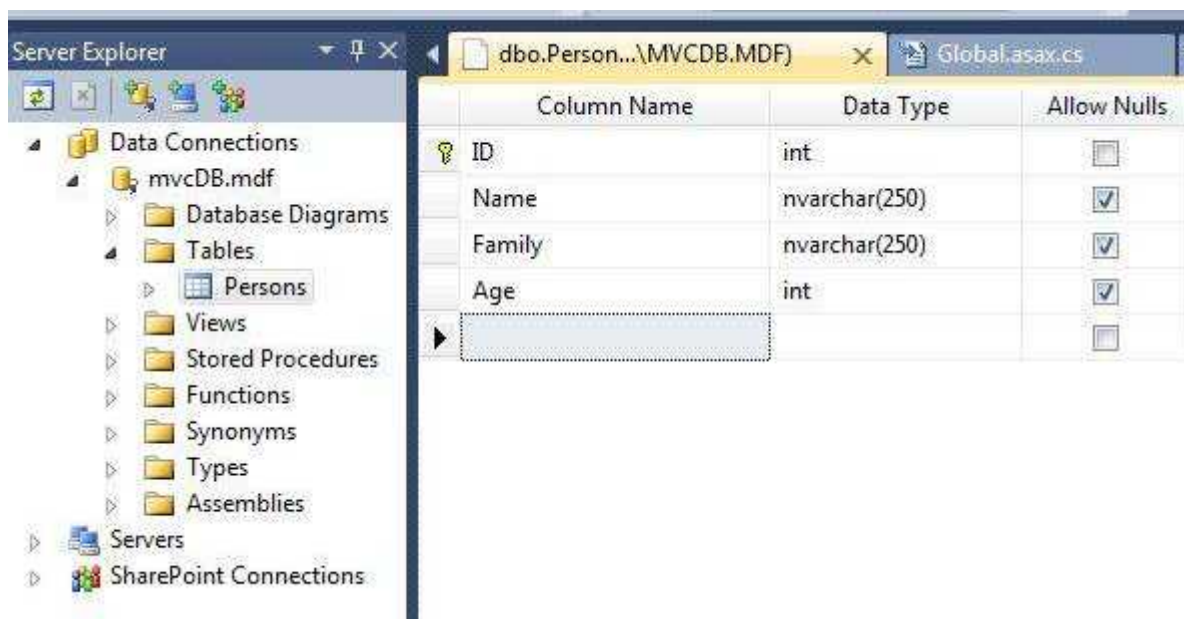


در بخش دوم درباره چگونگی نوشتن چند اکشن در یک کنترلر و ارتباط آن با URL صفحه دیدید، در این بخش درباره Model و نحوه استفاده از آن صحبت می کنیم .

### - Model

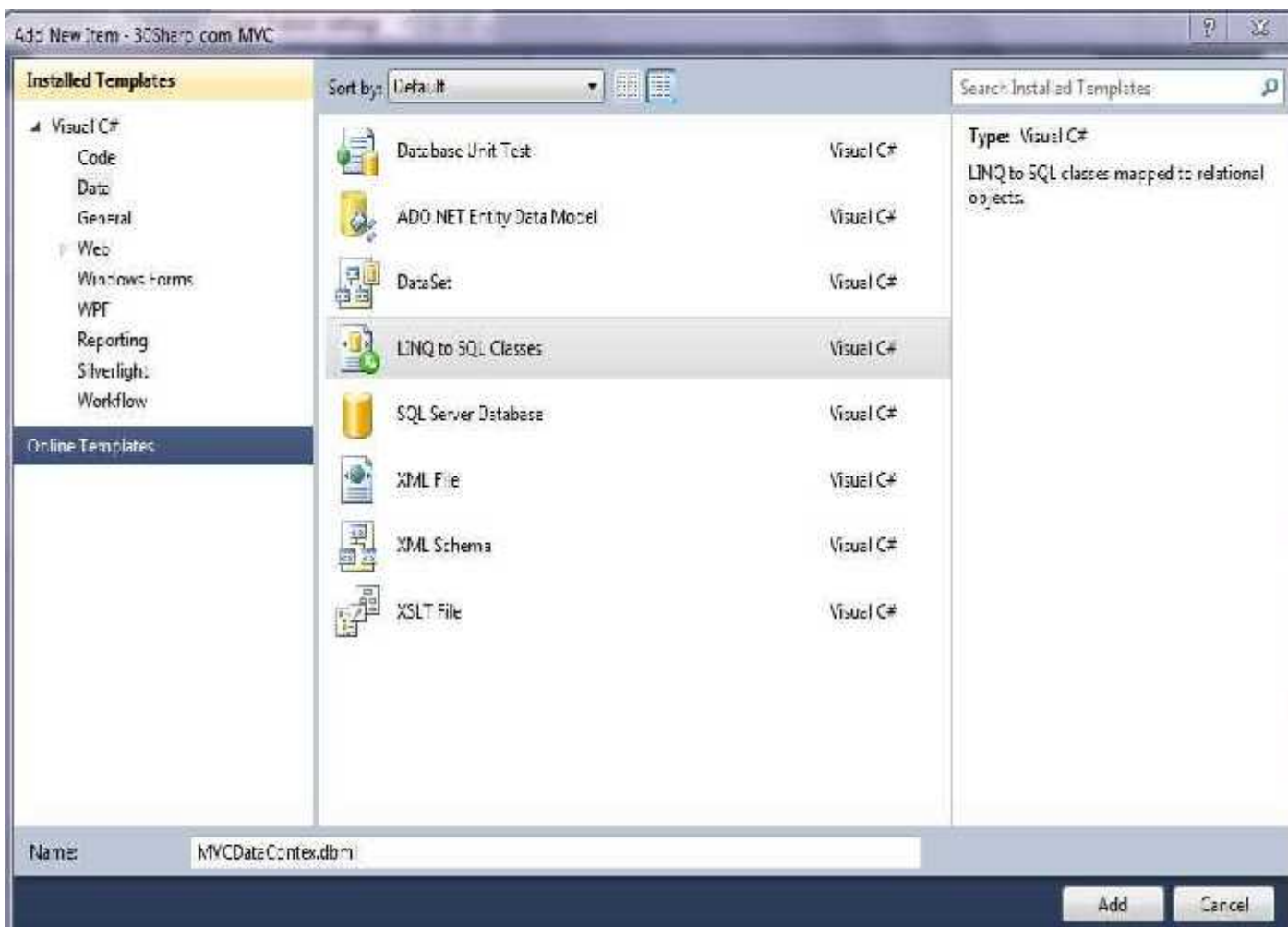
مدل در MVC ، محلی برای ایجاد کلاسهای مرتبط با بانک اطلاعاتی است. در مدل ها نیز می توانید از تمام الگوهای ارتباط با بانک اطلاعاتی مثل Linq یا Entity Framework استفاده کنید .

برای اینکه بتوانیم مثالی از Model ها بیاوریم ، ابتدا یک بانک اطلاعاتی SQL با یک جدول ساده شامل فیلدهایی مثل نام ، فامیل و سن ایجاد کنید ، یعنی باید جدولی به فرم زیر داشته باشیم :

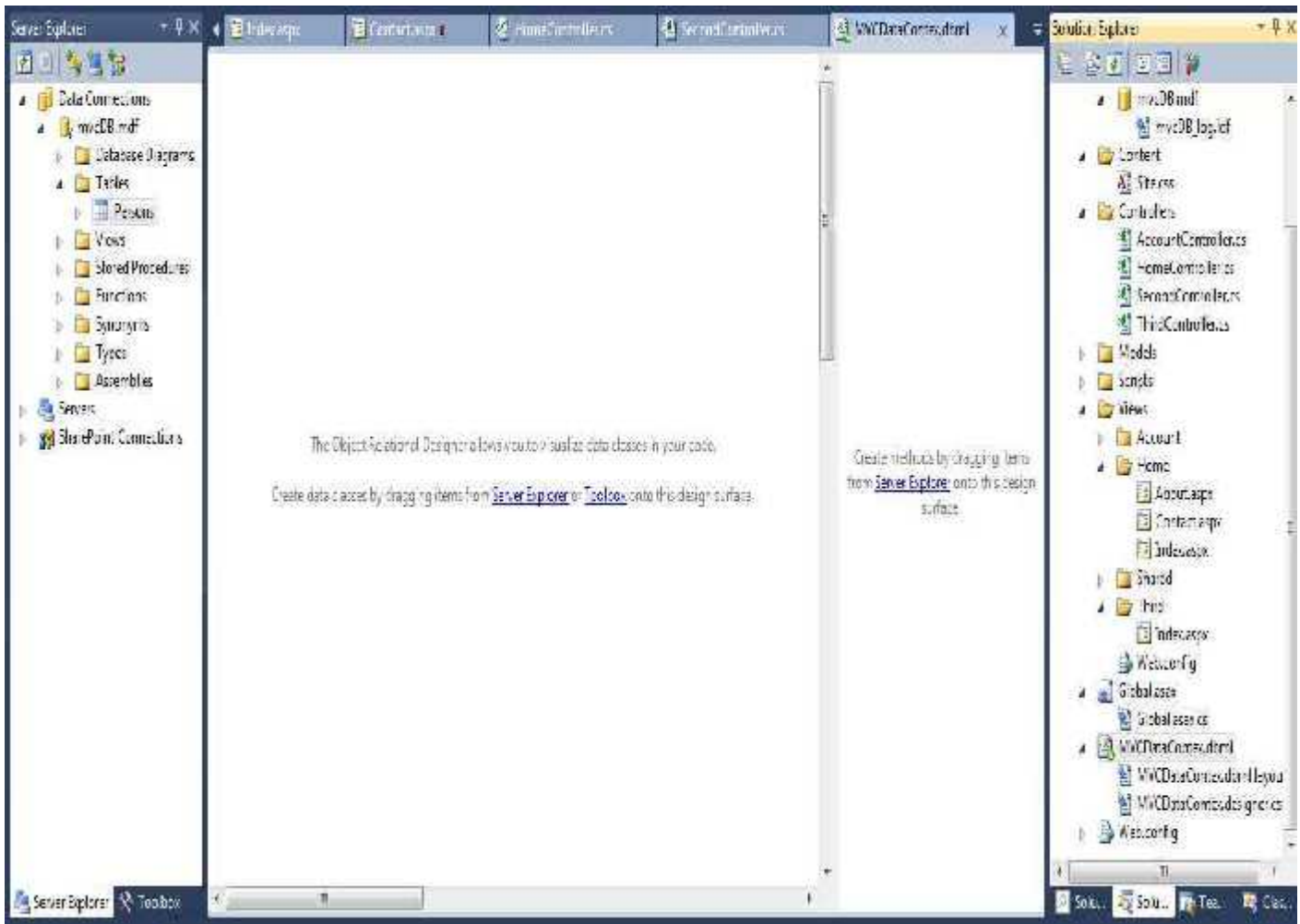


Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Name	nvarchar(250)	<input checked="" type="checkbox"/>
Family	nvarchar(250)	<input checked="" type="checkbox"/>
Age	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

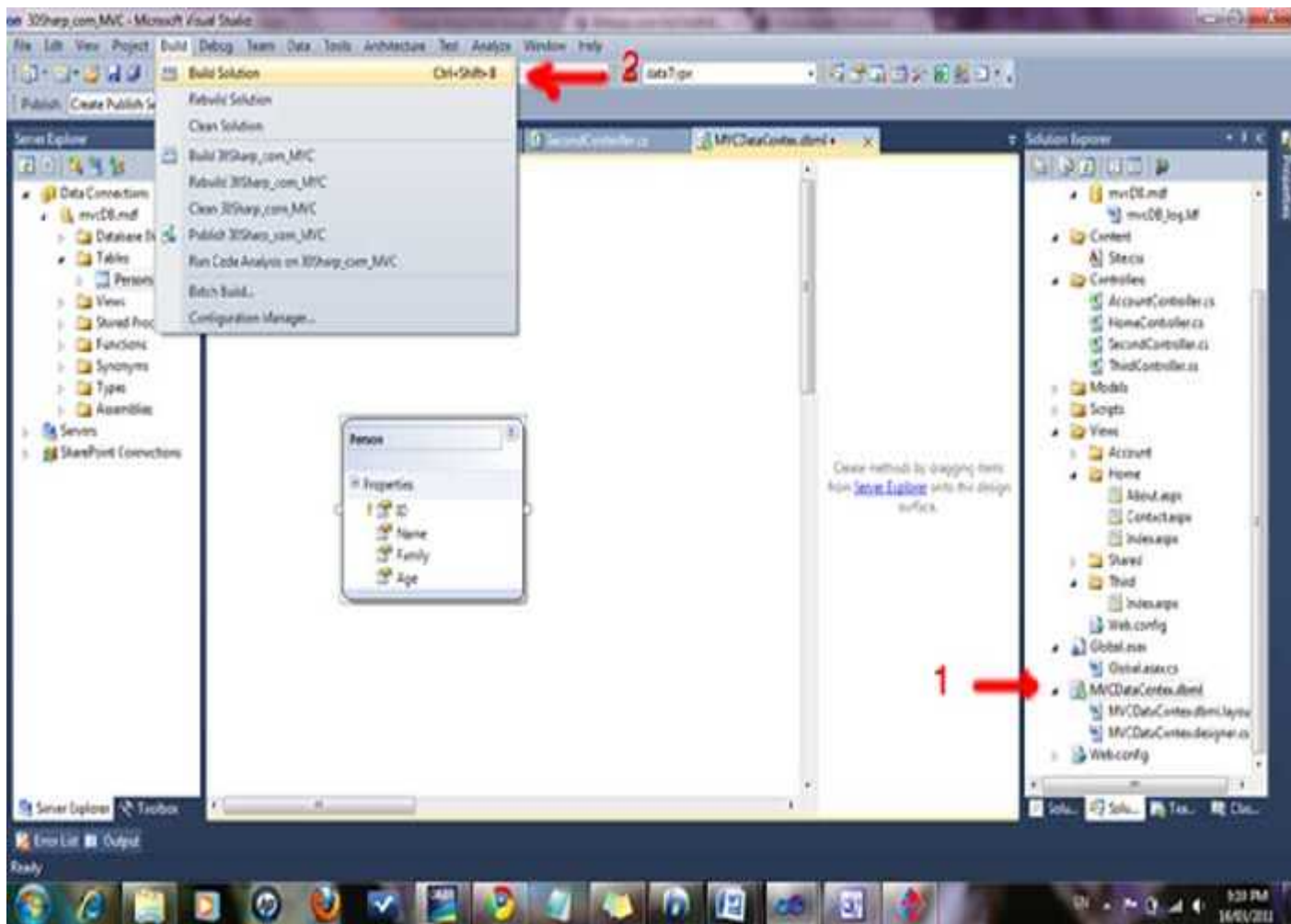
سپس برای ارتباط با بانک اطلاعاتی یک DataContext از نوع Linq to SQL ایجاد می کنیم . برای این کار روی Solution خود کلیک راست کرده و از قسمت Data گزینه Linq to SQL را انتخاب کنید:



سپس جدول ساخته شده در بانک اطلاعاتی را به داخل آن Drog&Drop کنید :



و Solution را Build کنید.



حال در Solution Explorer روی Controllers کلیک راست کرده و یک کنترلر جدید ایجاد می کنیم و نام آنرا PersonController می گذاریم و کد زیر را درون آن وارد می کنیم :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using _30Sharp_com_MVC.Models;

namespace _30Sharp_com_MVC.Controllers
```

```

{
    public class PersonController : Controller
    {
        MVCDataContexDataContext db = new MVCDataContexDataContext();

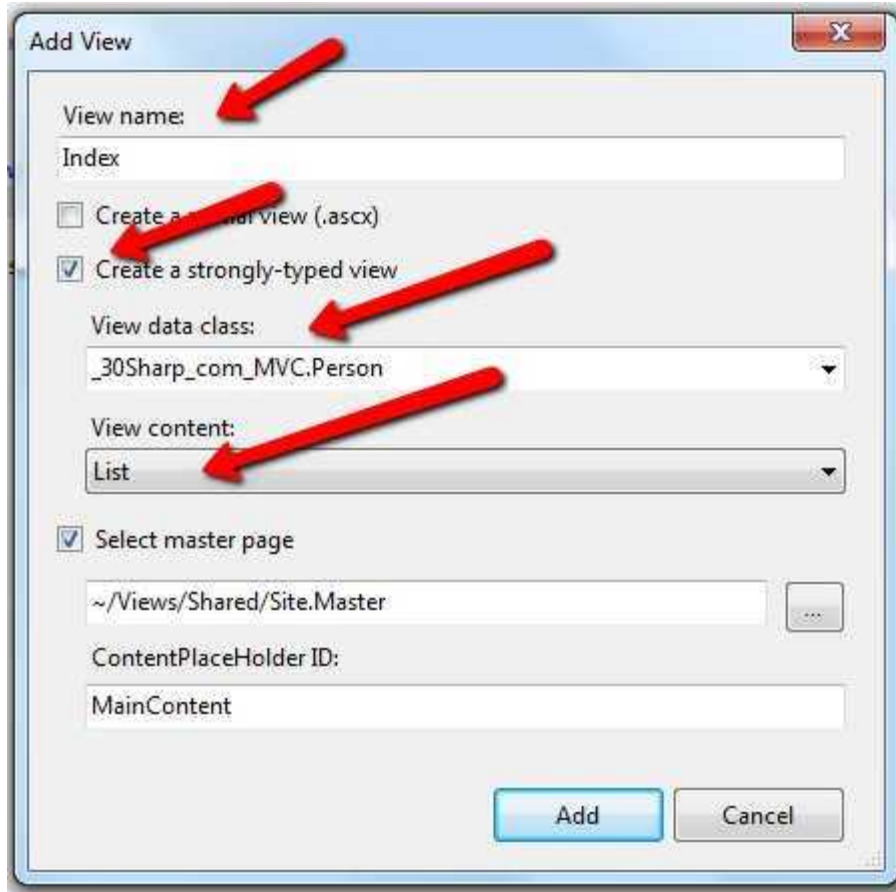
        public ActionResult Index()
        {
            var list = (from m in db.Persons select m ).ToList();

            return View(list);
        }
    }
}

```

همانطور که می بینید در این کنترلر یک نمونه از شیئی Data Context با نام db ایجاد کرده ایم و سپس با استفاده از Linq to Sql لیست کلیه سطرهای جدول Person را استخراج کرده ایم . توجه نمایید که خروجی اکشن از نوع ActionResult() است که لیستی از جدول Person را به View ارسال می کند ( اگر در مفاهیم Linq to Sql مشکل دارید ، توصیه می شود که قبل از ادامه این مبحث آن را مطالعه کنید)

پس تا اینجا نتایج را به View ی مربوطه انتقال دادیم . خب زمان آن رسیده که View مناسب برای نمایش این فیلدها ایجاد کنیم . با کلیک راست روی اکشن Index گزینه Add view را انتخاب می کنیم و آنرا Index نام می نهیم. در هنگام ایجاد View به پارامترهایی که با فلش مشخص شده اند توجه فرمایید.



کد View بصورت زیر ایجاد می شود :

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master"
Inherits="System.Web.Mvc.ViewPage<IEnumerable<_30Sharp_com_MVC.Person>>" %>

<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent"
runat="server">
    Index
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2>
        Index</h2>
    <table>
```

```

<tr>
  <th>
</th>
  <th>
    ID
  </th>
  <th>
    Name
  </th>
  <th>
    Family
  </th>
  <th>
    Age
  </th>
</tr>
<% foreach (var item in Model)
  { %>
<tr>
  <td>
    <%: Html.ActionLink("Edit", "Edit", new { id=item.ID }) %>
    |
    <%: Html.ActionLink("Details", "Details", new { id=item.ID
}) %>
    |
    <%: Html.ActionLink("Delete", "Delete", new { id=item.ID }) %>
  </td>
  <td>
    <%: item.ID %>

```

```

        </td>

        <td>

            <%: item.Name %>

        </td>

        <td>

            <%: item.Family %>

        </td>

        <td>

            <%: item.Age %>

        </td>

    </tr>

    <% } %>

</table>

<p>

    <%: Html.ActionLink("Create New", "Create") %>

</p>
</asp:Content>

```

همانگونه که مشاهده می کنید ، در کد فوق ، یک حلقه foreach وجود دارد که لیست ذخیره شده را چاپ می کند . اما چند نکته در مورد View فوق :

۱. در خط اول می بینید که در Inherit نام جدول Person آورده شده و منظور این است که از Model استفاده شده در این صفحه از نوع Person می باشد. این خط به این دلیل ایجاد شده که در هنگام ایجاد این View ، در قسمت View data class گزینه Person\_30Sharp\_com\_MVC را انتخاب کرده ایم . اگر دیدید که چنین گزینه ای وجود ندارد ، دلیلش آن است که پروژه را Build نکرده اید .
۲. در هنگام اضافه کردن View در قسمت View Content گزینه های دیگری مثل Create ، Delete ، Edit وجود دارد که با انتخاب هر کدام ، کد تولید شده در View متفاوت خواهد بود .



۳. پس از اجرا برای دیدن خروجی به آدرس زیر بروید :

localhost:0000/Person/index

که به جای ۰۰۰۰ شماره پورتی که ویژال استودیو از آن استفاده می کند قرار می گیرد .

۴. وقتی برنامه را اجرا کنید ، متوجه خواهید شد که لیست سطرهای جدول نمایش داده می شود ولی لینکهای Edit

/Details/ Delete / و همچنین Create New کار نمی کند. چرا؟ چون هنوز Action متناظر با آنها ایجاد نشده است .

۵. کلاس Html دارای متدهای مختلفی است از جمله ActionLink که برای لینک دادن بکار میرود . و دارای ۱۰ حالت

مختلف ( Overload ) است که در این حالت یعنی :

```
<%: Html.ActionLink("Create New", "Create") %>
```

پارامتر اول متن لینک و پارامتر دوم نام Action ی است که اجرا می کند.