بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

# آموزش LINQ در SQL

مدرس : مهندس افشین رفوآ

رمز فایل :tahlildadeh.com



## مقدمه

شرکت ماکروسافت، Language Integrated Query یا LINQ را همراه با NET Framework 3.5. معرفی کرد LINQ .، برنامه نویسان را قادر به queryکردن منبع داده ها با استفاده از یک query مانند syntax با C# و VB.NET می کند. این منبع داده ها می توانند collection ها، بانک های اطلاعاتی SQL Server ، XML ، و dataset ها باشند. به غیر از آنچه که از طرف Microsoft تامین می شود، LINQ گستردنی هم هست. این بدین معناست که شما می توانید منابع داده ها را فراتر از آنچه که مایکروسافت ship میکند، queryکنید. مثال هایی از چنین پیاده سازی هایی عبارتند از LINQ To Flickr ، LINQ To Amazon، LINQ to Google، و غیره. در این مقاله نشان می دهم چگونه می توان از LINQ To SQL جهت اجرای عملیات های CRUD روی یک بانک اطلاعاتی SQL Server استفاده کرد. من از بانک اطلاعاتی Northwind استفاده می کنم و یک برنامه ASP.NET می سازم تا قابلیت های LINQ To SQL را نشان دهم. شما می توانید بانک اطلاعاتی Northwind را از

http://www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46A0-8DA2-EEBC53A68034&displaylang=en

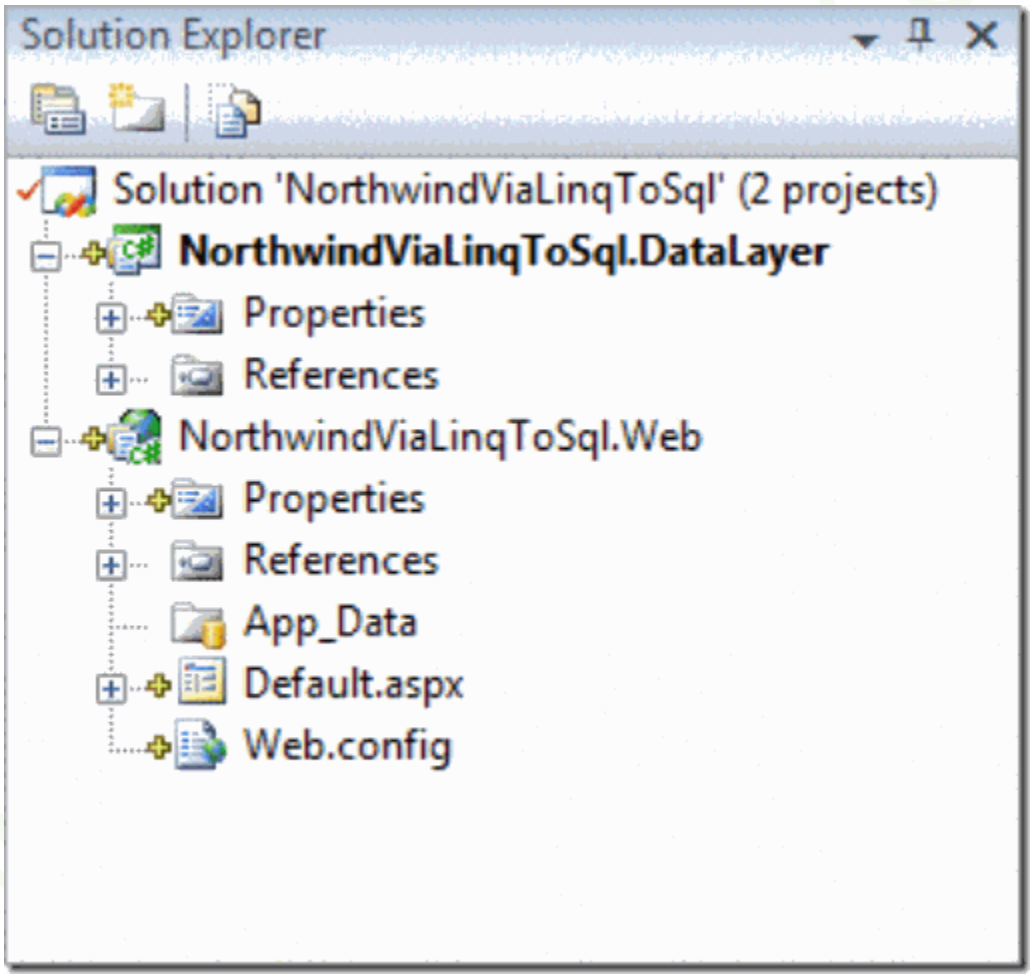
دانلود کنید.

## ابزار های لازم برای این مقاله:

۱. Visual Studio 2008
۲. .NET Framework 3.5
۳. SQL Server 2005

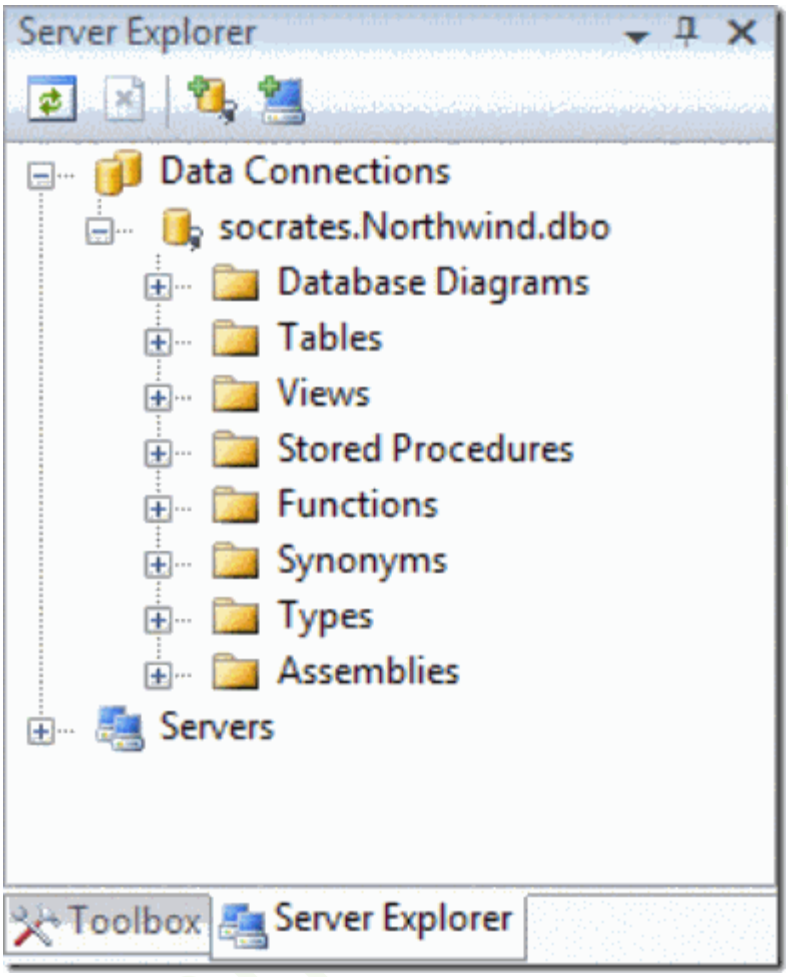
## ساختار راه حل(Solution Structure)

برای این مقاله، به دو پروژه نیاز داریم. یکی data layer است که generateخواهیم کرد، و دیگری یک برنامه تحت وب ASP.NET است. ساختار راه حل درSolution Explorer ، شبیه مثال زیر است.

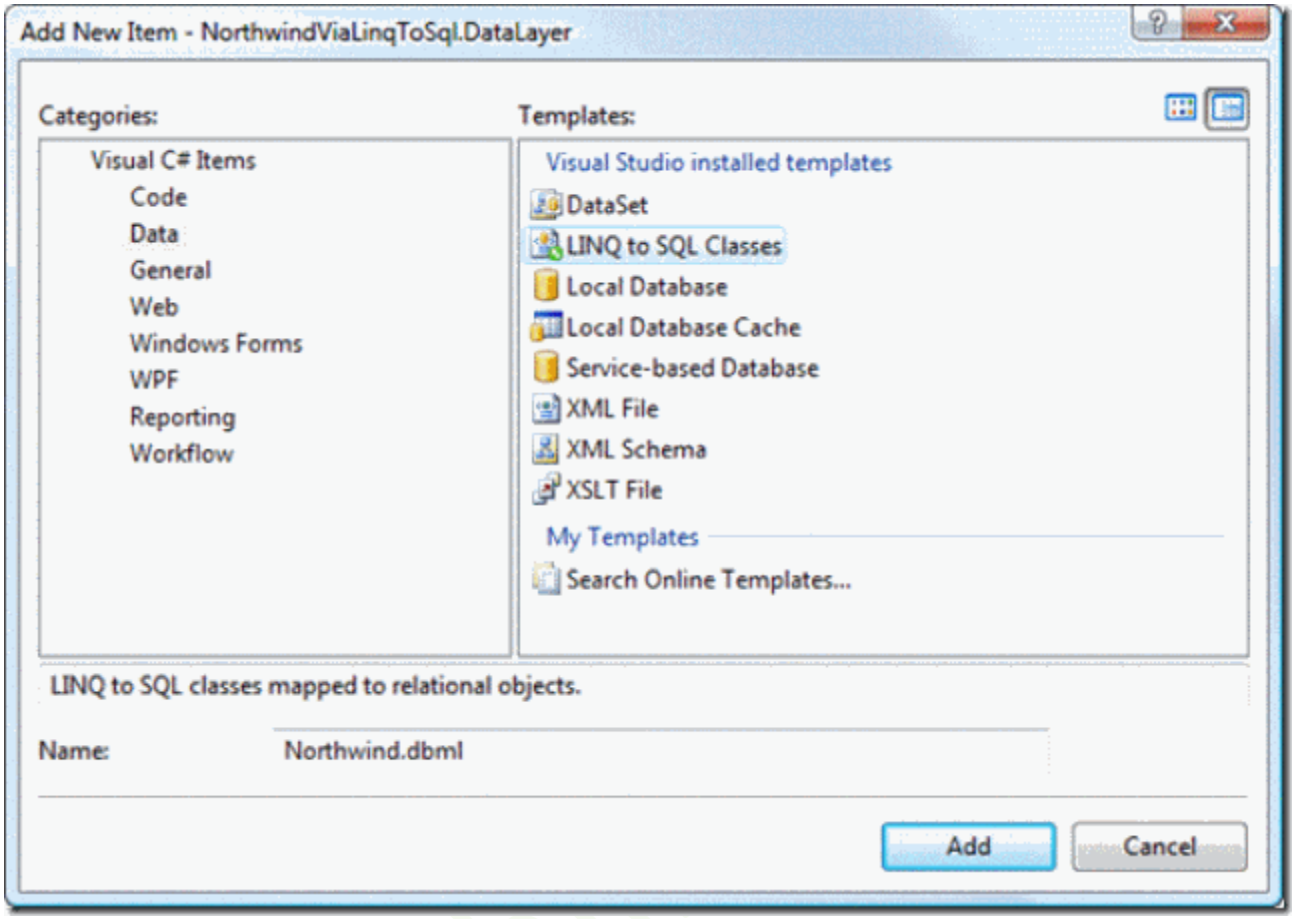



## ایجادData Layer

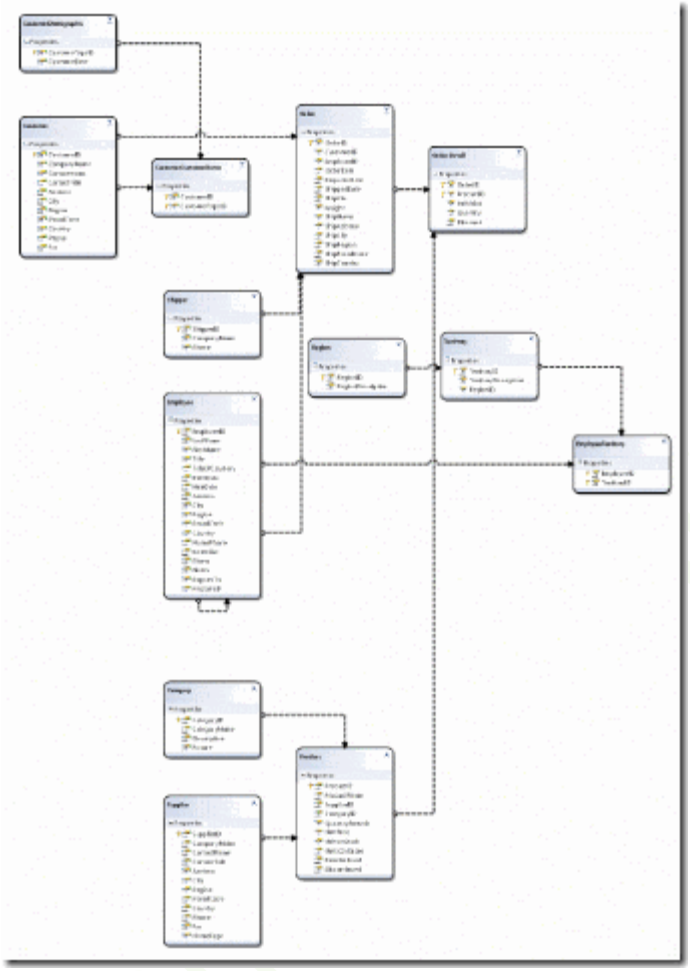
قبل از اینکه data layer را generate کنیم، باید یک connection در Server Explorerایجاد کنیم که به بانک اطلاعاتی Northwind اشاره می کند.



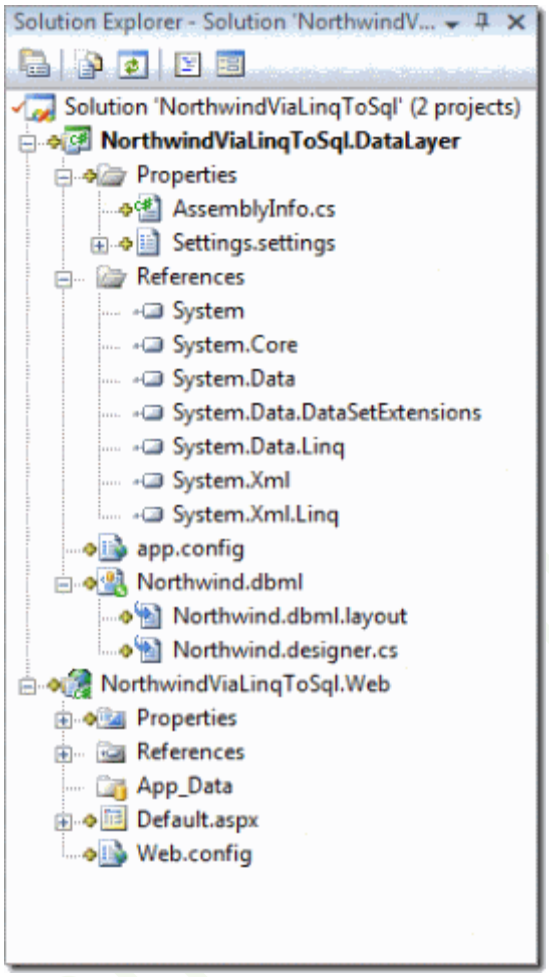

حالاdata layer مان را با استفاده LINQ To SQL ایجاد می کنیم. برای انجام این کار، باید یک item جدید به پروژه data layer از نوع کلاسهای LINQ to SQL اضافه کنیم. نام آن را Northwind می گذاریم، همانگونه که در زیر نشان داده شده.




بعد از اضافه کردن یک کلاسLINQ to SQL ، با یک designer surface روبرو می شویم. دراینجا تنها جداول را drag می کنیم که بخشی از data layer خواهد شد. در این مقاله، ما همه جداول را از طریق انتخاب کردن همه آنها در یک حرکت، رویdesigner ، dragمی کنیم Designer . ما باید بعد از drag کردن همه جداول روی آن شبیه شکل زیر باشد.



حالا باید solution را بسازیم تا مطمئن شویم همه چیز ok است. ما با موفقیت data layer مان را نصب کرده ایم. در Solution Explorer می بینم که دو فایل جدید داریم، یعنی Northwind.dbml.layout و Northwind.designer.cs. همچنین می بینیم که reference های لازم برای complie و run کردن کدمان نیز توسط Visual Studio اضافه شده اند.




فایل cs. حاوی کدی برای data layer ماست. بیایید کدی را که برای ما ایجاد شده، امتحان کنیم. نگاهی به کلاس Region خواهیم انداخت.

```
[Table(Name="dbo.Region")]
public partial class Region : INotifyPropertyChanging, INotifyPropertyChanged
```

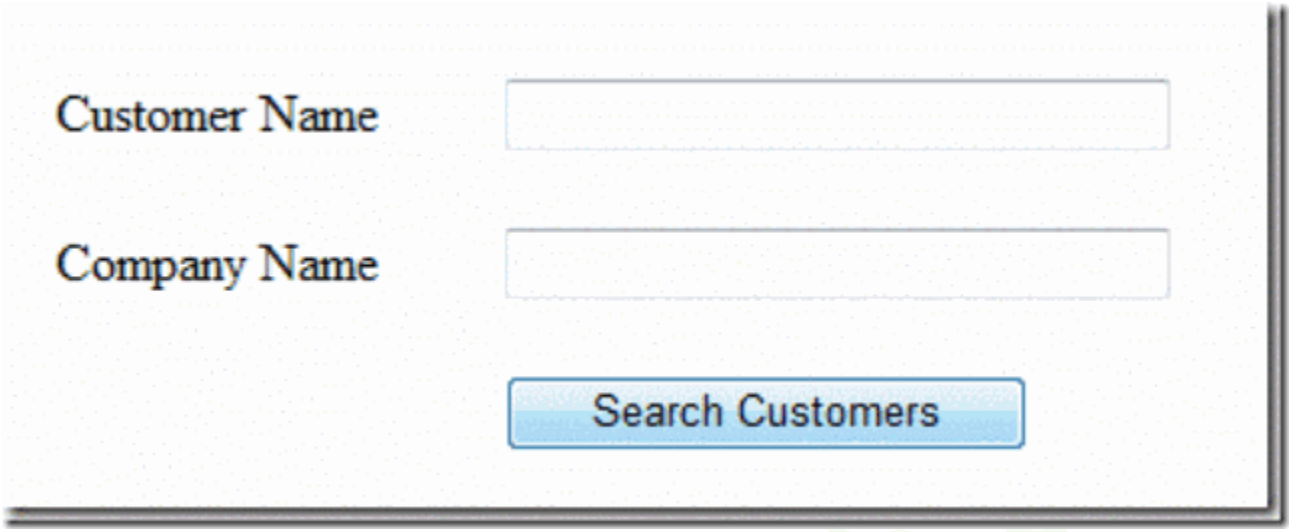
این کلاس با صفتTable ، آمیخته شده و برای خصوصیتName ، نام واقعی جدولی که در بانک اطلاعاتی مان موجود است، تخصیص داده شده. کلاس Region نیزinterface هایINotifyPropertyChanging و INotifyPropertyChangedرا پیاده سازی می کند. اینinterface ها برای databinding بکار می روند. کلاس Region همچنین حاوی یک خصوصیت(property) برای هر ستون است. بیایید نگاهی به خصوصیت RegionDescriptionبیاندازیم.

```
[Column(Storage="_RegionDescription", DbType="NChar(50) NOT NULL",
CanBeNull=false)]
public string RegionDescription
{
  get
  {
    return this._RegionDescription;
  }
  set
  {
    if ((this._RegionDescription != value))
    {
      this.OnRegionDescriptionChanging(value);
      this.SendPropertyChanging();
      this._RegionDescription = value;
      this.SendPropertyChanged("RegionDescription");
      this.OnRegionDescriptionChanged();
    }
  }
}
```

ستون ها با صفت Column آمیخته شده اند وvalue ها برای storasge ارسال می شوند، DbTypeو CanBeNullنشان می دهند آیا ستون null است یا خیر.

## استفاده ازData Layer

حالا که data layer را ایجاد کرده ایم، روی برنامه های تحت وب ASP.NET کار خواهیم کرد، یعنی جاییکه data layerرا خواهیم دید. در ابتدا یک web form ایجاد می کنیم تا مشتری ها را جستجو کنیم و نتایج را نمایش دهیم. همچنین یک web form ایجاد می کنیم تا مشتری های جدید را وارد کنیم. بیایید با web form برای جستجوی مشتری ها شروع کنیم؛ برای این کار، از صفحه Default.aspx استفاده خواهیم کرد. تعداد کمی کنترل روی web form قرارمی دهیم. این کنترل ها، پارامترهای جستجو و یک دکمه در اختیار ما می گذارد که وقتی رویش کلیک کنیم، جستجو را انجام می دهند و نتایج را نمایش می دهد. شکل زیر نمایی از فرم بعد از قرار دادن کنترل هاست:




همچنین یک کنترل GridView روی فرم مان قرار می دهیم تا نتایج جستجو را نمایش دهد. حالا مقداری کد در event handlerکلیک دکمه قرار می دهیم تا جستجو را انجام دهد و نتایج را در GridView نمایش دهد. باید مطمئن شوید که یک reference به پروژهData Layer ، System.Data.Linqو عبارت مناسب وجود دارد. در زیر آنچه که event handler کلیک دکمه دربرخواهد گرفت، آورده شده است:

```
protected void buttonSearch_Click(object sender, EventArgs e)
{
  using (NorthwindDataContext context = new NorthwindDataContext())
  {
    var customers =
      from c in context.Customers
      select c;
    gridViewCustomers.DataSource = customers;
    gridViewCustomers.DataBind();
  }
}
```

این کد، جدول مشتریان را در بانک اطلاعاتیnorthwind ، queryمی کند و همه مشتری ها را باز می گرداند. حالا باید کمی آن را modify کنیم تا نام مشتری ها و نام شرکت ها را به عنوان پارانترهایی برایquery مان قبول کند event handler . بعد از modify کردن به صورت زیر خواهد بود:

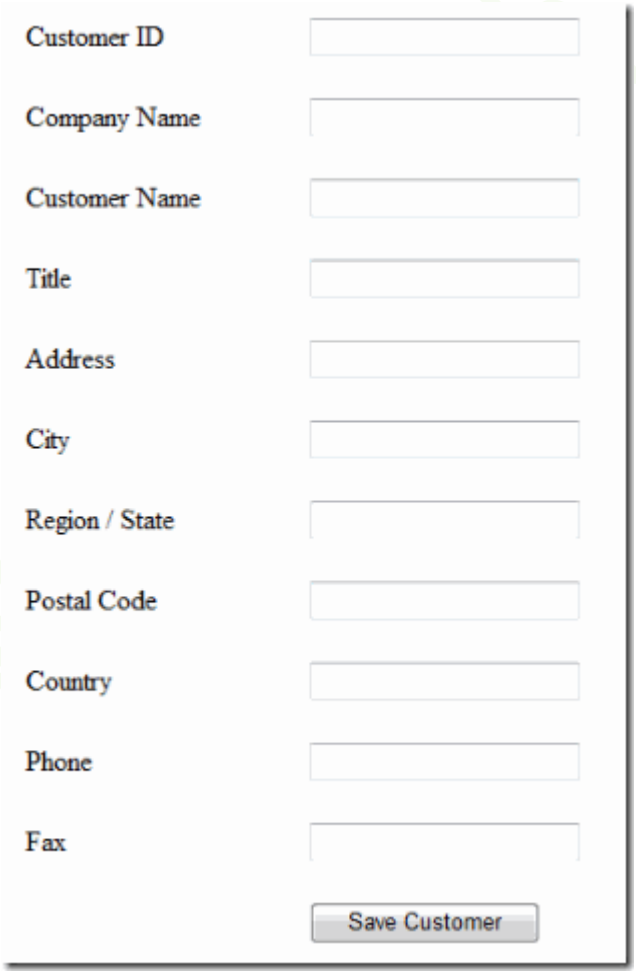
```
protected void buttonSearch_Click(object sender, EventArgs e)

{
  using (NorthwindDataContext context = new NorthwindDataContext())
  {
    var customers =
      from c in context.Customers
      where (
```

```
        c.ContactName.Contains(textBoxCustomerName.Text.Trim())
        &&
        c.CompanyName.Contains(textBoxCompanyName.Text.Trim()))
      select c;
      gridViewCustomers.DataSource = customers;
      gridViewCustomers.DataBind();
  }
}
```

حالا نتایج جستجو فیلتر می شود.

حالا بیایید یک فرم data entry برای مشتری ها ایجاد کنیم. باید یک web form جدید در پروژه ASP.NET مان وارد کنیم و آن را CustomerEntry بنامیم. برای شروع باید مطمئن شویم که فرم ما حاوی field های لازم برای وارد کردن یک مشتری جدید است. فرم ما بعد از تکمیل شبیه نمونه شکل یر خواهد بود.




انتظار ما این است که هنگامی که روی دکمه Save Customer کلیک می کنیم، یک ردیف (row) جدید به جدول مشتری ها اضافه شود. کد زیر این کار را برای ما انجام می دهد.

```
protected void buttonSave_Click(object sender, EventArgs e)
{
  using (NorthwindDataContext context = new NorthwindDataContext())
  {
    Customer customer = new Customer
    {
      CustomerID = textBoxCustomerID.Text,
      CompanyName = textBoxCompanyName.Text,
      ContactName = textBoxCustomerName.Text,
      ContactTitle = textBoxTitle.Text,
      Address = textBoxAddress.Text,
      City = textBoxCity.Text,
      Region = textBoxRegion.Text,
      PostalCode = textBoxPostalCode.Text,
      Country = textBoxCountry.Text,
      Phone = textBoxPhone.Text,
      Fax = textBoxFax.Text
    };
    context.Customers.InsertOnSubmit(customer);
    context.SubmitChanges();
  }
}
```

می توان یک ستون که از قبل در بانک اطلاعاتی موجود است را ابتدا توسط بازیابی داده ها و سپس توسط submit کردن آن از طریقDataContext ، آپدیت کرد.

## نتیجه گیری

در این مقاله، ما فقط یک عبارت SQL مستقل ننوشتیم تا داده ها را به یک بانک اطلاعاتی وارد کنیم یا آن را بازیابی کنیم. این، زیبایی LINQ To SQL است. به علاوه، کد بازیابی ما هنگامی که در C# است بسیار شبیه یک query است. باید از مزایای چنین روش موثر و یکپارچه ای برای کار کردن با داده ها قدردانی کرد.