

Print in C#

چاپ در C# توسط Microsoft Report

ویرایش: مرداد ۱۳۹۷

YousefAmiri.ir



در این مقاله، با مفاهیم زیر آشنا می شوید:

ایجاد Relation بین جداول

ایجاد Diagram در SQL Server

استفاده از LINQ در گزارشات

ایجاد گزارشات rdlc

چاپ توسط Microsoft Report

ایجاد Data Source برای استفاده در گزارش

تعیین خروجی برای BindingSource توسط LINQ

پیکربندی Microsoft Report Viewer

قراردادن تصویر و جدول در گزارش

معرفی

همیشه طراحی گزارشات کاری وقت گیر بوده است. بیشترین زمان این کار مربوط به چیدن فیلدها روی فرم گزارش و اجراهای متوالی برنامه است تا به خروجی مورد نظر برسیم. اما این به آن معنی نیست که شما با پروسه پیچیده ای سروکار دارید که یادگیری آن وقت زیادی بخواهد.

پیش از شروع باید بدانید Crystal Report تنها انتخاب برای گزارشگیری نیست. بلکه ابزار مهیج تر با ظاهر ساده اما فوق العاده قوی در .NET وجود دارد که کار با آن لذت بخش است. با Report Viewer کاربران نرم افزار شما خروجیهای چاپ، PDF و اکسل خواهند داشت و نیازی به کدنویسی برای قالب خروجی گزارش خود ندارید. از مزایای دیگر Microsoft Report این است که گزارشات طراحی شده در C# به راحتی در ASP.NET نیز قابل استفاده هستند و نیازی به کار اضافه نیست. این مزیت خوبی برای برنامه نویسانی است که روی سیستمهای متمرکز کار می کنند و همزمان خروجی نرم افزار را روی وب یا پورتال داخلی قرار می دهند.

پروژه Library

برای شروع، یک بانک اطلاعاتی به نام Library با دو جدول که با همدیگر ارتباط دارند، ایجاد می کنیم و سپس از اطلاعات گزارشگیری می کنیم. هدف این پروژه آموزش نحوه گزارشگیری است پس در مورد ثبت و حذف داده ها صحبت نخواهیم کرد.

ایجاد جدولها

	Column Name	Data Type	Allow Nulls
PK	NationalCode	varchar(15)	<input type="checkbox"/>
	Name	nvarchar(20)	<input checked="" type="checkbox"/>
	Family	nvarchar(30)	<input checked="" type="checkbox"/>
	Birthday	char(10)	<input checked="" type="checkbox"/>

شکل ۱- جدول tStudent

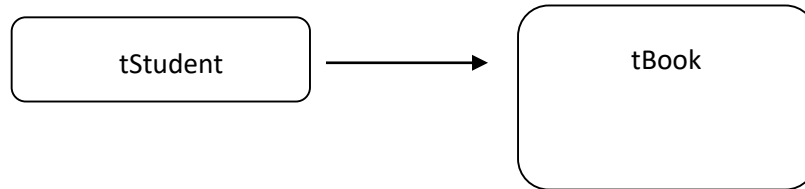
	Column Name	Data Type	Allow Nulls
PK	ISBN	varchar(15)	<input type="checkbox"/>
	NationalCode	varchar(15)	<input checked="" type="checkbox"/>
	Title	nvarchar(50)	<input checked="" type="checkbox"/>
	Date	char(10)	<input checked="" type="checkbox"/>

شکل ۲ - جدول tBook

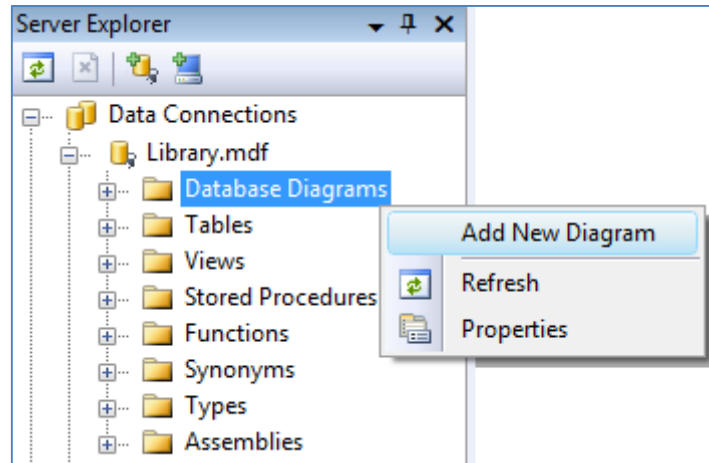
ارتباط بین جدولها

جدول tStudent نام دانشجویان و جدول tBook لیست کتابهای امانت گرفته شده را برای دانشجویان نشان می دهد. برای تمرین بیشتر می توانید مباحث نرمال سازی را روی این دو جدول و جدول سومی به نام tBookList پیاده سازی کنید.

هر دانشجو (tStudent) می تواند چند کتاب (tBook) تحویل بگیرد. پس ارتباط جدولها بصورت ۱ به چند خواهد بود:



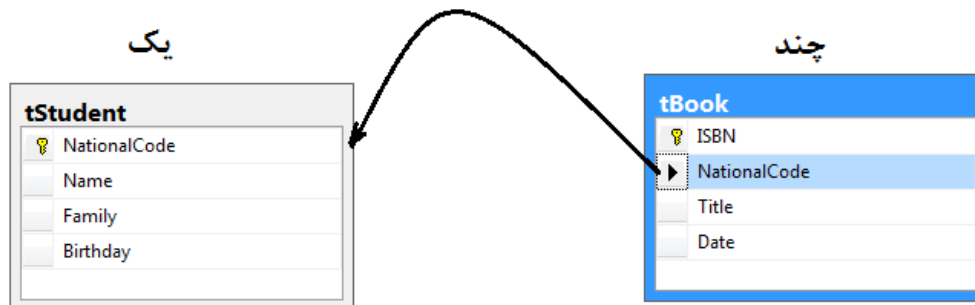
برای پیاده سازی این ارتباط، در ویژوال استودیو و در پنل Server Explorer بانک اطلاعاتی را باز کنید و مراحل را زیر را دنبال کنید:



شکل ۳ - ایجاد دیاگرام برای پیاده سازی ارتباط (Relation) بین جداول

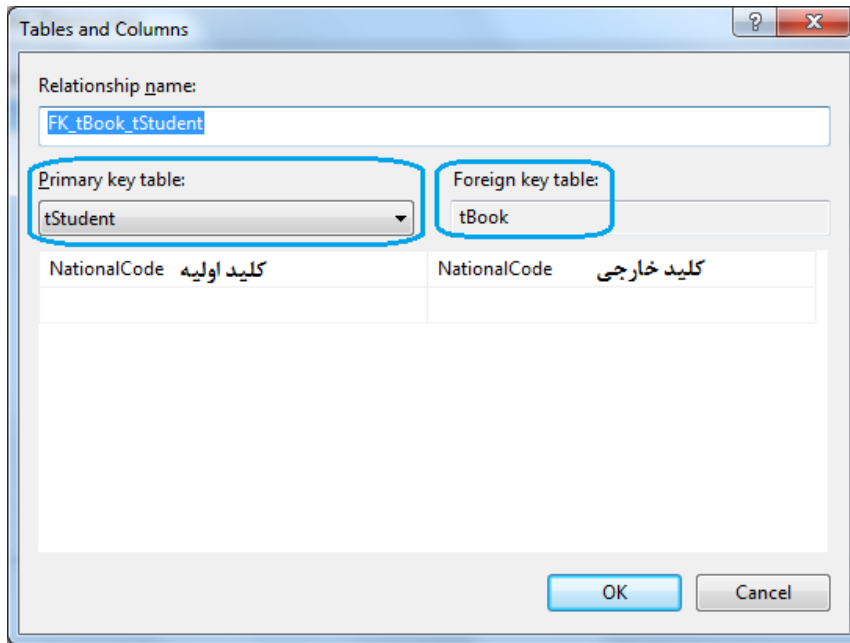
جداول را به محیط دیاگرام اضافه کنید و فیلد NationalCode را از جدول tBook بر روی فیلد NationalCode از جدول tStudent بکشید. تا یک ارتباط یک به چند ایجاد شود.

کشیدن کلید خارجی بر روی کلید اولیه



شکل ۴ - ایجاد ارتباط یک به چند بین دو جدول

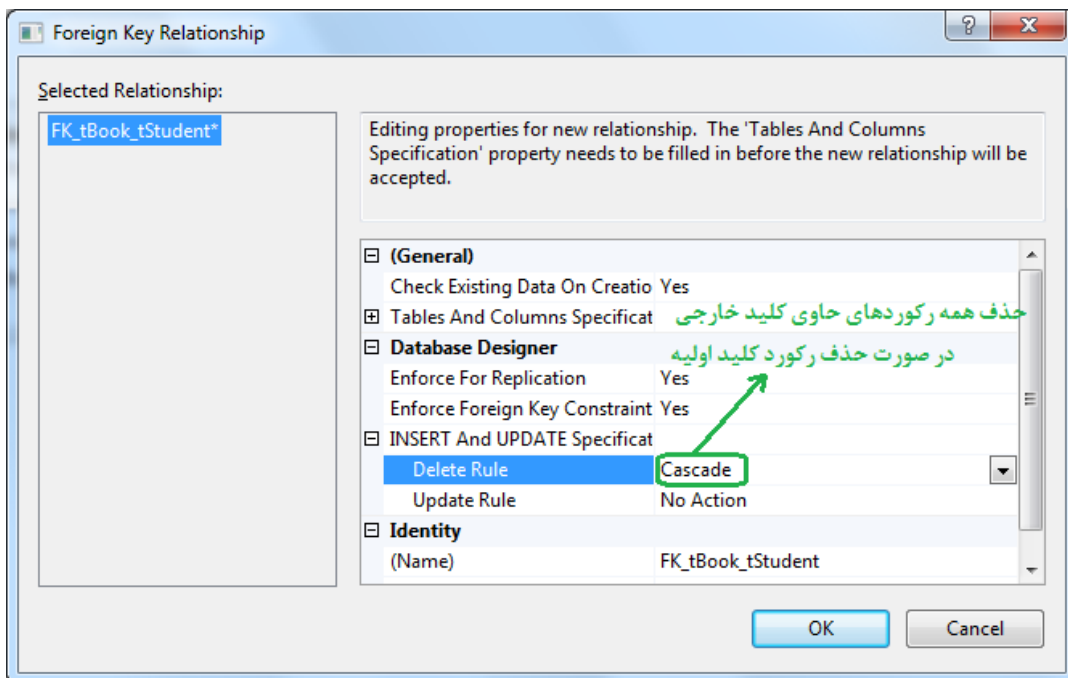
برای ایجاد ارتباط، دکمه Ok را بزنید:



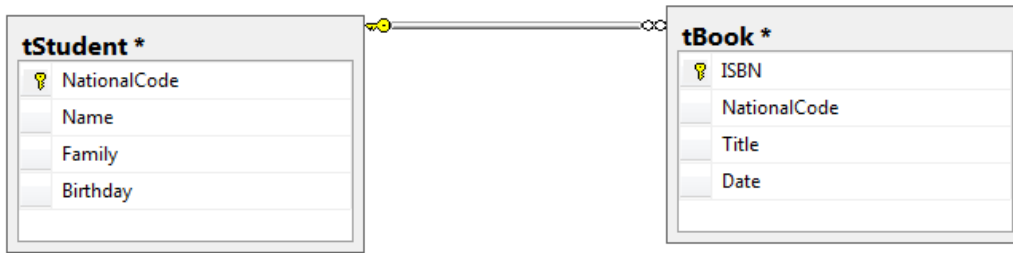
شکل ۵ - ایجاد Relation بین دو جدول

می توانید فرم زیر را نیز با فشردن دکمه Ok ببندید اما یک نکته در اینجا وجود دارد:

اگر دانشجویی از جدول tStudent حذف شود، باید همه رکوردهای مربوط به او در جدول tBook نیز حذف شود تا از سربار اضافی جلوگیری کرده باشیم و اطلاعات مرده در جدول tBook وجود نداشته باشد. برای این کار راه‌های مختلفی وجود دارد می‌توانید توسط LINQ و یا روآلهای ذخیره شده، پس از حذف از جدول tStudent، عمل حذف در جدول tBook را نیز در پس زمینه انجام دهید، اما راه ساده تر این است که بصورت ویژوالی عمل کنید:



شکل ۶ - تنظیمات مربوط به رابطه

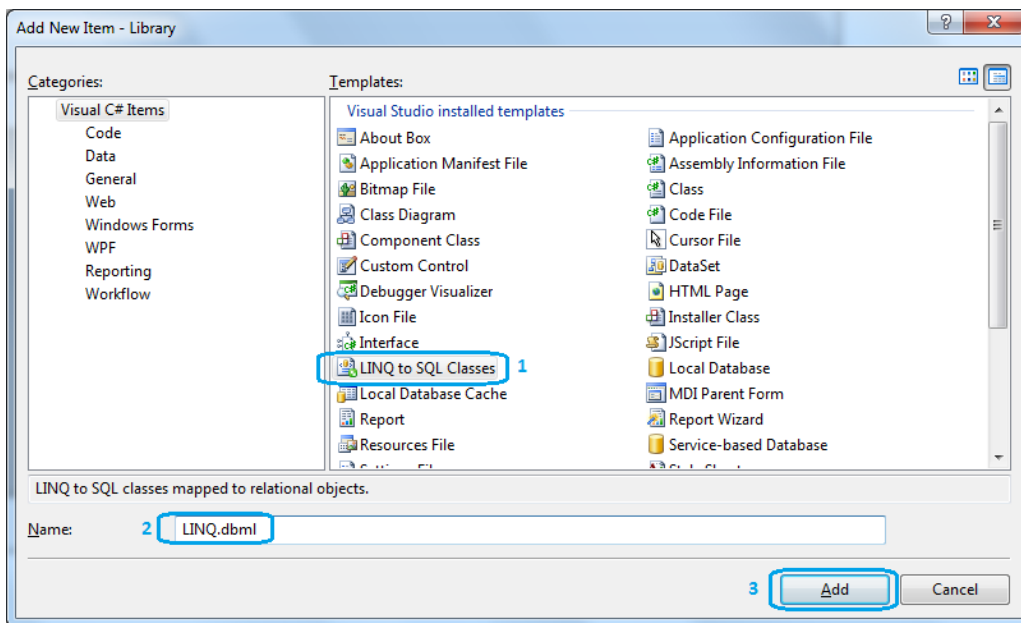


شکل ۷ - پیاده سازی ارتباط یک دانشجو با چند کتاب

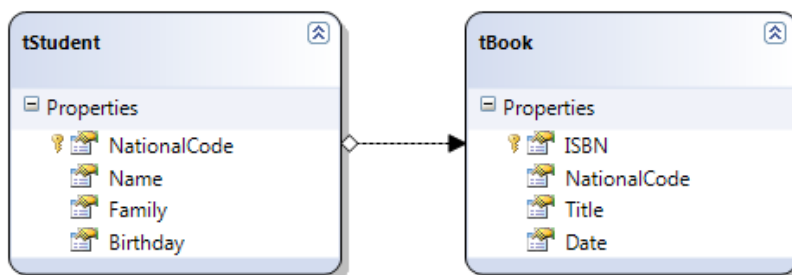
دیگرام را با همان نام پیش فرض Diagram1 ذخیره کنید. و در پیام نشان داده شده بعدی دکمه Yes را بزنید. سپس جدول tStudent را باز کنید و چند رکورد به آن اضافه کنید. بعد جدول tBook را باز کرده و به آن هم چند رکورد اضافه کنید. به خاطر داشته باشید ابتدا باید در جدول اصلی tStudent رکوردی را ایجاد کنید تا بتوانید در جدول tBook به آن کتاب اختصاص دهید.

ایجاد LINQ به SQL

از منوی Project > Add New Item یک شیء LINQ به پروژه اضافه کنید و دو جدول بانک اطلاعاتی را به داخل آن بکشید. با کشیدن اولین جدول، یک پیام نشان داده خواهد شد دکمه Yes را کلیک کنید.



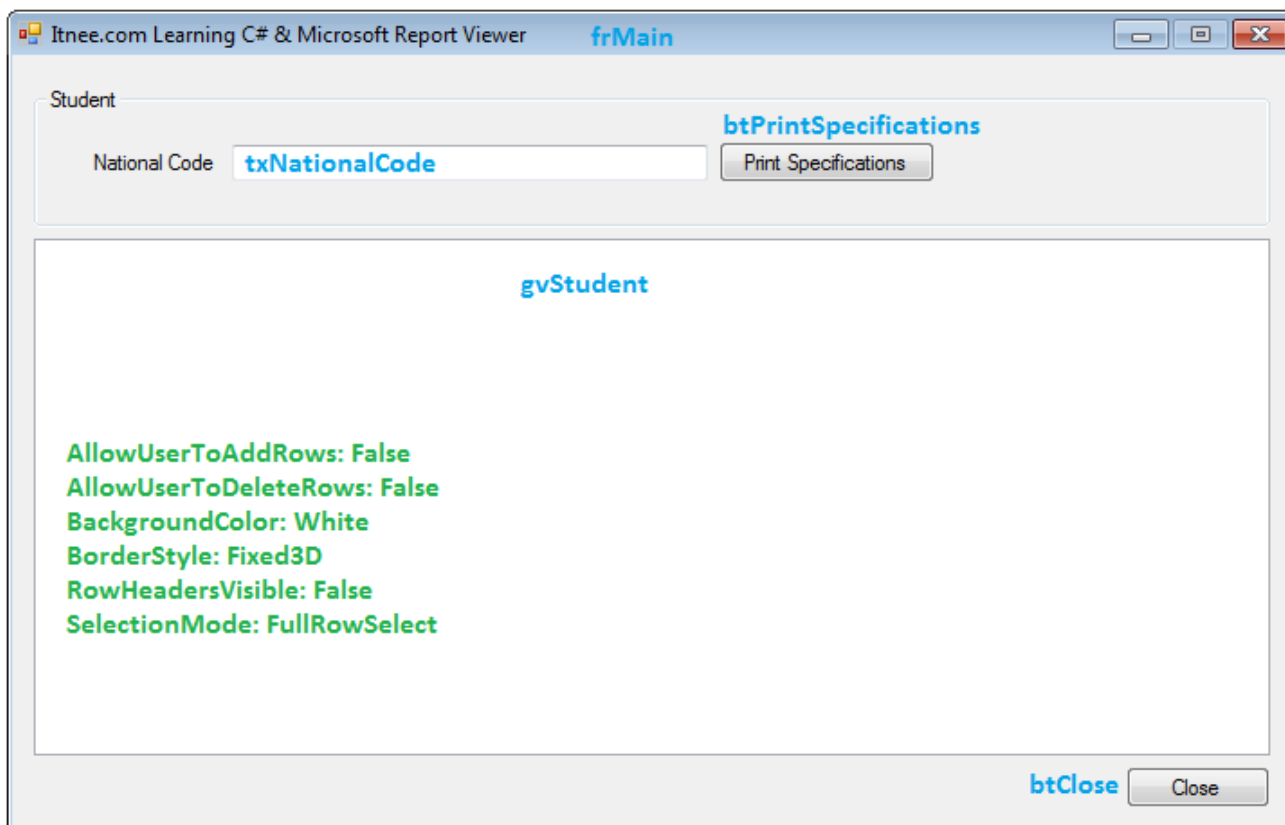
شکل ۸ - اضافه کردن محیط LINQ به پروژه



شکل ۹ - جداول با رابطه در محیط LINQ

فرم Form1 را به frMain تغییر نام دهید و مانند شکل زیر آنرا طراحی کنید.

در این فرم، کد ملی را جستجو کرده و مشخصات دانشجو را در یک گزارش نشان خواهیم داد. برای تمرین، گزارش دیگری بسازید که کتابهای امانت گرفته شده توسط دانشجو را نشان دهد.

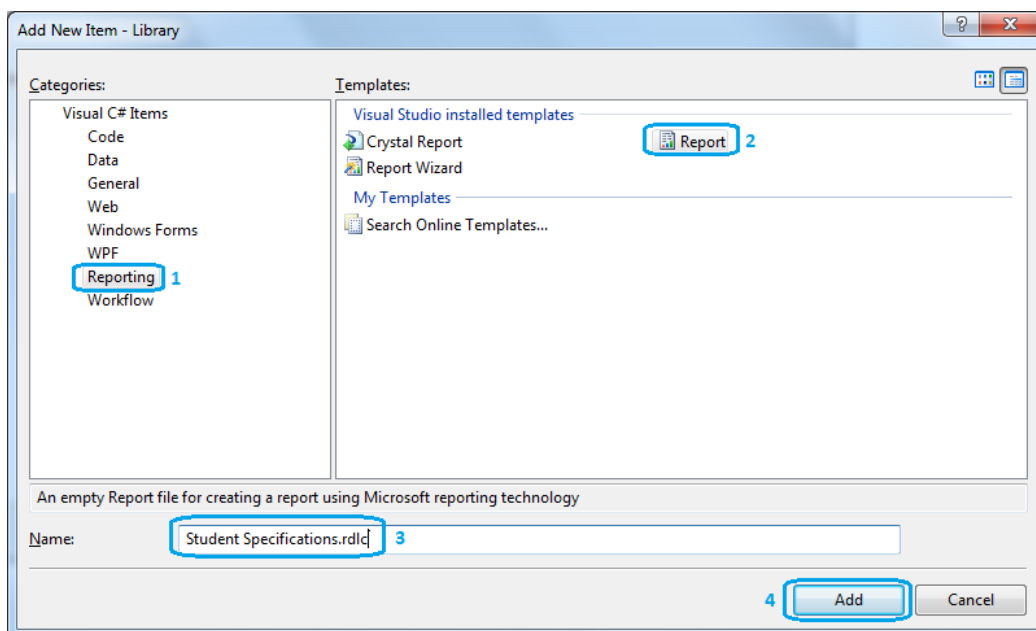


شکل ۱۰ - طراحی فرم اصلی برنامه

برای لود کردن لیست دانشجویان در گراید gvStudent، رویداد لود شدن فرم را بصورت زیر بنویسید:

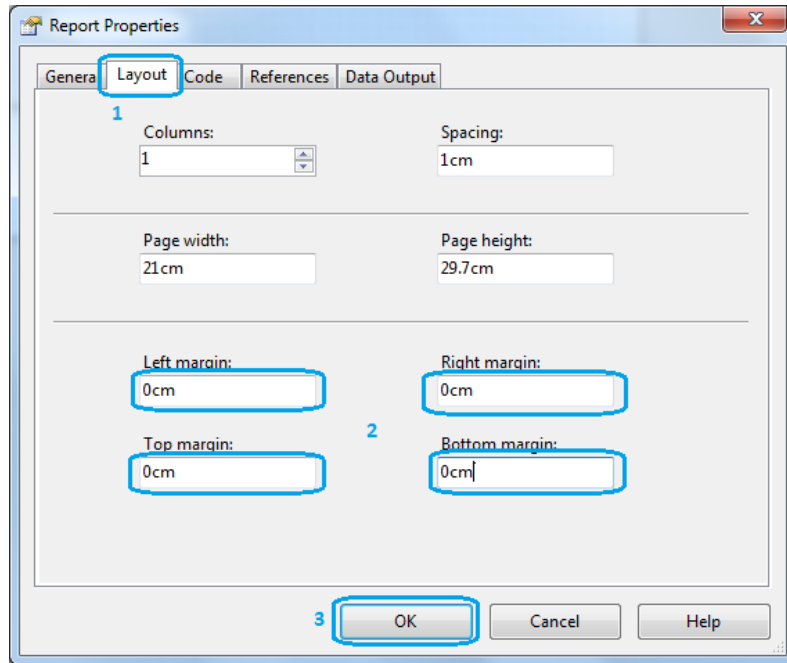
```
private void frMain_Load(object sender, EventArgs e)
{
    var db = new LINQDataContext();
    gvStudent.DataSource = db.tStudents;
}
```

۱. از منوی **Project > Add New Item** به بخش **Reporting** رفته و **Report** را انتخاب کنید. (شکل ۱۱)



شکل ۱۱ - ایجاد گزارش از نوع **Microsoft Report**

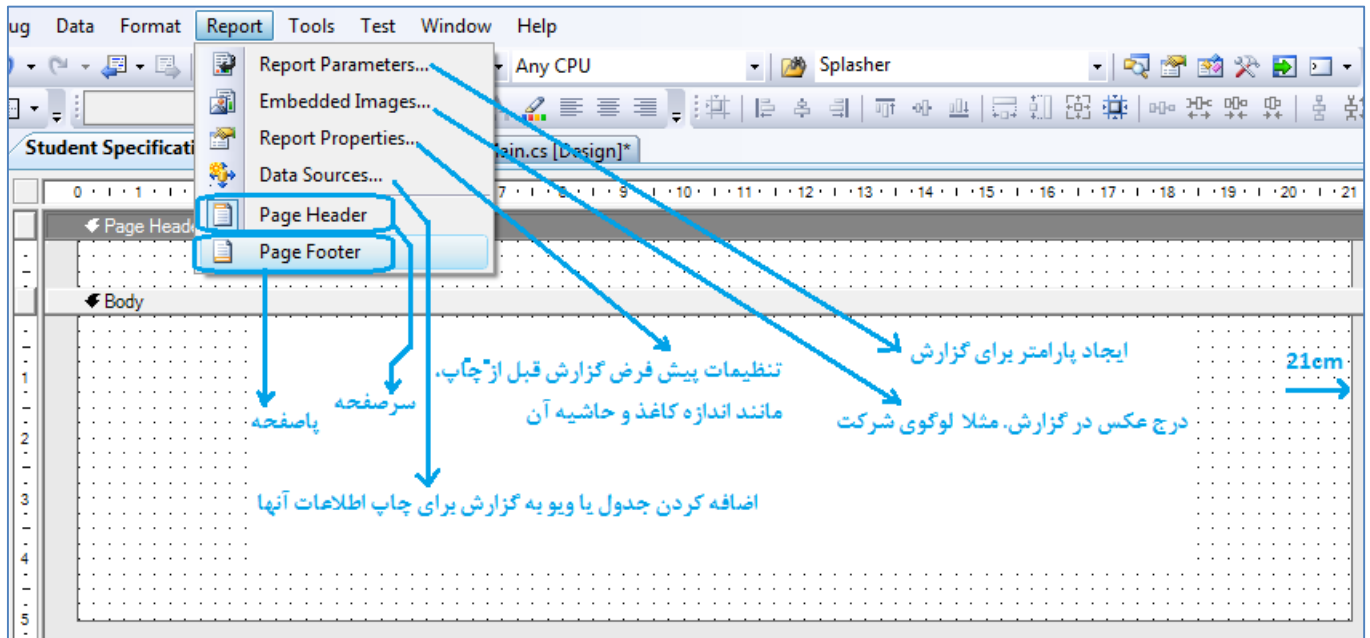
۲. به مسیر **Report > Report Properties** بروید و در فرم باز شده، مطابق شکل زیر حاشیه ها را به صفر تغییر دهید، حاشیه گزارش را در زمان طراحی در نظر می گیریم.



شکل ۱۲ - تنظیم مشخصات پیش فرض گزارش قبل از چاپ


۳. عرض گزارش را به 21cm افزایش دهید و مطابق شکل ۱۳ سرصفحه و پاصفحه را فعال کنید:

توجه: برای اینکه منوی **Report** در ویزوال استودیو ظاهر شود، لازم است روی گزارش کلیک کنید.



شکل ۱۳ - پیکربندی گزارش توسط منوی **Report**

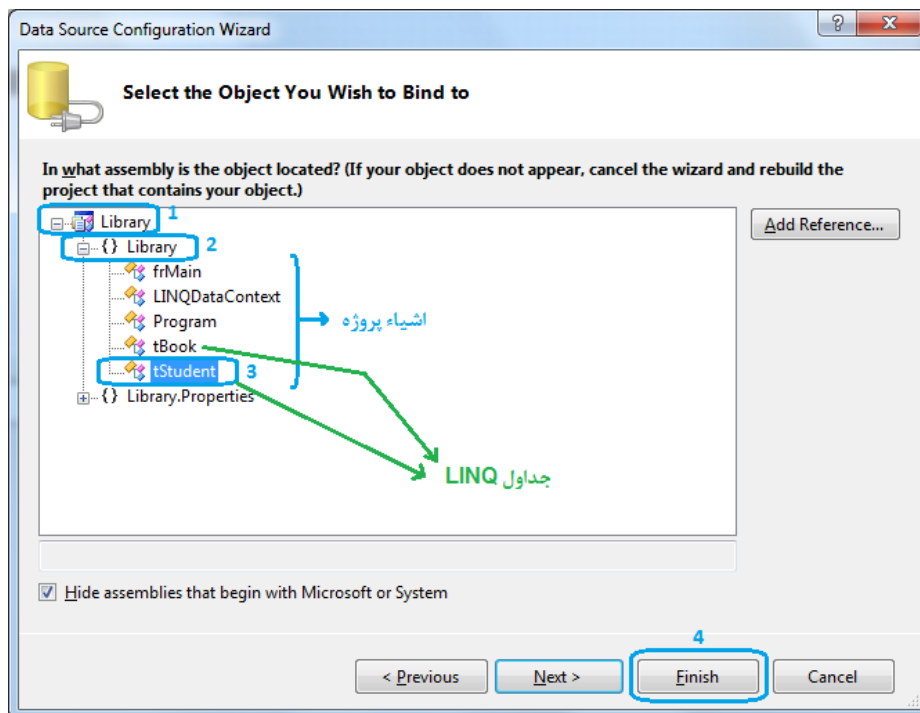
ایجاد Data Source

برای ایجاد ارتباط بین LINQ و گزارش، به یک منبع داده نیاز داریم تا اطلاعات در آن نگهداری شود. بنابراین از منوی **Data** گزینه **Show Data Sources** را بزنید تا پانل **Data Sources** باز شود. سپس آیکن  را کلیک کنید. در ویزارد باز شده، **Object** را انتخاب کنید. و مانند شکل ۱۴ عمل کنید:

Add New Data Sources

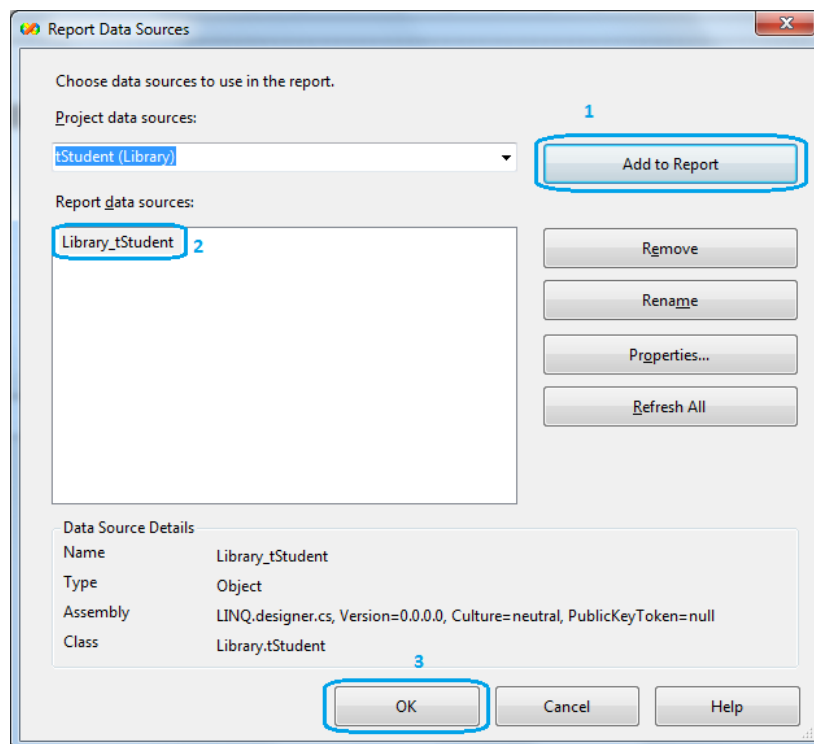
پس از فشردن دکمه **Finish**، هر دو جدول به محیط **Data Sources** اضافه می شوند (به دلیل رابطه دار بودن آنها) در مواردیکه جداول ارتباطی با هم ندارند باید این مرحله را برای هر جدول تکرار کنید تا به **Data Sources** اضافه شود و بتوان در گزارش از آن استفاده کرد.

توجه: در صورتیکه شمای جدول خود را در بانک اطلاعاتی تغییر دهید، باید آنرا از **LINQ** حذف کرده و دوباره اضافه کنید، تا تغییرات قابل مشاهده باشد. همینطور لازم است جدول را از پانل **Data Sources** هم حذف کنید (**Remove Object**) و دوباره اضافه نمایید.



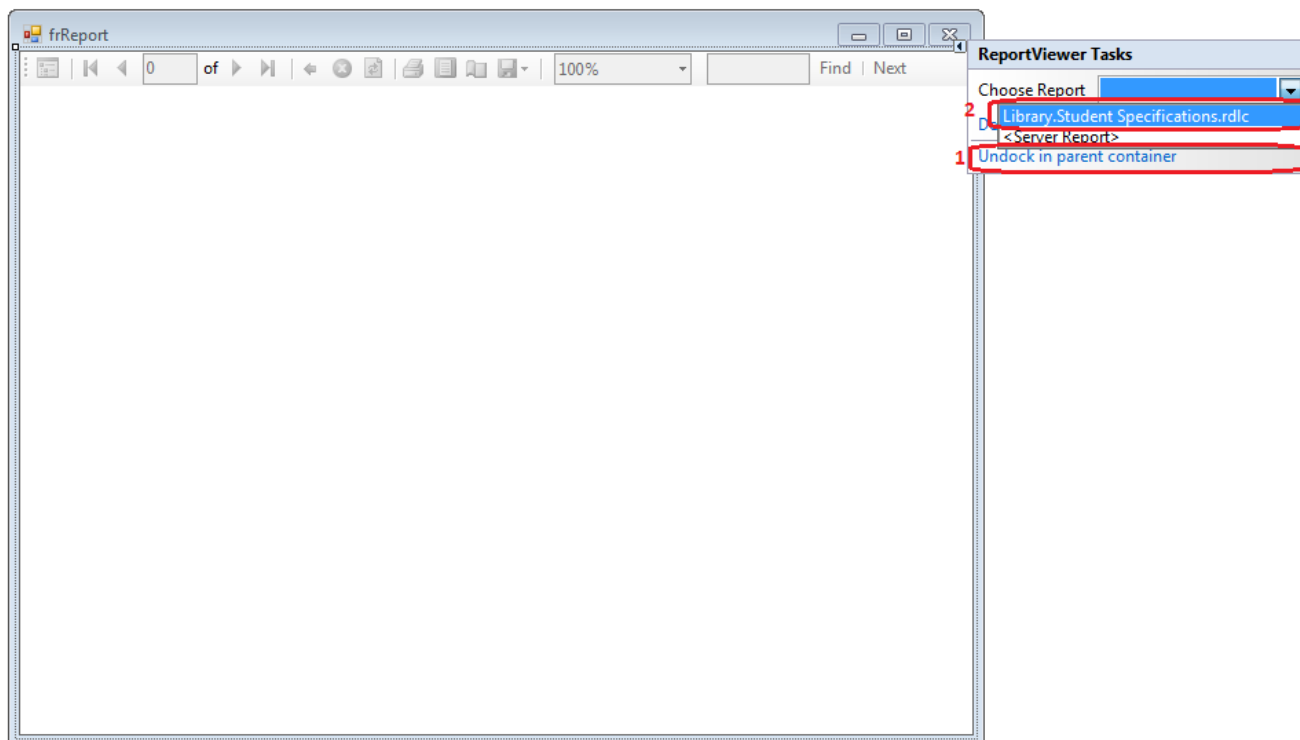
شکل ۱۴ - انتخاب جدول **tStudent** برای منبع داده

۴. از منوی **Report** گزینه **Data Sources** را انتخاب کنید. خواهید دید دو جدول به لیست آن اضافه شده اند. جدول **tStudent** را انتخاب کنید و **Ok** را بزنید. (شکل ۱۵)



شکل ۱۵ - انتخاب جدول از Data Source برای گزارشگیری

۵. حالا فیلدها را از Data Sources بر روی گزارش بکشید و برای آنها برچسب مناسب قرار دهید و سپس گزارش را ذخیره کنید.
۶. یک فرم جدید به نام frReport بسازید و در Toolbox از قسمت Reporting شیء Microsoft Report Viewer را روی فرم بکشید
۷. در پانل شناور ReportViewer Tasks لینک Dock in parent container را کلیک و نام گزارش را انتخاب کنید. (شکل ۱۶) با این کار شیئی به نام tStudentBindingSource در نوار پایین ایجاد می شود که توسط آن و با استفاده از LINQ خروجی گزارش را کنترل می کنیم.



شکل ۱۶ - انتخاب گزارش برای Microsoft Report Viewer

۸. حال نوبت به کد نویسی این فرم است که شامل دو بخش است:

الف) تعریف یک متغیر عمومی، برای ارسال کلید از فرم اصلی به این فرم.

ب) نسبت دادن یک عبارت LINQ به خاصیت DataSource از شیء tStudentBindingSource

```
public partial class frReport : Form
{
    public string nationalCode = ""; 1
    public frReport()...

    private void frReport_Load(object sender, EventArgs e) 2
    {
        var db = new LINQDataContext();
        tStudentBindingSource.DataSource = db.tStudents.Where(c => c.NationalCode == nationalCode).Select(c => c).First();
        this.reportViewer1.RefreshReport();
    }
}
```

شکل ۱۷ - کد فرم frReport

۹. به فرم اصلی (frMain) رفته و کد دکمه Print Specifications را بصورت زیر بنویسید.

```
private void btPrintSpecifications_Click(object sender, EventArgs e)
{
    var db = new LINQDataContext();
    if (db.tStudents.Where(c => c.NationalCode == txNationalCode.Text).Count() == 1)
    {
        var f = new frReport();
        f.nationalCode = txNationalCode.Text;
        f.ShowDialog();
    }
    else
        MessageBox.Show("Student not found!");
}
```

با فرض اینکه txNationalCode را کنترل می کنید که مقدار معتبری داشته باشد.

برنامه را اجرا کنید، با وارد کردن یک کد ملی معتبر، و فشردن دکمه چاپ، فرم frReport اجرا و کد ملی برای آن ارسال می شود. سپس در هنگام لود شدن فرم چاپ، کنترل tStudentBindingSource با عبارت LINQ دارای شرط محدود کننده مقداردهی می شود و فقط یک رکورد به خروجی می رود.

مثالهای بیشتر

۱. برای چاپ همه رکوردها، یک کنترل Table روی گزارش بکشید و خاصیت DataSetName آنرا برابر Library_tStudent قرار دهید سپس آنرا مانند شکل ۱۸ طراحی کنید:

Page Header				
Body				
جدول	نام	نام خانوادگی	تاریخ تولد	کدملی
	=Fields!Name.Value	=Fields!Familv.Value	=Fields!Birthdav.Value	=Fields!NationalCode.Value
			Footer	
Page Footer				

شکل ۱۸ - استفاده از کنترل Table برای نمایش چند رکورد

در رویداد Load فرم frReport عبارت LINQ را بصورت زیر تغییر دهید تا همه رکوردها به خروجی فرستاده شود.

```
tStudentBindingSource.DataSource = db.tStudents;
```

۲. برای مرتب کردن خروجی از عبارت زیر استفاده کنید:

```
tStudentBindingSource.DataSource = db.tStudents.OrderBy(c => c.Family);
```

نکته: از جدول نمی توان در Header یا Footer استفاده کرد.

قراردادن عکس داخلی (Embedded) در گزارش

از منوی Report گزینه Embedded Images را انتخاب کنید.

دکمه New Image... را کلیک کنید. یک تصویر انتخاب کنید و Ok را بزنید.

یک کنترل Image در Header گزارش قرار دهید و خاصیت Source آنرا برابر Embedded و سپس خاصیت Value را برابر با نام تصویر قرار دهید.

ذخیره عکس در بانک اطلاعاتی (از PictureBox)

```
//using System.IO;
MemoryStream ms = new MemoryStream();
pbPic.Image.Save(ms, pbPic.Image.RawFormat);
byte[] arrImage = ms.GetBuffer();
ms.Close();
```

```
Main.db.pCardInsert(Convert.ToInt64(txID.Text), txName.Text, txFamily.Text,
txSemat.Text, cbDepartment.Text, Convert.ToInt64(txCode.Text), arrImage);
```

بازبایی عکس از بانک اطلاعاتی (به PictureBox)

```
var q = Main.db.tCards.Where(c => c.ID == id).Select(c => c).First();
lbNameFamily.Text = q.Name + " " + q.Family;
lbCode.Text = q.Code.ToString();
lbDepartment.Text = q.Department;
```

```
lbSemat.Text = q.Semat;  
lbID.Text = q.ID.ToString();  
  
byte[] arrImage = q.Pic.ToArray();  
MemoryStream ms = new MemoryStream(arrImage);  
pbMain.Image = Image.FromStream(ms);
```

پروسیجر ذخیره عکس در بانک اطلاعاتی (فقط یک پارامتر از نوع image اضافه کنید)

```
ALTER PROCEDURE [dbo].[pCardInsert] @ID bigint, @Name nvarchar(20), @Family  
nvarchar(30), @Semat nvarchar(40), @Department nvarchar(30), @Code bigint, @Pic image  
AS  
INSERT INTO tCard VALUES(@ID, @Name, @Family, @Semat, @Department, @Code, @Pic)  
RETURN
```

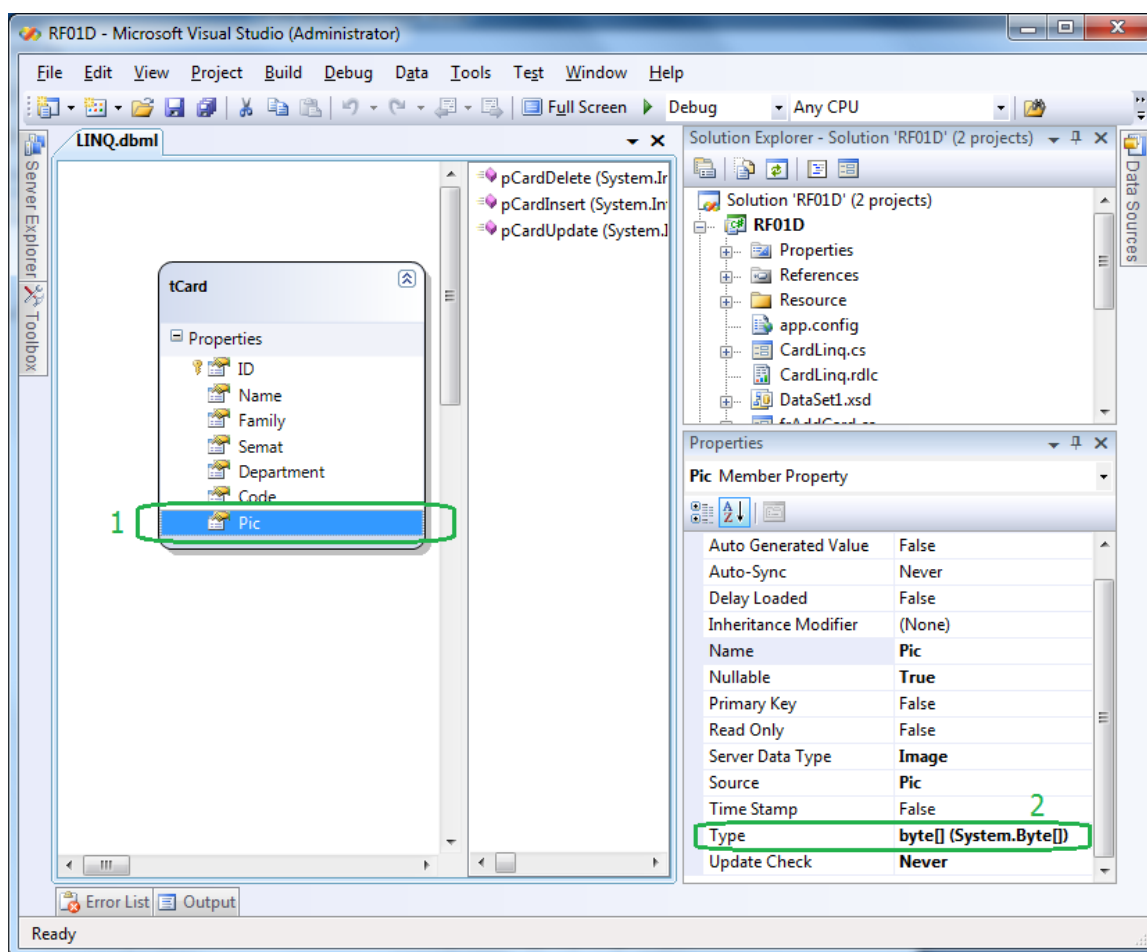
بازیابی عکس از بانک اطلاعاتی و نمایش دادن در گزارش

۱. یک فیلد با نوع image در جدولی به نام tCard بسازید و جدول را ذخیره کنید.

Column Name	Data Type
ID	bigint
Name	nvarchar(20)
Family	nvarchar(30)
Semat	nvarchar(40)
Department	nvarchar(30)
Code	bigint
Pic	image

شکل ۱۹ - جدول tCard با فیلدی از نوع Image

۲. جدول را به LINQ بکشید و مانند شکل زیر نوع فیلد Pic را تغییر دهید.



شکل ۲۰ - تنظیمات فیلد Pic در محیط LINQ

۳. یک کنترل Textbox و یک Image روی گزارش rdlc قرار دهید.

در Properties مربوط به کنترل Textbox خاصیت Visibility > Hidden را برابر True قرار دهید.

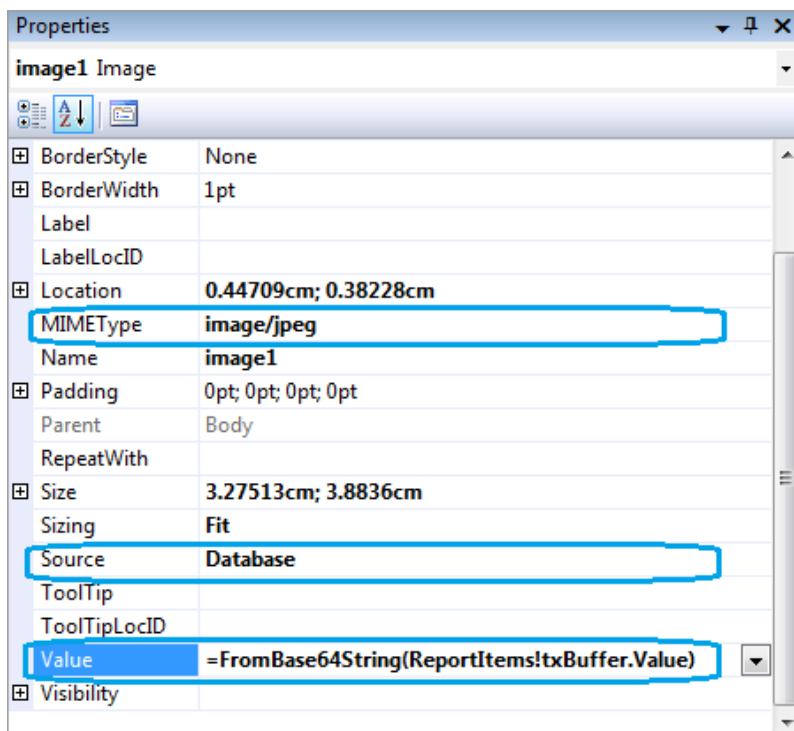
عکس را توسط کنترل میانی Textbox از بانک اطلاعاتی دریافت می کنیم و پس از تبدیل، به کنترل Image می فرستیم.

۴. کنترل Textbox را به txBuffer تغییر نام دهید. و برای خاصیت Value آن مقدار زیر را بنویسید:

```
=ToBase64String(First(Fields!Pic.Value, "Library_tStudent"))
```

۵. برای خاصیت Value کنترل Image مقدار زیر را بنویسید و مطابق شکل آنرا تنظیم کنید.

```
=FromBase64String(ReportItems!txBuffer.Value)
```



شکل ۲۱ - تنظیمات کنترل Image

با اجرای برنامه، عکس در Image به نمایش در می آید.

توجه داشته باشید برای ذخیره عکس در فیلدی از نوع Image آنرا به یک آرایه از نوع byte تبدیل کردیم و برای بازیابی این فیلد از بانک اطلاعاتی و نمایش در PictureBox آنرا دوباره تبدیل به عکس کردیم.

بازیابی عکس از بانک اطلاعاتی به ASP . NET

۱. یک فایل از نوع Generic Handler با نام `HandlerPersonel.ashx` به پروژه اضافه کنید. و متد `ProcessRequest` را بصورت زیر جایگزین کنید:

```
public void ProcessRequest(HttpContext context)
{
    //using System.Data.SqlClient;
    //using System.Configuration;
    string id = context.Request.QueryString["id"];
    SqlConnection Cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["testConnectionString"].ConnectionString);
    Cn.Open();
    SqlCommand command = new SqlCommand("SELECT Img FROM vPersonel WHERE PersonelCode=" + id, Cn);
    SqlDataReader dr = command.ExecuteReader();
    dr.Read();
    context.Response.BinaryWrite((Byte[])dr[0]);
    Cn.Close();
    context.Response.End();
}
```

۲. یک کنترل `Image` در صفحه ای که می خواهید عکس را نمایش دهید، قرار دهید.

۳. رویداد زیر را برای `GridView` بنویسید. با انتخاب هر ردیف از `GridView` کلید جدول به فایل `HandlerPersonel.ashx` ارسال می شود و عکس در `Image1` به نمایش در می آید. در صورتیکه عکس وجود نداشته باشد، تصویر `NoImage.png` نمایش داده می شود.

```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    var db = new LINQAccountingDataContext();
    if (db.vPersonels.Where(c => c.PersonelCode ==
Convert.ToDouble(GridView1.SelectedRow.Cells[3].Text)).Select(c => c.Img).First() != null)
        Image1.ImageUrl = "HandlerPersonel.ashx?id=" +
GridView1.SelectedRow.Cells[3].Text;
    else
        Image1.ImageUrl = "Images/NoImage.png";
}
```

