenSP, p.MaLoai });`

### **TAKE, SKIP**

VD 3: Lấy mẫu tin đầu tiên trong bảng (Lấy sản phẩm đầu tiên trong bảng SanPham).

`dbDataContext db = new dbDataContext();`

Cách 1: `var sp = (from p in db.SanPhams  
select p).Skip(0).Take(1);` //Tương đương SELECT TOP 1 trong SQL

Cách 2: `var sp = db.SanPhams  
.Select(x => x).Skip(0).Take(1);`

VD 4: Tương tự trên, nhưng lấy mẫu tin cuối cùng

`var sp = (from p in db.SanPhams  
orderby p.MaSP descending  
select p).Skip(0).Take(1);`

VD 5: Tương tự trên, nhưng lấy mẫu tin thứ 5 và 6

`var sp = (from p in db.SanPhams  
select p).Skip(4).Take(2);`

### **ORDERBY**

VD 6: Sắp xếp giảm dần theo cột đơn giá (mặc định tăng dần)

`dbDataContext db = new dbDataContext();`

Cách 1: `var sp = from p in db.SanPhams  
orderby p.DonGia descending  
select new { p.MaSP, p.TenSP, p.DonGia };`

Cách 2: `var sp = db.SanPhams  
.OrderByDescending(x=>x.DonGia)  
.Select(x=>new {x.MaSP, x.TenSP, x.DonGia });`

### **DISTINCT**

VD 7: Loại bỏ các phần tử trùng nhau

`dbDataContext db = new dbDataContext();`

`var sp = (from p in db.SanPhams  
select new { p.MaSP, p.TenSP, p.MaLoai}).Distinct();`

### **WHERE**

VD 8: Lấy điều kiện theo MaLoai

`dbDataContext db = new dbDataContext();`

Cách 1: `var sp = from p in db.SanPhams  
where p.MaLoai=="L01"  
select new { p.MaSP, p.TenSP, p.MaLoai };`

Cách 2: `var sp = db.SanPhams  
.Where (p=>p.MaLoai=="L1")  
.Select(x => new { x.MaSP, x.TenSP, x.DonGia });`

### **JOIN**

VD 9: Liên kết nhiều bảng

`dbDataContext db = new dbDataContext();`

```

var sp = from p in db.SanPhams
        join k in db.LoaiSPs
        on p.MaLoai equals k.MaLoai
        select new { p.MaSP, p.TenSP, k.TenLoai };

```

## GROUP

VD 10: Nhóm dữ liệu (Tính tổng đơn giá theo loại sản phẩm)

```
dbDataContext db = new dbDataContext();
```

```

var sp = from p in db.SanPhams
        join k in db.LoaiSPs
        on p.MaLoai equals k.MaLoai
        group p by p.MaLoai into kq
        select new { MaLoai = kq.Key, TongDonGia = kq.Sum(t => t.DonGia) };

```

Kết quả:

MaLoai	TongDonGia
L01	10000000
L02	900000

## BÀI TẬP ỨNG DỤNG

### THIẾT KẾ GIAO DIỆN WEB

Mã sản phẩm:	<input type="text" value="sp4"/>	<input type="button" value="Thêm"/>
Tên sản phẩm:	<input type="text" value="Tivi"/>	<input type="button" value="Sửa"/>
Loại sản phẩm:	<div>Điện tử ▼</div>	<input type="button" value="Xóa"/>
Đơn giá:	<input type="text" value="4000"/>	<input type="button" value="Tìm"/>

Mã SP	Tên SP	Mã loại	Đơn giá	
sp1	Quần bò	L2	400	<a href="#">Chọn dòng</a>
sp2	Áo cộc tay	L2	350	<a href="#">Chọn dòng</a>
sp3	Áo khoác	L2	800	<a href="#">Chọn dòng</a>
sp4	Tivi	L1	4000	<a href="#">Chọn dòng</a>
sp5	Tủ lạnh	L1	3500	<a href="#">Chọn dòng</a>



```
public void Load_SanPham()
```

```
{
```

```
    dbDataContext data = new dbDataContext();
```

```
    var sp = from p in data.SanPhams
```

```
        select p;
```

```
    dgSanPham.DataSource = sp; dgSanPham.DataBind();
```

```
}
```

```
public void Load_ComboLoaiSP()
```

```
{
```

```
    dbDataContext data = new dbDataContext();
```

```
    var lsp = from p in data.LoaiSPs
```

```
        select p;
```

```
    comboLoai.DataSource = lsp;
```

```
    comboLoai.DataTextField = "TenLoai";
```

```
    comboLoai.DataValueField = "MaLoai";
```

```
    comboLoai.DataBind();    }
```

```
private void Form1_Load(object sender, EventArgs e)
```

```
{
```

```
    Load_SanPham();
```

```
    if (!IsPostBack)    { Load_ComboLoaiSP(); }
```

```
}
```

```
THÊM DỮ LIỆU: Sử dụng InsertOnSubmit() và SubmitChanges()
```

```
    dbDataContext data = new dbDataContext();
```

```
    SanPham them = new SanPham();
```

```
    them.MaSP = txtMaSanPham.Text.Trim();
```

```
    them.TenSP = txtTenSanPham.Text;
```

```
    them.MaLoai = cboLoai.SelectedValue.ToString();
```

```
    them.DonGia = Convert.ToInt32(txtDonGia.Text);
```

```
    data.SanPhams.InsertOnSubmit(them);
```

```
    data.SubmitChanges();
```

```
    Load_SanPham();
```

```
XÓA DỮ LIỆU: Sử dụng DeleteOnSubmit() và SubmitChanges()
```

```
    dbDataContext data = new dbDataContext();
```

```
    var xoa = from sp in data.SanPhams
```

```
        where sp.MaSP == txtMaSanPham.Text select sp;
```

```
    foreach(var i in xoa)
```

```
    {
```

```
        data.SanPhams.DeleteOnSubmit(i);
```

```
        data.SubmitChanges();
```

```
    }
```

```
    Load_SanPham();
```

<b>SỬA DỮ LIỆU:</b> <i>Sử dụng SubmitChanges()</i>
dbDataContext data = new dbDataContext();
var capnhat = data.SanPhams.Single(sp => sp.MaSP == txtMaSanPham.Text);
capnhat.TenSP = txtTenSanPham.Text;
capnhat.MaLoai = cboLoai.SelectedValue.ToString();
capnhat.DonGia = Convert.ToInt32(txtDonGia.Text);
data.SubmitChanges();
Load_SanPham();
<i>*Phương thức <b>Single</b>: trả về 1 phần tử duy nhất trong 1 tập với điều kiện nào đó</i>
<b>protected void</b> dgSanPham_SelectedIndexChanged( <b>object</b> sender, <b>EventArgs</b> e)
{
<b>int</b> dong = dgSanPham.SelectedIndex;
txtMa.Text = dgSanPham.Rows[dong].Cells[0].Text;
txtTen.Text = dgSanPham.Rows[dong].Cells[1].Text;
comboLoai.SelectedValue = dgSanPham.Rows[dong].Cells[2].Text;
txtDongia.Text = dgSanPham.Rows[dong].Cells[3].Text;
}
<b>Bài tập:</b>
1- Tìm danh sách sản phẩm theo loại sản phẩm được chọn trong ComboBox
2- Tìm danh sách sản phẩm trong khoảng đơn giá [a-b]