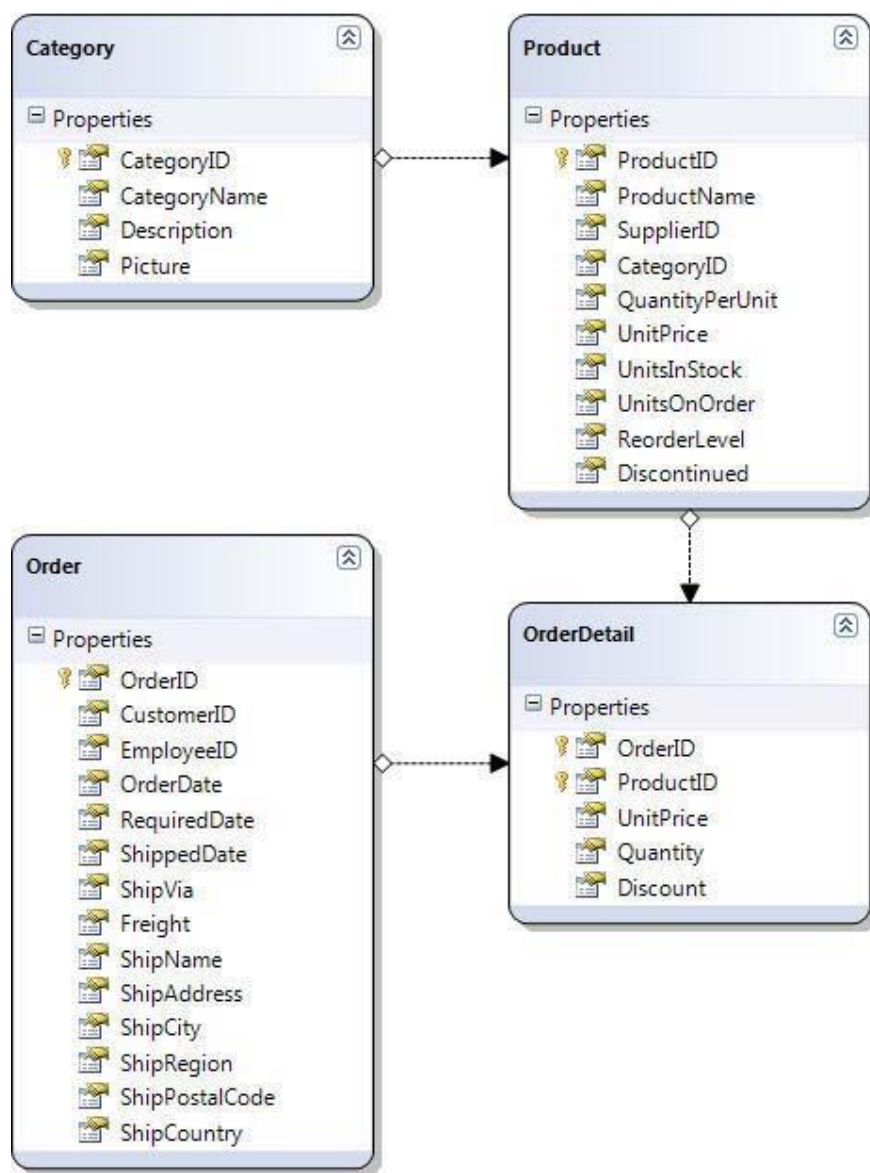


# Using LINQ in C# 1

## استفاده از LINQ در C# بخش ۱

ویرایش: شهریور ۱۳۸۸

YousefAmiri.ir



## معرفی

ارتباط با پایگاه داده و انجام اعمال ثبت، ویرایش، حذف و جستجو در آن، همواره از خسته کننده ترین مراحل کار برنامه نویسان است. با وجود پیشرفتهای چشمگیر در ارتقاء IDE ها، و بکارگیری رابطهای مختلف مانند ADO، DAO، ADO.NET همچنان این مرحله از کار برنامه نویسی، خالی از لذت بود. تا اینکه مایکروسافت سرانجام در .NET 3.0. نگاه شیء گرای خود را با LINQ به تکامل تحسین برانگیزی رساند.

### LINQ بطور ساده

قبل از یادگیری، مهم است از ابتدا بدانیم چه چیزی را یاد می گیریم و آیا نیازی به یادگیری آن هست؟ LINQ مدل مرسوم ارتباط با بانک اطلاعاتی را تغییر می دهد:

- دیگر خبری از دستورات SQL و ADO.NET که در گوشه و کنار برنامه، با سورس C# ترکیب شده بودند تا تراکنشی را با بانک اطلاعاتی انجام دهند، نخواهد بود. و سورس برنامه بسیار کوتاهتر و خواناتر خواهد بود
- لازم نیست هر بار سری به جداول، View و یا Stored Procedure ها بزنید تا ببینید فیلدها به چه نام، ترتیب و یا از چه نوعی هستند!
- رهایی از اشیاء پردردسر ADO.NET که مثلاً هنگام درگ کردن یک جدول از DataSet بر روی فرم ایجاد می شدند
- رهایی از مشکلات بسیاری که هنگام تغییر دادن یک جدول در بانک اطلاعاتی بوجود می آمد. در این صورت باید دوباره DataSet را به روز می کردید و Query های TableAdapter را از اول می نوشتید
- GridView روی فرم از کار می افتاد یا دچار انواع خطاها می شد.
- ...

### مقایسه دستور Insert در ADO.NET و LINQ

#### ADO.NET

```
//using System.Data;

//using System.Data.SqlClient;

Cn = new SqlConnection(@"Server=.\SQLEXPRESS;AttachDbFilename=" + Application.StartupPath +
@"\Database.mdf;Integrated Security=True;User Instance=True");

Cn.Open();

/*****
SqlCommand Cm = new SqlCommand("", Cn);

Cm.CommandText = "INSERT INTO tbPerson(Code, Name, Family, Sex, Nation)
VALUES (@Code,@Name,@Family,@Sex,@Nation);

Cm.Parameters.AddWithValue("@Code",lbCode.Text);
Cm.Parameters.AddWithValue("@Name", txName.Text);
Cm.Parameters.AddWithValue("@Family", txFamily.Text);
Cm.Parameters.AddWithValue("@Sex", cbSex.Text);
Cm.Parameters.AddWithValue("@Nation", cbNation.Text);

Cm.ExecuteNonQuery();
```

#### LINQ

```
//using System.Linq;
var db = new LINQDataContext();
db.tbPerson_Insert(lbCode.Text, txName.Text, txFamily.Text, cbSex.Text, cbNation.Text);
```

۱. در Visual Studio گزینه Project > Add New Item... را انتخاب کنید
۲. در پنجره باز شده گزینه LINQ to SQL Classes را انتخاب کنید. در قسمت Name نام LINQ.dbml را وارد کنید.
۳. پنجره ای با دو قسمت در پروژه شما ایجاد می شود. این همان محیط اشیاء LINQ است. پنجره سمت چپ برای درگ کردن جداول و پنجره سمت راست برای درگ کردن پروسیجرهای ذخیره شده بانک اطلاعاتی می باشد.
۴. از پانل View > Server Explorer بانک اطلاعاتی را باز کنید و جداول و پروسیجرهای ذخیره شده آنرا به داخل محیط LINQ بکشید.
۵. حالا وقت کد نویسی است؛ با فرض داشتن جدول زیر در بانک اطلاعاتی و گرایی به نام gvPerson بر روی فرم:

| tbPerson |              |
|----------|--------------|
| Code     | Char(10)     |
| Name     | Nvarchar(20) |
| Family   | Nvarchar(30) |
| Sex      | Char(1)      |
| Nation   | Nvarchar(30) |

دستور زیر کلیه داده های جدول را در گراید می ریزد:

```
var db = new LINQDataContext();
gvPerson.DataSource = db.tbPersons;
```

اجرای یک پروسیجر ذخیره شده دارای پارامتر (در اینجا رکوردی از جدول حذف می شود و گراید به روز می شود)

```
var db = new LINQDataContext();
db.tbPersons_Delete(txCode.Text);
gvPerson.DataSource = db.tbPersons;
```

درج یک رکورد در جدول و به روز رسانی بانک اطلاعاتی توسط پروسیجر ذخیره شده

```
var db = new LINQDataContext();
db.tbPerson_Insert(lbCode.Text, txName.Text, txFamily.Text, cbSex.Text, cbNation.Text);
gvPerson.DataSource = db.tbPersons;
```

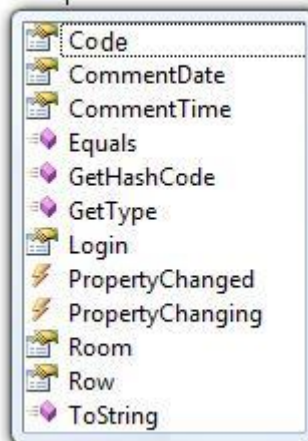
## عبارات لامبدا

LINQ کلمات کلیدی SQL را بصورت اشیاء و متدها در خود دارد. معنی این حرف این است که دیگر نیازی به حفظ کردن زبان SQL نیست و می توان از دستورات SQL همانند دستورات C# استفاده کرد. پیش زمینه این امکان فوق العاده مهیج، آشنایی با عبارات لامبداست. هدف از نوشتن عبارات لامبدا، ارتباط برقرار کردن بین زبان C# و زبان SQL به شیوه ای مختصر، مفید و قوی است. راههای مختلفی برای نگارش عبارات لامبدا وجود دارد که در اینجا از کوتاهترین و سریعترشکل دستورات استفاده می شود. کافیت بدانید ما نیازمند متغیری هستیم که در عبارت Where یا Select یا OrderBy و ... توسط آن به فیلدهای جدول دسترسی داشته باشیم. این متغیر بصورت بسیار مختصر و به شکل زیر تعریف می شود:

c=>

تعجب نکنید! از این به بعد شما متغیری به نام c دارید که می توانید به فیلدها دسترسی داشته باشید. کافیت پس از آن یک نقطه بنزید تا لیست فیلدها، متدها و اشیاء را ببینید.

```
db.tbComments.Where(c=>c.
```



## اجرای شرط WHERE

```
var db = new LINQDataContext();
int count = db.tbPersons.Where(c=> c.Grade == cbGrade.Text).Count();
```

## انتخاب یک فیلد و نمایش آن در کومبو باکس

```
var db = new LINQDataContext();
cbNation.DataSource = db.tbPersons.Select(c=> c.Nation);
```

## انتخاب ملیت Iran یا France

```
var db = new LINQDataContext();
cbFamily.DataSource = db.tbPersons.Where(c=> c.Nation=="Iran" ||
                                             c.Nation=="France").Select(c=> c.Family);
```

