



Linq To SQL

تکنولوژی Linq To SQL

در زبان برنامه نویسی C#.Net

نویسنده: میثم پاسداری هریس

سخنی با شما

مقاله حاضر که در اختیارتان قرار گرفته است سعی کرده که تکنولوژی لینک (Linq) را معرفی کند و همچنین به بررسی بخش ارتباط با پایگاه داده آن پرداخته است. نویسنده نیز سعی بر این دارد که مطالب را با شکل ها و تصاویر گرفته شده از محیط برنامه نویسی معرفی کند که این ویژگی موجب می شود خیلی راحت این مطلب رو یاد بگیرید. بدون شک هیچ اثر عاری از اشکال نخواهد بود بنابراین از شما خواننده گرامی خواهشمندم نظرات و پیشنهادات خود را عنوان نمایید تا در مقاله های بعدی ترتیب اثر داده شود.

با تشکر میثم پاسداری هریس – دانشجوی نرم افزار

Pasdari.Meysam@yahoo.com

تقدیم به پدر و مادرم که بزرگترین نعمت زندگی ام هستند.

معرفی

LINQ مختصر عبارت Language Integrated Query که یک زبان انعطاف پذیر برای پرس و جو می باشد این تکنولوژی در سال ۲۰۰۵ معرفی شد و در نسخه ۳.۵ .NetFramework (Visual Studio 2008) به کار برده شد. در واقع این زبان برای تقاضا از هر گونه مجموعه داده (بانک اطلاعاتی، آرایه‌ها، Xml و...) می باشد و ترکیب نحوی آن بسیار شبیه زبان SQL بوده و با زبان‌های برنامه نویسی C# و Vb.NET قابل پیاده سازی است.

هدف اصلی خلق LINQ ارائه یک زبان یکتا جهت تقاضا، برنامه ریزی و فیلتر کردن اطلاعات ذخیره شده در اشیاء مختلف اعم از پایگاه داده، اشیاء، آرایه‌ها و از همه مهتر XML است، این زبان قابلیت Debug شدن به صورت Runtime را داراست و بسیار قابل انعطاف است.

انواع LINQ

به وسیله Linq می توان بر روی مجموعه ای پرس و جو انجام داد و یا عناصر مجموعه ای را مدیریت کرد. خصوصیت کلیدی لینک (Linq) این است که این زبان با زبان های برنامه نویسی دیگر ادغام شده است و در انواع زبان های برنامه سازی دارای یک ساختار مشخص و ثابت است بنابراین این ویژگی کلیدی به برنامه نویسان این امکان را می دهد که بتوانند در چندین زبان برنامه نویسی به راحتی از این تکنولوژی استفاده کنند.

تیم توسعه دهنده .Net. با معرفی لینک زیر ساخت هایی را نیز برای کار با انواع موجودیت ها طراحی کرد که به برنامه نویسی امکان کار بر روی داده های از انواع مختلف را می دهد که مهمترین این انواع موارد زیر می باشد :

۱) Linq to Object

۲) Linq to XML

۳) Linq to Sql

و....

که در این مقاله به بررسی Linq to Sql می پردازیم.

Linq To Sql

لینک به Sql نیز سیستمی است که امکان کار بر روی جداول ، Stored Procedure ها و .. برای کاربر امکان پذیر می کند. علاوه بر این لینک به توسعه دهندگان این امکان را فراهم می کنند که بتوانند به سادگی و با دستوراتی مشابه زبان Sql به پایگاه داده دسترسی داشته و اعمالی را (پرس و جو ها ، درج ، حذف و ...) بر روی آن انجام دهد.

شاید تصور کنید که این سیستم جایگزینی برای ADO.Net است اما این تعریف درستی از لینک به Sql نمی باشد بلکه این سیستم در کنار ADO.net امکانات جدیدی را در اختیار توسعه دهندگان قرار می دهد.



معرفی لینک تا این حد برای بحث ما کافی می باشد. حال به کدنویسی می پردازیم. قبل از اینکه بخواهیم به طور عملی با ابزار

Linq to Sql Classes در ویژوال استادیو (ترجیحا ۲۰۰۸ به بعد) کار کنیم چندین دستور SQL و معادل آن ها در Linq را بررسی کنیم.

جدول زیر را در نظر بگیرید. (نام جدول : Student)

	Column Name	Data Type	Allow Nulls
	ID	int	<input type="checkbox"/>
	Fname	varchar(50)	<input checked="" type="checkbox"/>
	Lname	varchar(50)	<input checked="" type="checkbox"/>
	avg	float	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

کلمه کلیدی : var

از این کلمه کلیدی جهت معرفی متغیری که نوع آن از قبل مشخص و معلوم نیست استفاده می کنیم و ویژوال استادیو پس از انتساب مقداری در آن متغیر نوع آن را تشخیص می کند. به مثال های زیر توجه کنید.

```
var anInteger = 5;  
var aString = "a string";  
var unknown_type = new MyClass();
```

دستورات SQL و معادل Linq آنها

```
Select * From Student
```

دستور Select

```
Var Query=From P in StudentDataContext.Student
```

```
Select p;
```

معادل Linq

```
Select Fname,Lname From Student
```

دستور Select چند ستونی

معادل Sql

```
Var Query=From P in StudentDataContext.Student
```

```
Select new{P. Fname , P.Lname}
```

معادل Linq

```
Select * From Student
```

```
Where ID=1
```

دستور Where

معادل Sql

```
Var Query=From P in StudentDataContext.Student
```

```
Where P.ID==1
```

```
Select p;
```

معادل Linq

```
Update Student Set Fname="Ali"
```

```
where ID=2
```

دستور Update

معادل Sql

```
Var Query=(From P in StudentDataContext.Student
```

```
Where P.ID=2
```

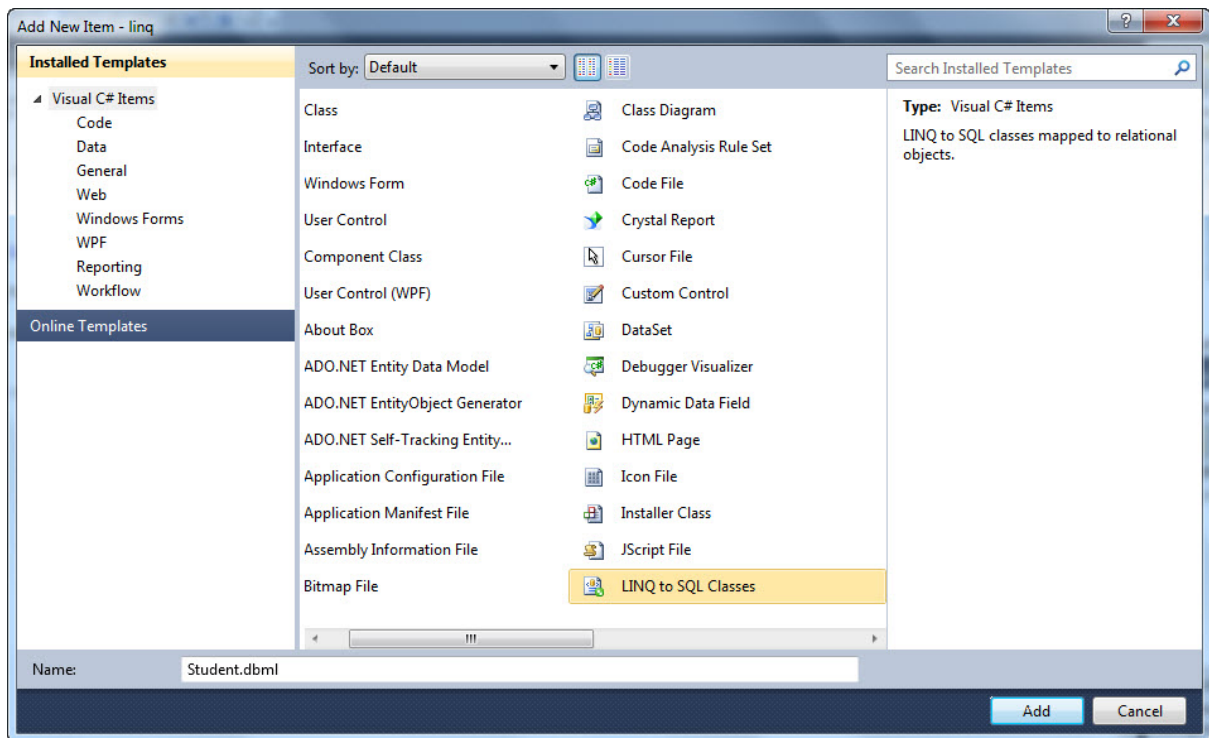
```
Select p).Single();
```

```
Query.Fname="Ali";
```

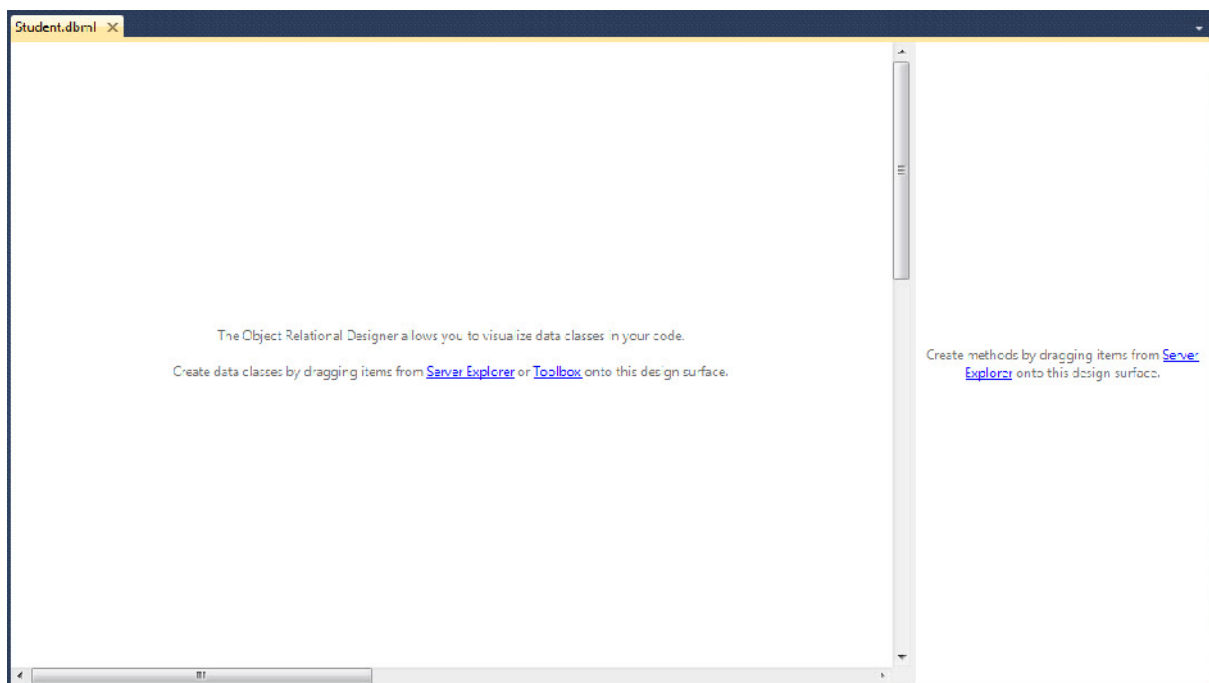
معادل Linq

با دقت در مثال های بالا می توان فهمید که ترکیب نحوی (Syntax) این دو زبان چقدر به یکدیگر نزدیک اند.

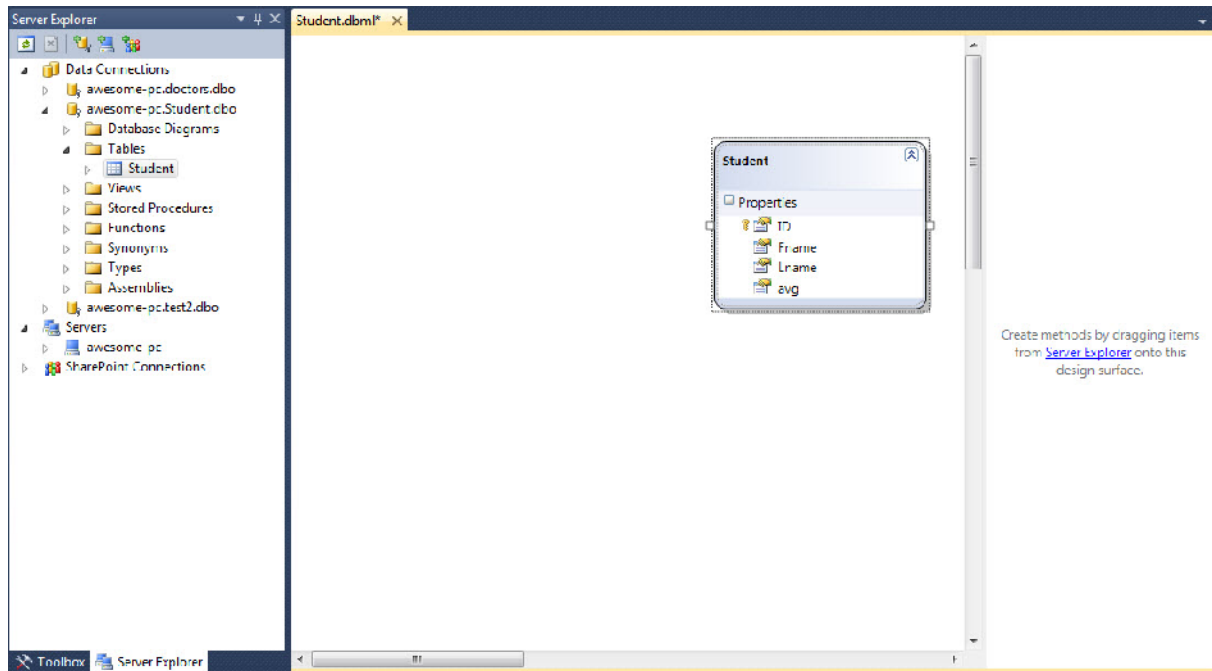
برای یک کار عملی پروژه ای WindowsFormApplication تحت نام Test_Linq ایجاد کنید. سپس در بخش Solution Explorer بر روی نام پروژه (Test_Linq) کلیک راست نموده و از طریق گزینه New item ، مطابق شکل زیر آیتم جدیدی از نوع Linq to Sql classes تحت نام Student.dbml به پروژه خود اضافه کنید.



پس از اضافه کردن آیتم جدید شکل زیر ظاهر خواهد شد.



در این قسمت مطابق شکل زیر شما باید از طریق منوی **Server Explorer** در سمت چپ پایگاه ای که قبلا ایجاد کرده اید را انتخاب و جدول **Student** را **Drag & Drop** کرده به محیط شکل بالا اضافه کنید.



پس از اضافه نمودن جدول به پروژه، دو کلاس یکی برای پایگاه داده و یکی هم برای جدول ها اضافه می شود. در این مثال کلاس هایی با نام های زیر به پروژه اضافه می شوند.

StudentDataContext کلاس مربوط به پایگاه داده

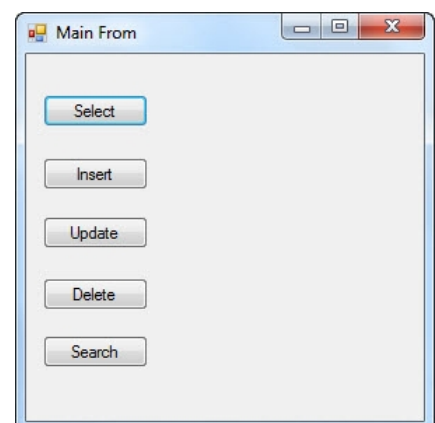
Student کلاس مربوط به جدول

حال فرم هایی با اسامی (frmInsert,frmUpdate,frmSelect,frmDelete,frmSearch) به پروژه خود اضافه کنید.

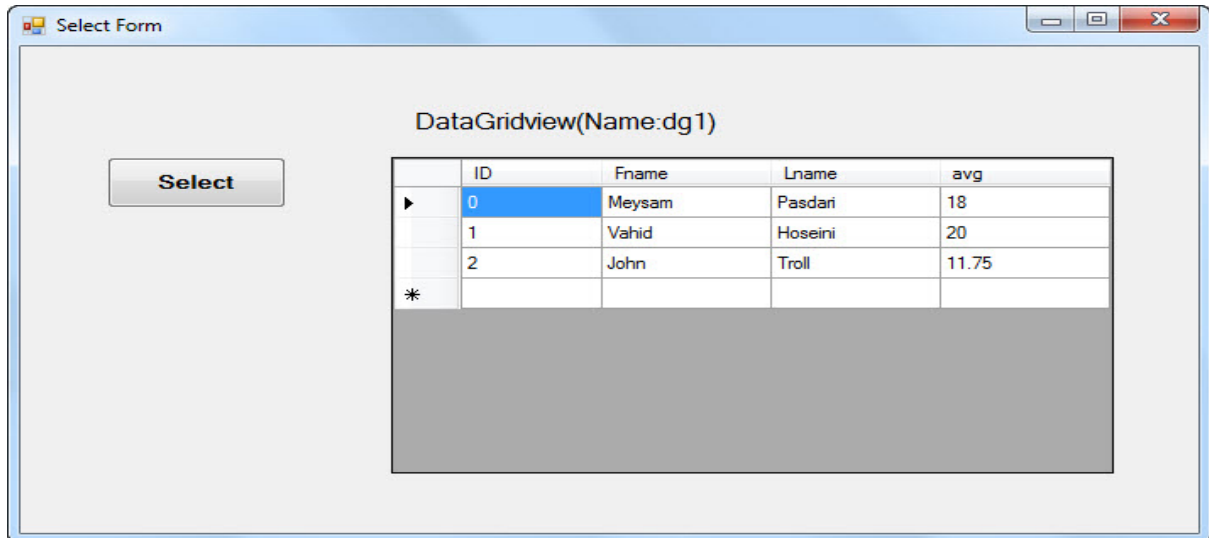
در هر فرم نام ابزار ها به شرح زیر است:

- txtid : ID جعبه متن
- txtfname : FName جعبه متن
- txtlname : Lname جعبه متن
- txtavg : Avg جعبه متن
- dg1 : دیتا گرید

یک فرم دیگر (Main Form) نیز به شکل زیر برای اجرای دیگر فرم های برنامه ایجاد کنید.



فرم Select را مطابق شکل طراحی کنید.



ID	Fname	Lname	avg
0	Meysam	Pasdari	18
1	Vahid	Hoseini	20
2	John	Troll	11.75
*			

دستورات این فرم را در رویداد کلیک دکمه Select بنویسید.

```
private void btnselect_Click(object sender, EventArgs e)
{
    StudentDataContext bank = new StudentDataContext();

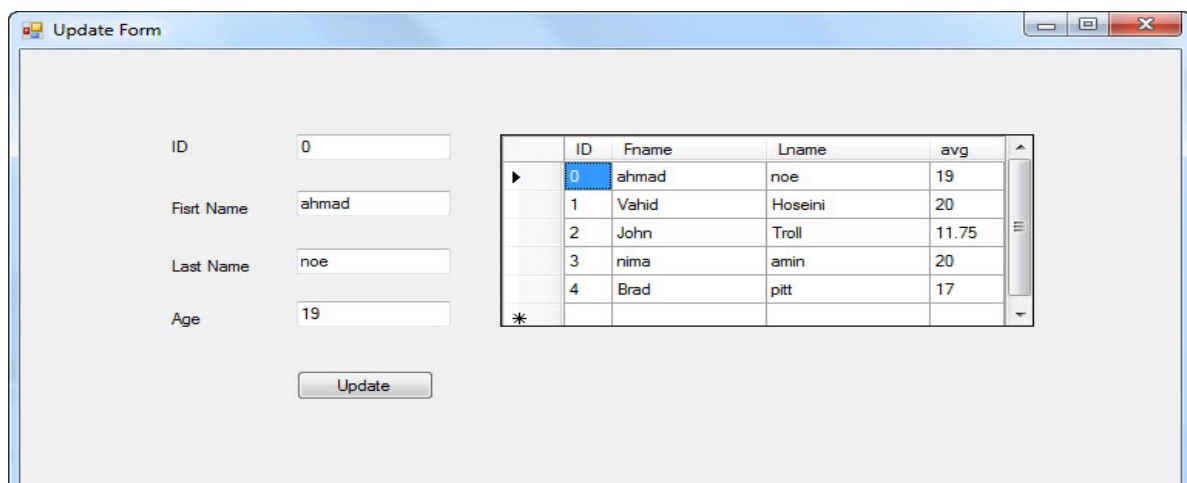
    var query = from P in bank.Students
                select P;

    dg1.DataSource = query; //Show Result Of Query
}
```

توضیح دستورات :

ابتدا یک Object از کلاس StudentDataContext ایجاد گردید، سپس از طریق دستور Select کلیه فیلدهای جدول را فراخوانی شده و در خاصیت DataSource ابزار DataGridView قرار داده گرفت.

فرم Update



ID	Fname	Lname	avg
0	ahmad	noe	19
1	Vahid	Hoseini	20
2	John	Troll	11.75
3	nima	amin	20
4	Brad	pitt	17
*			

دستورات این فرم را در رویداد کلیک مربوط به دکمه Update بنویسید.

```
private void btnupdate_Click(object sender, EventArgs e)
{
    StudentDataContext bank = new StudentDataContext();

    var query = (from P in bank.Students
                where P.ID == int.Parse(txtid.Text)
                select P).Single();
    query.Fname = txtfname.Text;
    query.Lname = txtlname.Text;
    query.avg = Convert.ToInt16(txtage.Text);
    dg1.DataSource = query; //Show Table Data In DataGridView
    bank.SubmitChanges(); // Store Resault Of Operations

    var q = from p in bank.Students
            select p;
    dg1.DataSource = q;
}
```

توضیح دستورات :

در ابتدا همانند مثال قبلی یک Object از کلاس StudentDataContext ایجاد گردید سپس دستور Update (جستجو بر اساس فیلد ID) مطابق شکل بالا نوشته شد و خروجی پرس و جو در متغیر query قرار گرفت.

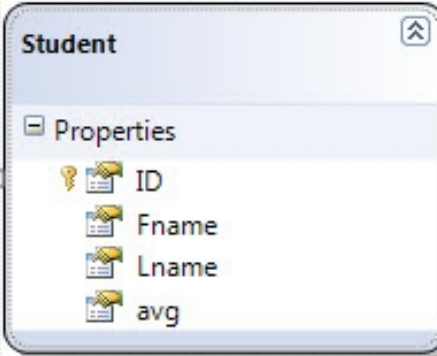
در انتها نیز یک دستور Select جهت نمایش داده ها در جدول نوشته شد.

Single(): مشخص می کند که خروجی query فقط یک خط را شامل می شود.

bank.SubmitChanges(): جهت ذخیره تغییرات اعمال شده در فیلد ها مورد استفاده قرار گرفته است.

تذکر : برای قابل تغییر بودن جدول مقدار مشخصه Inheritance Modifier را برابر Sealed قرار دهید.(مطابق شکل زیر)

Access	Public
Delete	Use Runtime
Inheritance Modifier	sealed
Insert	Use Runtime
Name	Student
Source	dbo.Student
Update	Use Runtime



فرم Insert (درج مقادیری در جدول)

ID	Fname	Lname	avg
0	ahmad	noe	19
1	Vahid	Hoseini	20
2	John	Troll	11.75
3	nima	amin	20
4	akbar	nourani	19.75

دستورات این فرم را در رویداد کلیک مربوط به دکمه Insert بنویسید.

```
private void btnupdate_Click(object sender, EventArgs e)
{
    StudentDataContext bank = new StudentDataContext();
    Student u = new Student();

    u.ID = int.Parse(txtid.Text);
    u.Fname = txtfname.Text;
    u.Lname = txtlname.Text;
    u.avg = float.Parse(txtage.Text);
    bank.Students.InsertOnSubmit(u);
    bank.SubmitChanges();
    idnum();
    var q = from p in bank.Students
            select p;
    dg1.DataSource = q;
}

public void idnum()
{
    int c1 = 0;
    StudentDataContext bank = new StudentDataContext();

    var query = from P in bank.Students
                select new { c1 = P.ID };

    txtid.Text = (query.Max(P => P.c1+1)).ToString();
}

private void frminsert_Load(object sender, EventArgs e)
{
    idnum();
}
```

توضیح دستورات :

در این قسمت علاوه بر تعریف Object از کلاس StudentDataContext ، Object دیگری (u) از کلاس Student ساخته شد.

بدلیل اینکه در این دستور همانند مثال های قبل پرس و جو (query) نداریم از یک شی از کلاس Student برای دستیابی به فیلدها استفاده شده است. پس از انتساب مقادیر به فیلدها توسط دستور bank.Students.InsertOnSubmit(u); مقادیر شی u در جدول قرار گیرد و با اجرای دستور bank.SubmitChanges(); تغییرات ایجاد شده در جدول اعمال می شود. اما متد دیگری نیز بنام idnum() نوشته شده است در واقع وظیفه اصلی این تابع این است که با توجه به متد Max() مربوط شی کوئری بزرگترین مقدار فیلد ID را واگشی کرده و یک واحد به آن اضافه می نماید. بنابراین در این قسمت مقدار فیلد ID به صورت خودکار داده می شود. (و خاصیت ReadOnly جعبه متن نیز مقدار True گرفته تا از تغییر توسط کاربر و بروز خطا جلوگیری شود)

فرم Delete

ID	Fname	Lname
0	ahmad	noe
1	Vahid	Hoseini
2	John	Troll
3	nima	amin
4	akbar	nourani

دستورات این فرم را در رویداد کلیک مربوط به دکمه Delete بنویسید.

```
private void btnupdate_Click(object sender, EventArgs e)
{
    try{
        StudentDataContext bank = new StudentDataContext();
        var Query = (from P in bank.Students
                    where P.ID == int.Parse(txtid.Text)
                    select P).Single();
        bank.Students.DeleteOnSubmit(Query);
        bank.SubmitChanges();
        MessageBox.Show("Selected Item Deleted");
        //refresh The Datagrid Value
        var query = from P in bank.Students
                   select P;
        dg1.DataSource = query;
        var query = from P in bank.Students
                   select P;
        dg1.DataSource = query;
    }
    catch(Exception ex)
    {
        MessageBox.Show (ex.Message) ;
    }
}
```

توضیح دستورات :

در این تابع با استفاده از دستور `Select` یک کوئری بر اساس مقدار فیلد `ID` صورت گرفته تا نتایج محدود شود. سپس با اجرای دستور

`bank.Students.DeleteOnSubmit(Query)` رکورد مورد نظر حذف شده و پس از اجرای دستور

`bank.SubmitChanges();` تغییرات انجام شده اعمال می گردد.

فرم `Search` (جستجو)

	ID	Fname	Lname	avg
▶	1	Vahid	Hoseini	20
*				

دستورات این فرم را در رویداد کلیک مربوط به دکمه `Search` بنویسید.

```
private void btnupdate_Click(object sender, EventArgs e)
{
    StudentDataContext bank = new StudentDataContext();

    var query = from P in bank.Students
                where P.ID == int.Parse(txtid.Text)
                select P;
    dg1.DataSource = query;
}
```

توضیح دستورات :

در این قسمت با استفاده از دستور `Select` و یک عبارت شرطی `Where` خروجی پرس و جو به `ID` مورد نظر محدود شده است.

