

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

آموزش مقدماتی LINQ to SQL

این جزوه به این امید تهیه شده است که بتواند در ارتقای سطح علمی
دوستان موثر واقع شود.

از خواننده محترم تقاضا می کنم با ارسال نظرات و پیشنهادات خود در
مورد این جزوه و یا اشکالات موجود در محتوی آن به آدرس

در بهبود آن سهیم باشد Mr.Gharahi@gmail.com

با تشکر

مهدی قرهی

پاییز ۱۳۸۷

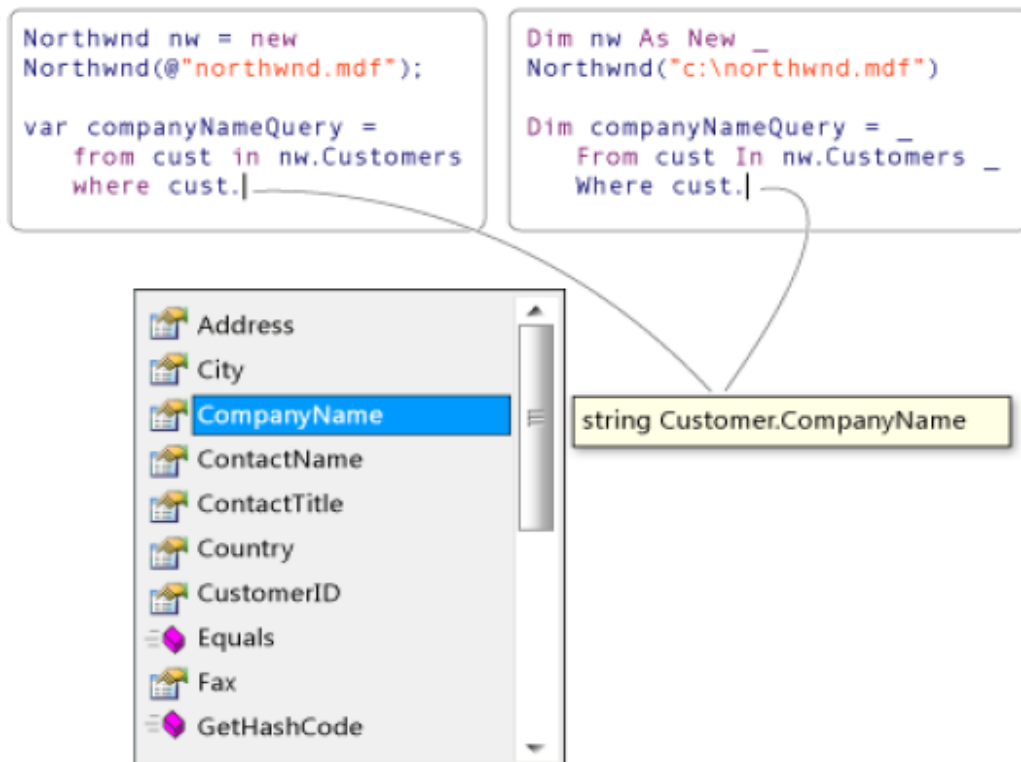
LINQ چیست ؟

(LINQ(Language-INtegrated Query) یک حرکت جدید در

Visual Studio 2008 و .NET Framework 3.5 است که شکاف

بین دنیای اشیا و دنیای داده ها را پر میکند. این بدان معناست که مثلا برای کار با بانک اطلاعاتی شما می توانید با آن همانند یک کلاس در برنامه خود رفتار می کنید و مسلم است کار بسیار راحت تر خواهد شد.

به شکل زیر توجه کنید :



Address , City , CompanyName , ...فیلدهایی از بانک هستند که

توسط LINQ to SQL داخل کلاسی از برنامه در دسترس می باشند

LINQ تنها برای کار با بانک اطلاعاتی نیست و در مواردی چون کار با
سندهای XML (LINQ to XML), کار با اشیا (LINQ to Objects)
و ... کاربرد دارد.

LINQ to SQL

همان طور که قبلاً توضیح داده شد LINQ to SQL فرایندی جدید برای کار با داده های ذخیره شده توسط SQL می باشد می خواهیم برنامه ای بنویسیم تا به کمک LINQ داده های بانکی با فیلد های زیر را پردازش کنیم :

Column Name	Data Type	Length	Allow Nulls
ID	numeric	9	
FName	varchar	50	✓
LName	varchar	50	✓
Age	smallint	2	✓

پروژه جدیدی ایجاد کنید.

برای ادامه باید یک LINQ to SQL Class به پروژه اضافه کنید. پس در

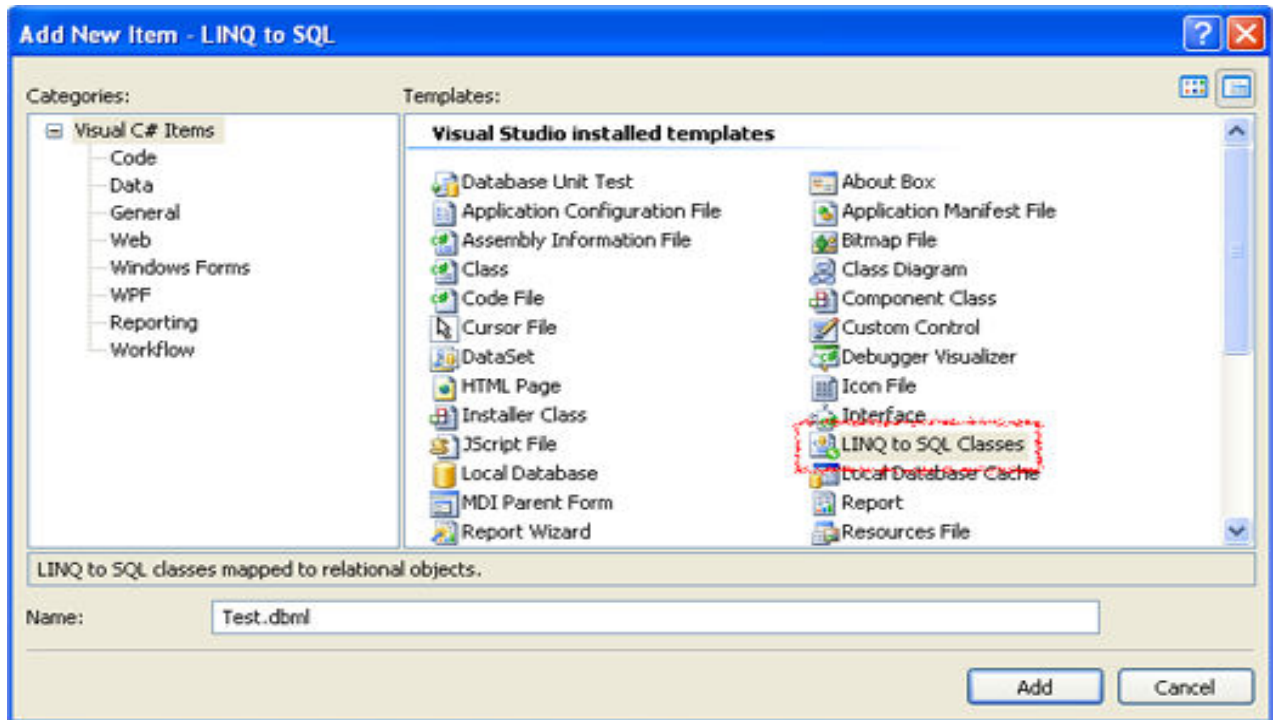
Solution Explore بر نام پروژه کلیک راست کرده و

در زیر منوی Add گزینه New Item را انتخاب کنید

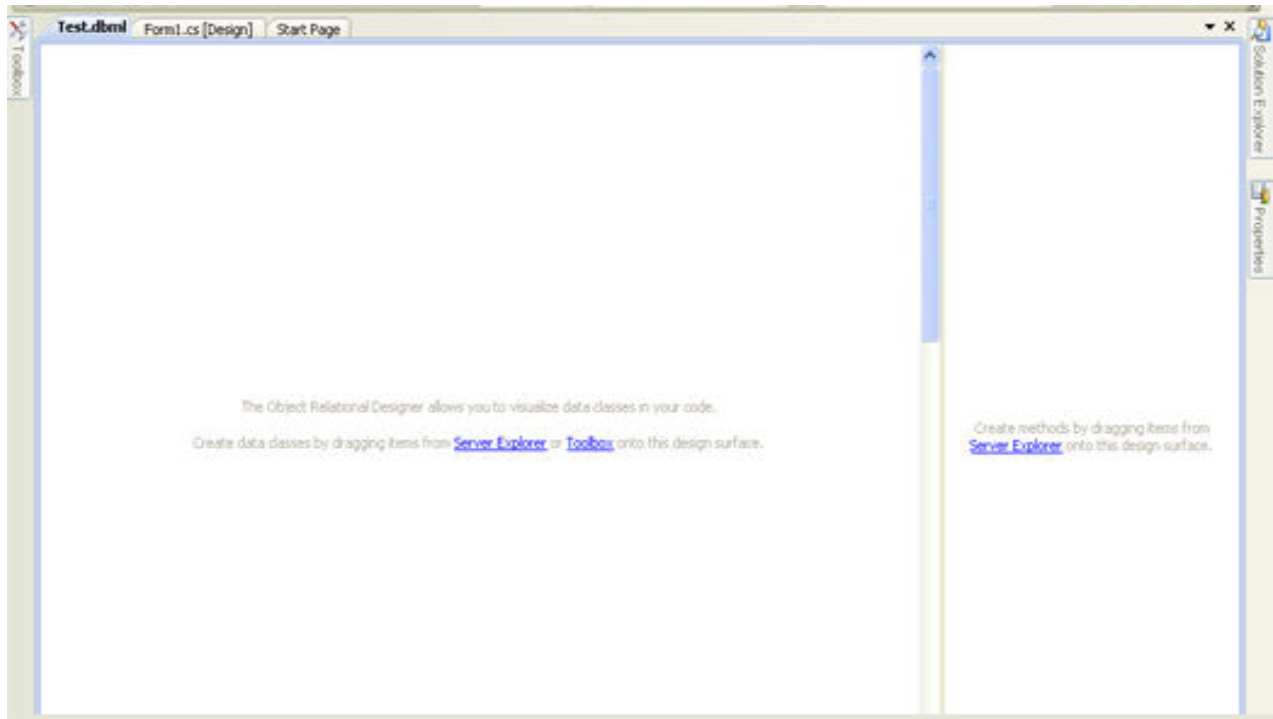
یک LINQ to SQL Classes انتخاب کرده و نام آن را Test بگذارید

(شکل 1)

Test.dbml مانند شکل 2 ظاهر می شود

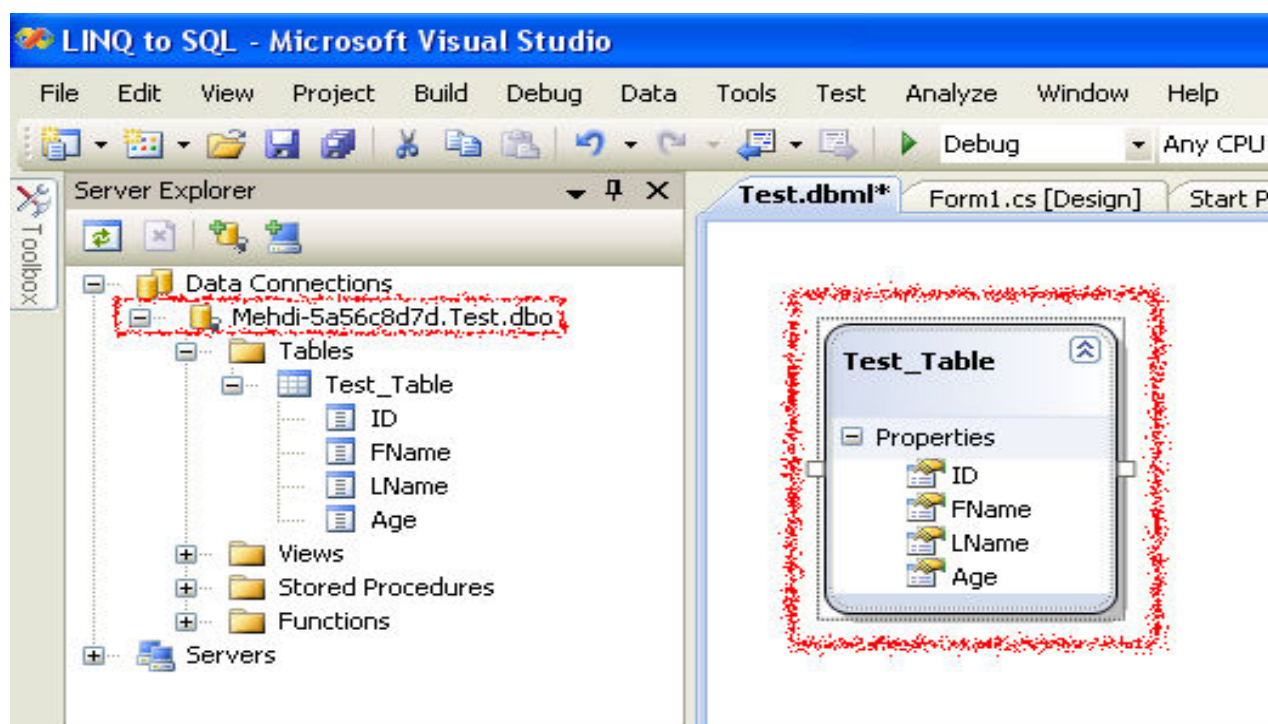


شکل 1



شکل 2

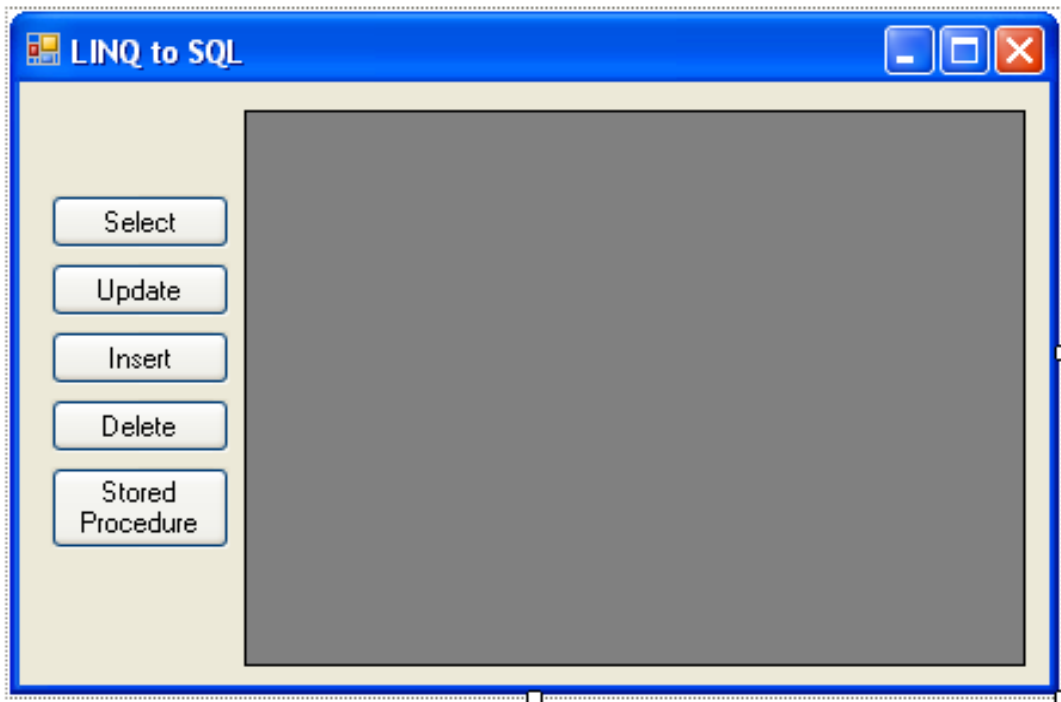
روی Server Explorer کلیک کنید.
در قسمت اضافه شده در سمت چپ بانک خود را انتخاب کنید
در این مثال من بانک Test را انتخاب میکنم
با Drag & Drop جدول ها را در ناحیه خالی وسط پنجره اضافه می کنم
(شکل 3)



شکل 3

حالا وقتش رسیده که یه کمی هم کد بنویسیم
توی فرم برنامه 5 تا Button بزارید برای Select, Delete

Update, Insert و Stored Procedure
و یک dataGridView برای نمایش نتیجه.



: Select

وقتی بانک رو به برنامه اضافه می کنید یه کلاس به برنامهتون اضافه می شه به نام `DataContext` که نماینده بانکه. به ازای هر جدول یه کلاس با نام جدول ایجاد می شه. برای `Select` اول باید یه نمونه از کلاس بانک بسازیم تا `Query` روی اون اجرا بشه.

```
DataContext Bank=new DataContext ();
```

حالا باید یه `Query` ایجاد کنیم :


```
var Query = from T in Bank.Test_Tables
            where T.Age > 10
            select T;
```

کلمه کلیدی Var اجازه میدهد به متغیر بدون تعیین نوع اون تعریف کنیم
حالا کافیه این Query رو به Datasource نسبت بدیم

```
dataGridView1.DataSource = Query;
```

```
TestDataContext Bank=new TestDataContext();
var Query = from T in Bank.Test_Tables
            where T.Age > 10
            select T;
```

```
dataGridView1.DataSource = Query;
```

: Update

برای Update هم باید Query بنویسیم :

```
TestDataContext Bank=new TestDataContext();
int SelectedId = 5;
var Query = (from P in Bank.Test_Tables
            where P.ID == SelectedId
            select P).Single();
Query.FName = "Change";
Query.LName = "Change";
```

```
Query.Age = 0;
```

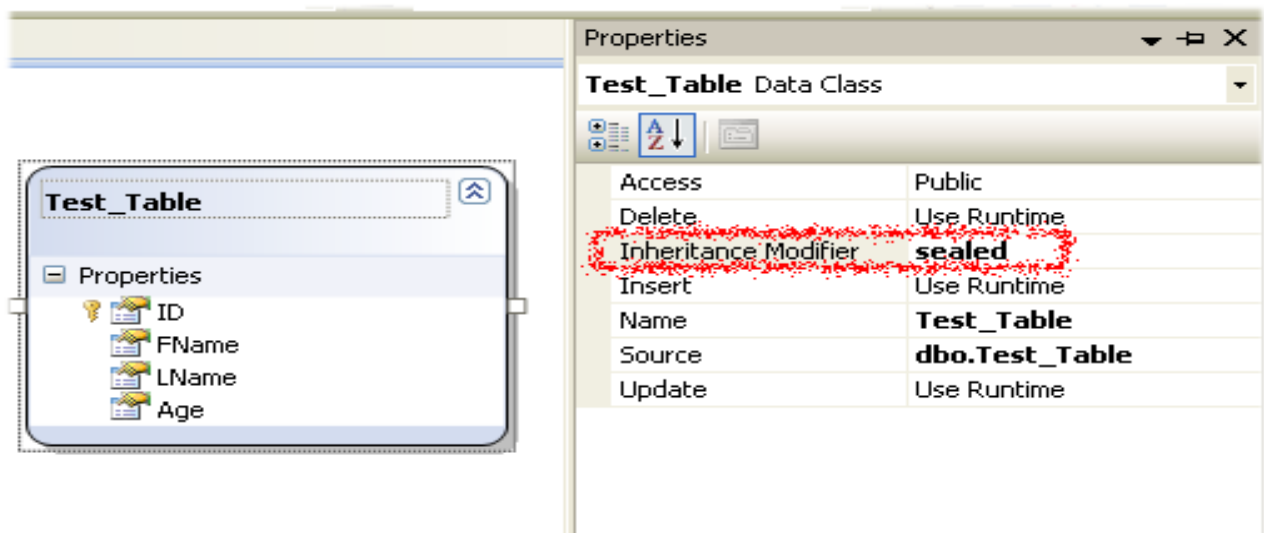
```
Bank.SubmitChanges();
```

متد Single() تضمین میکند نتیجه Query فقط یک خط رو شامل میشه

متد SubmitChanges() هم تغییرات رو تو بانک اعمال میکند

نکته : برای اینکه جدول قابل تغییر باشه باید Inheritance Modifier

رو به Sealed تغییر بدیم



: Insert

```
TestDataContext Bank = new TestDataContext();  
Test_Table test_table = new Test_Table();
```

```
test_table.Age = 22;  
test_table.FName = "X";  
test_table.LName = "XXXX";
```

```
Bank.Test_Tables.InsertOnSubmit(test_table);  
Bank.SubmitChanges();
```

: Delete

```
TestDataContext Bank = new ApplicationDbContext();
```

```
var Query = (from P in Bank.Test_Tables  
             where P.ID == 5  
             select P).Single();
```

```
Test_Table TT = new Test_Table();
```

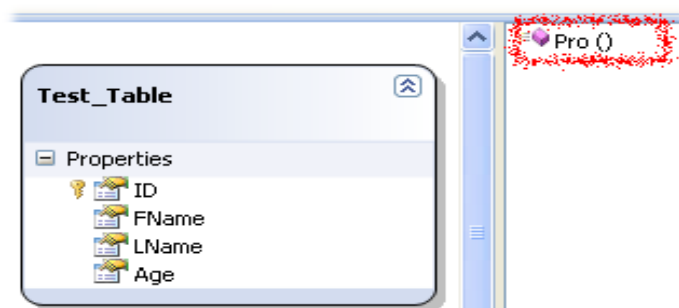
```
Bank.Test_Tables.DeleteOnSubmit(Query);
```

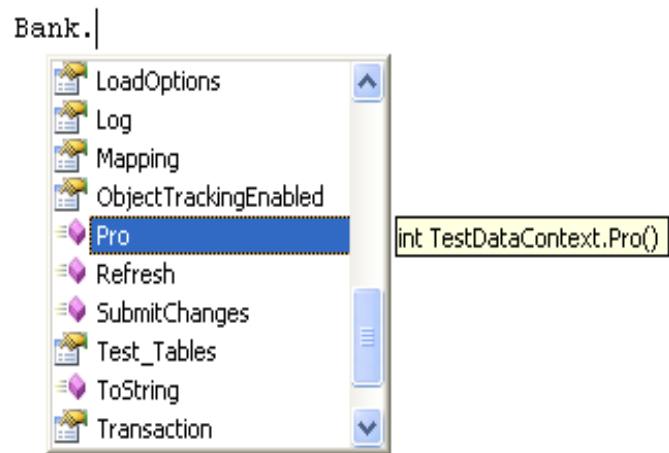
```
Bank.SubmitChanges();
```

: Stored Procedure

برای اجرای یک پروسیجر ذخیره شده باید اونو مثل یه جدول به Test.dbml اضافه کنیم. حالا ما پروسیجر رو مثل یه متد داخل کلاس داریم.

من پروسیجر رو به اسم Pro میزنم :





```
TestDataContext Bank = new TestDataContext ();  
Bank.Pro ();
```

موفق باشید