

بسم الله الرحمن الرحيم

## أستخدام تقنية Linq To SQL في ASP.Net

### الدرس الاول

#### مقدمة :

بعيدا عن المقدمات المعقدة والطويلة سنختصر الكلام الى اقل ما يمكن ، عندما تقوم بعمل مشروع سواء تطبيق (Application) او ويب (WebSite) لربطه مع قاعدة البيانات بالتأكد ستحتاج الى استعمال كائنات مثلما رأيتموه في (الجزء الاول من دورة تصميم موقع أخباري عندما استعملنا كلاس ووضعنا فيه دوال للإضافة والحذف والتعديل وحتى استرجاع البيانات ) بالتالي فأنت ستحتاج وقت طويل لعمل شي بسيط (Ado.net) على سبيل المثال أسترجاع سجل واحد بمعنى أنك ستكون مجبوراً الى تحميله هذا السجل الى أحد الكائنات ثم أستخدم الكائن فكيف اذا احتجت الى التعامل بصورة أكثر مع قاعدة البيانات؟! ، هنا ستكون فجوة بين اللغة التي تستخدمها للبرمجة وبين قاعدة البيانات .

اذا نستطيع ان نقول الغاية الاساسية من ظهور تقنية Linq To SQL هو ازالة هذا التعقيد وتسهيل ربط لغة البرمجة مع قاعدة البيانات بالإضافة الى الوثوقية والسرعة الخ .

ف Linq To SQL عبارة عن تقنية انتجها مطوري مايكروسوفت وتحوي على أدوات وأستعلامات تستخدم للتعامل مع البيانات الموجودة في قاعدة البيانات ، وللعمل هذه التقنية هي جزء من التقنية الام Linq والتي تحوي بالإضافة الى تقنية Linq To SQL تقنية Linq To Object للتعامل بالبيانات الموجودة في الكائنات مثل المصفوفات والقوائم ( Lists ) الخ ،

وتقنية Linq To XML للوصول الى البيانات في مستندات Xml، كلامنا سيكون فقط على Linq To SQL وأستخدامها بأبسط صورة مع Asp.net اما التقنيات البقية فهي مشابهة الى حد ما ل Linq To SQL في طريقة الاستخدام والعمل والاستعلامات .

---

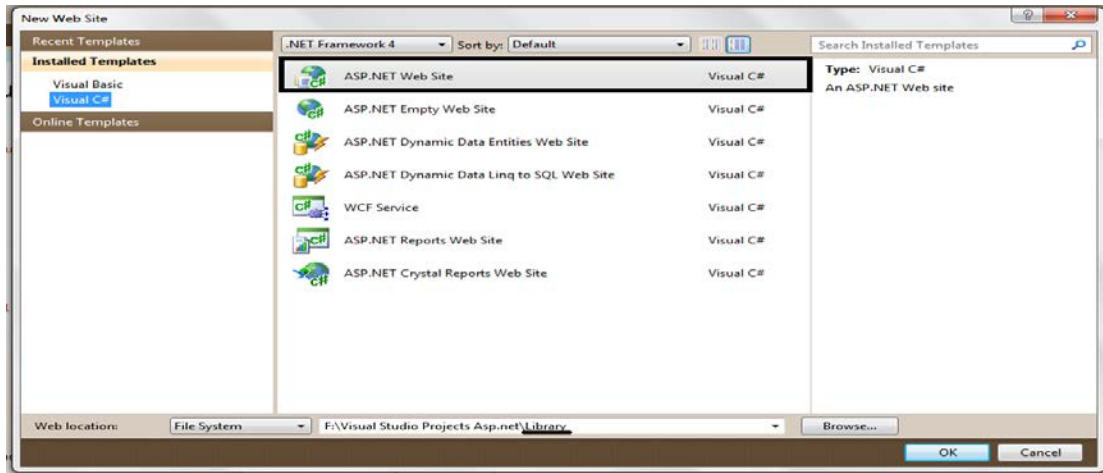
## العملی

مثالنا سيكون عبارة عن مكتبة بسيطة وفيها اصناف الكتب والكتب .

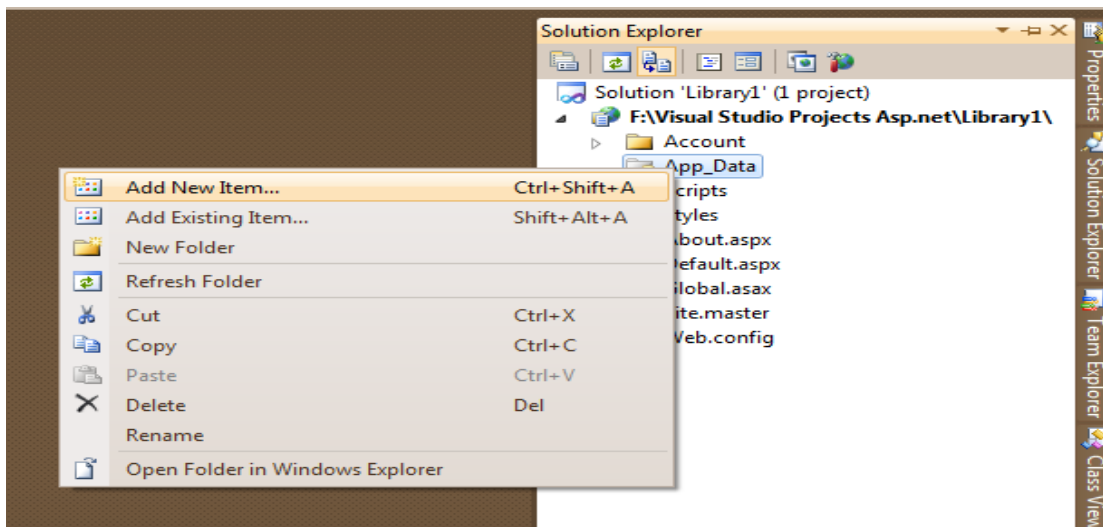
ننشئ مشروع

File >>New >> WebSite

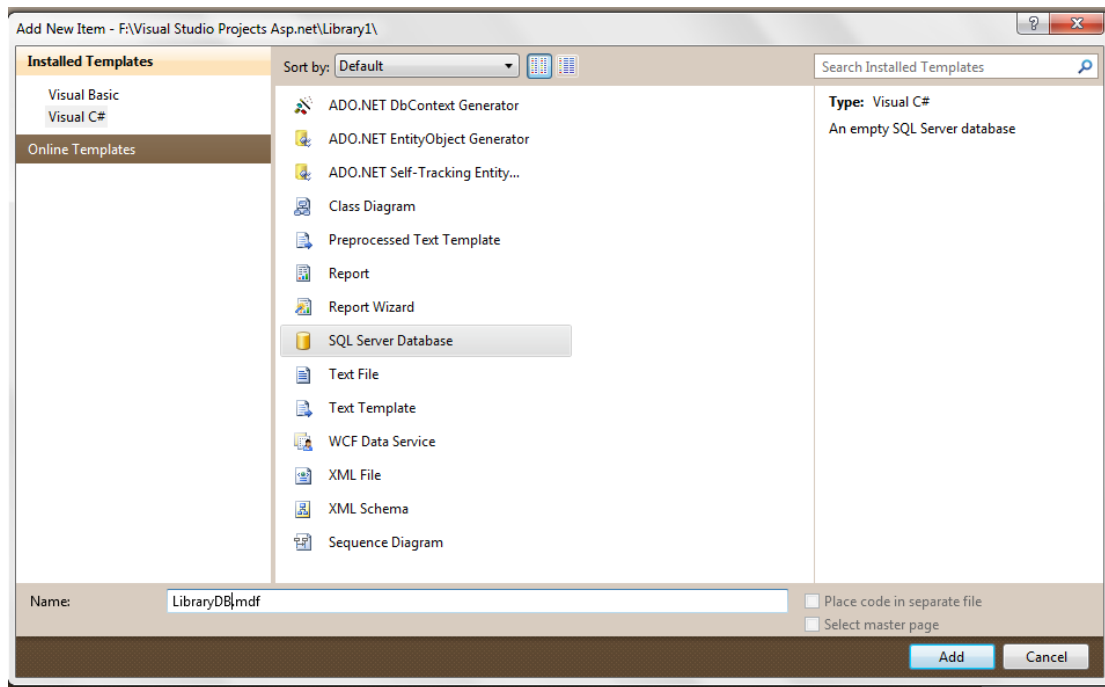
اسم المشروع Library .



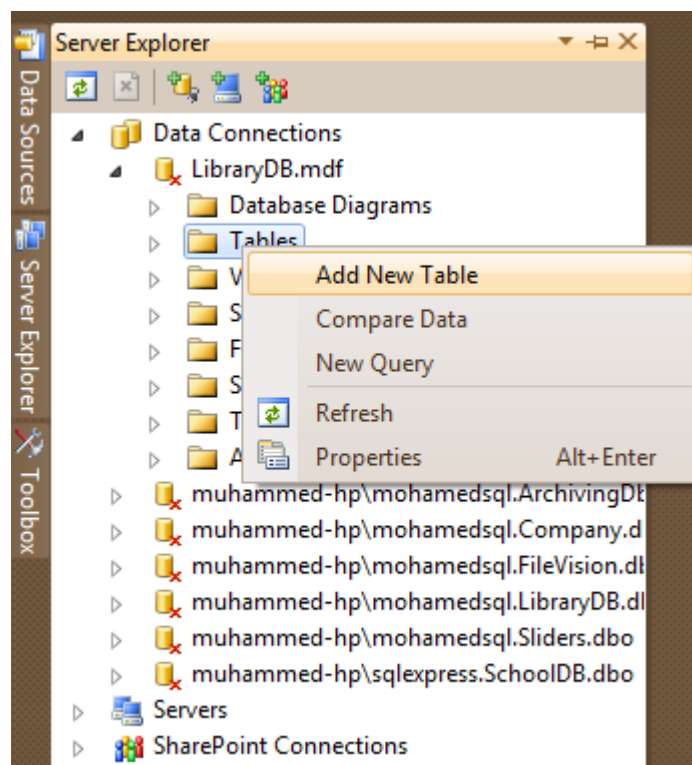
ننشئ قاعدة البيانات :



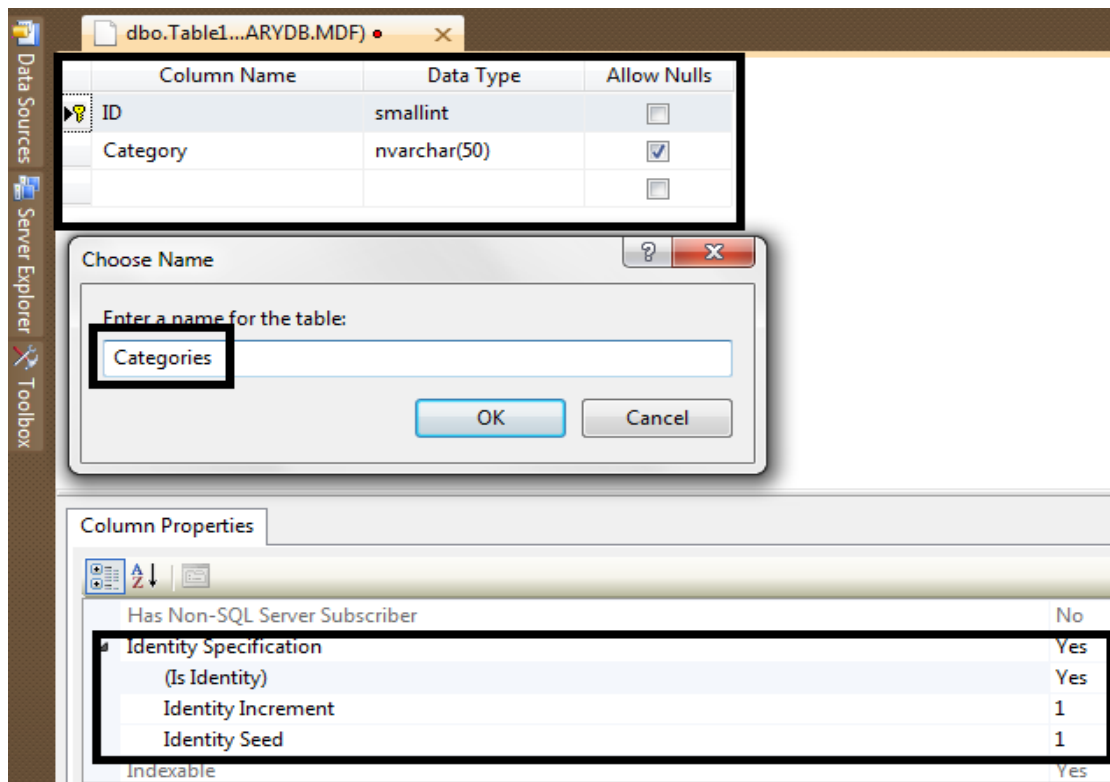
اسم القاعدة (LibraryDB) .



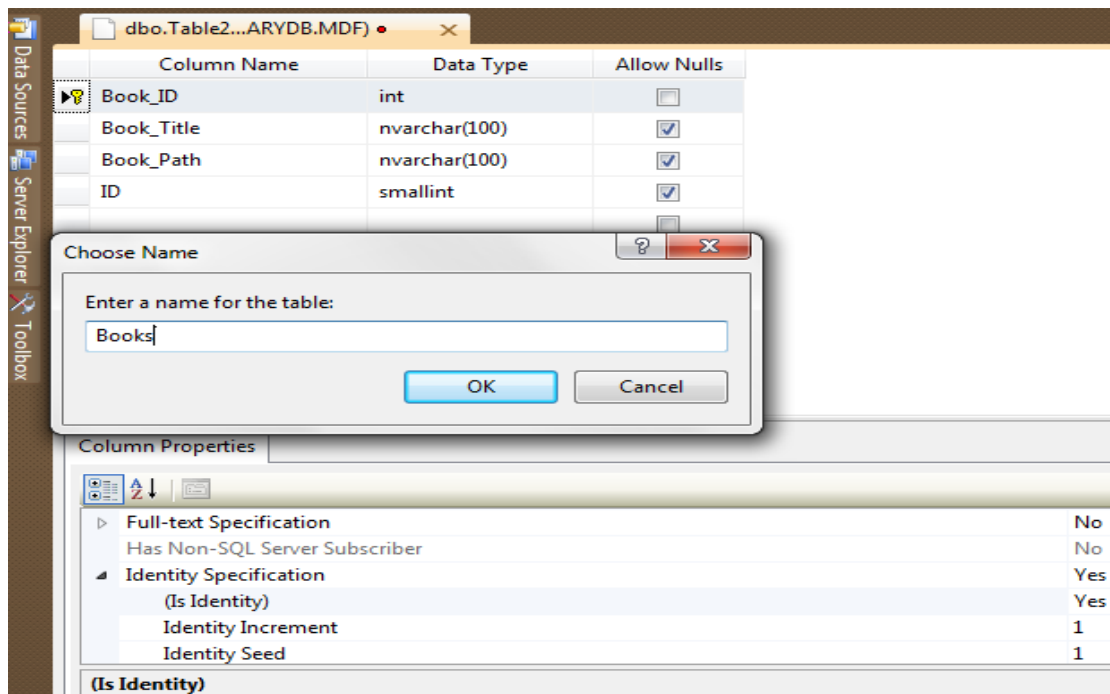
: Add New Table **Table** ونختار **Right Click** على



## الجدول الاول (Categories)

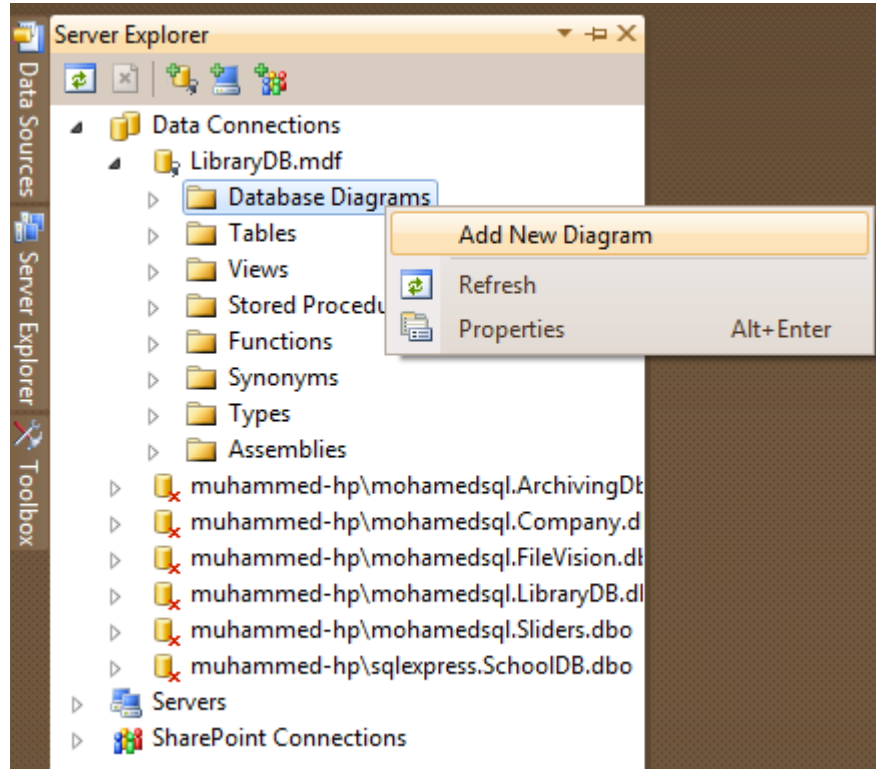


## نضيف جدول ثاني بأسم Books :

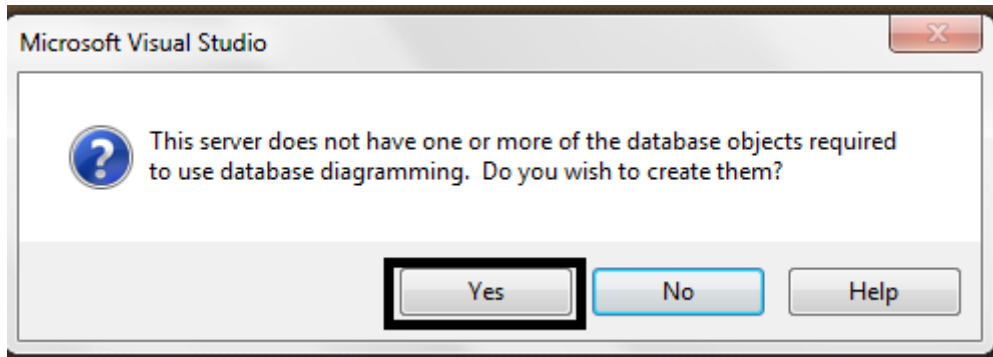


## لكي نضيف علاقة بين الجدولين (One To Many) :

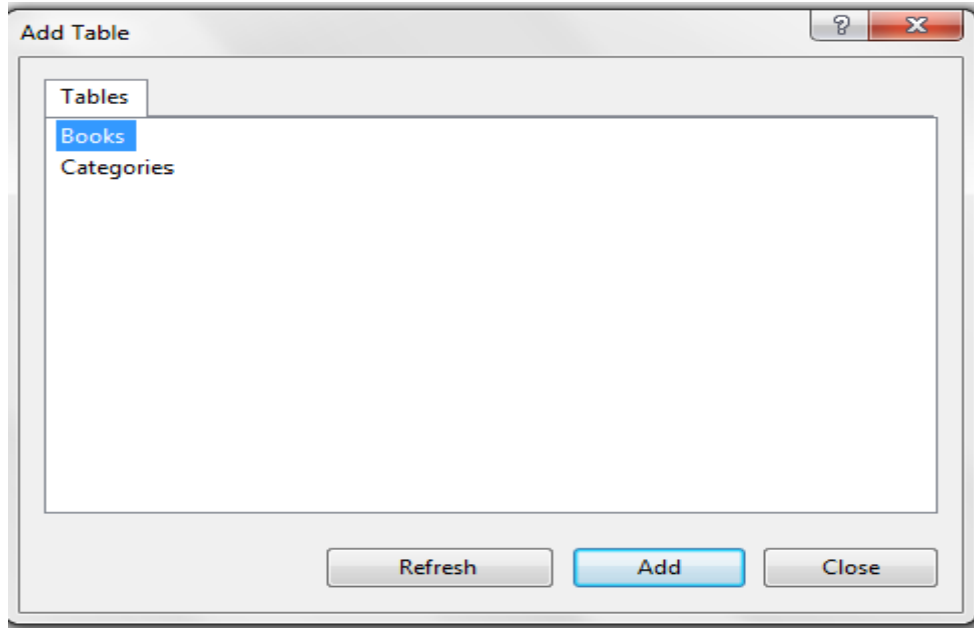
Right Click على DataBase Diagram ونختار Add New Diagram :



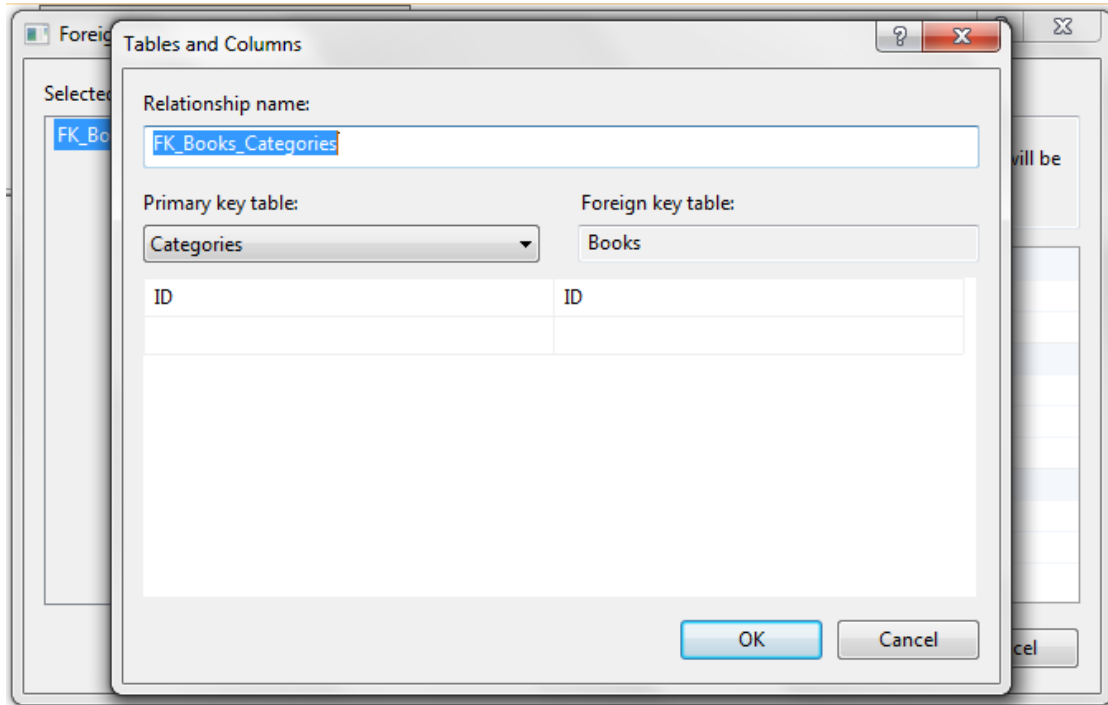
ستظهر لك رسالة تخبرك ان السيرفر لا يحوي على اي مخطط (Diagram) وهل تريد انشاء مخطط.....ننقر على Yes



تختار الجدول الاول وتنقر على Add ثم تختار الجدول الثاني وتنقر على Add ثم  
Close

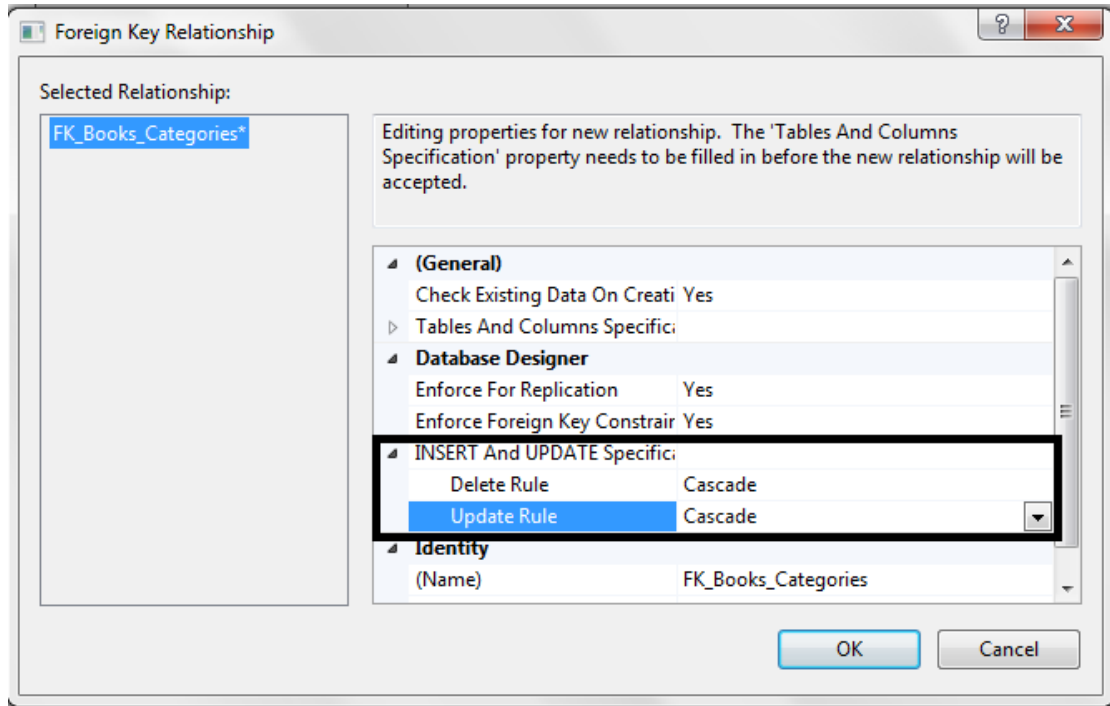


ثم نضع المؤشر على الحقل ID الموجود في الجدول Categories ونسحبه حتى نوصله الى الحقل ID في الجدول Books ستظهر لك :

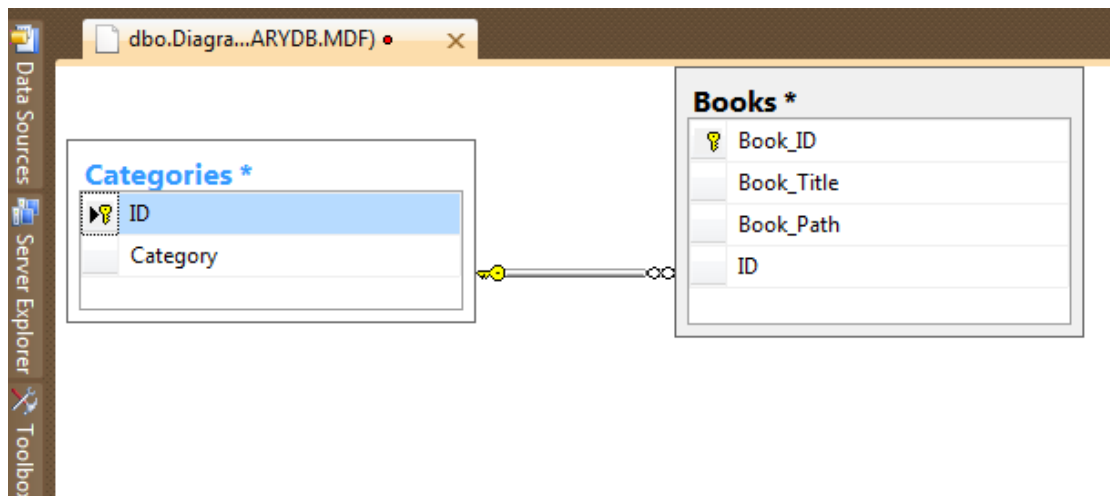


في الشاشة اعلاه يطلب تسمية العلاقة ،أتركها على اسمها الافتراضي وانقر على OK :

الشاشة أدناه تستعرض خصائص العلاقة فقط نغير بالخصائص Delete Rule و Update Rule نجعلهما Cascade بمعنى ان اي تغيير يحدث للجدول الاب (Categories) يؤثر على الجدول الابن (Books) مثلا حذف سجل سيحذف اي بيانات تابعة له في السجل الابن .



ثم ننقر على OK فستظهر لك العلاقة بهذا الشكل :

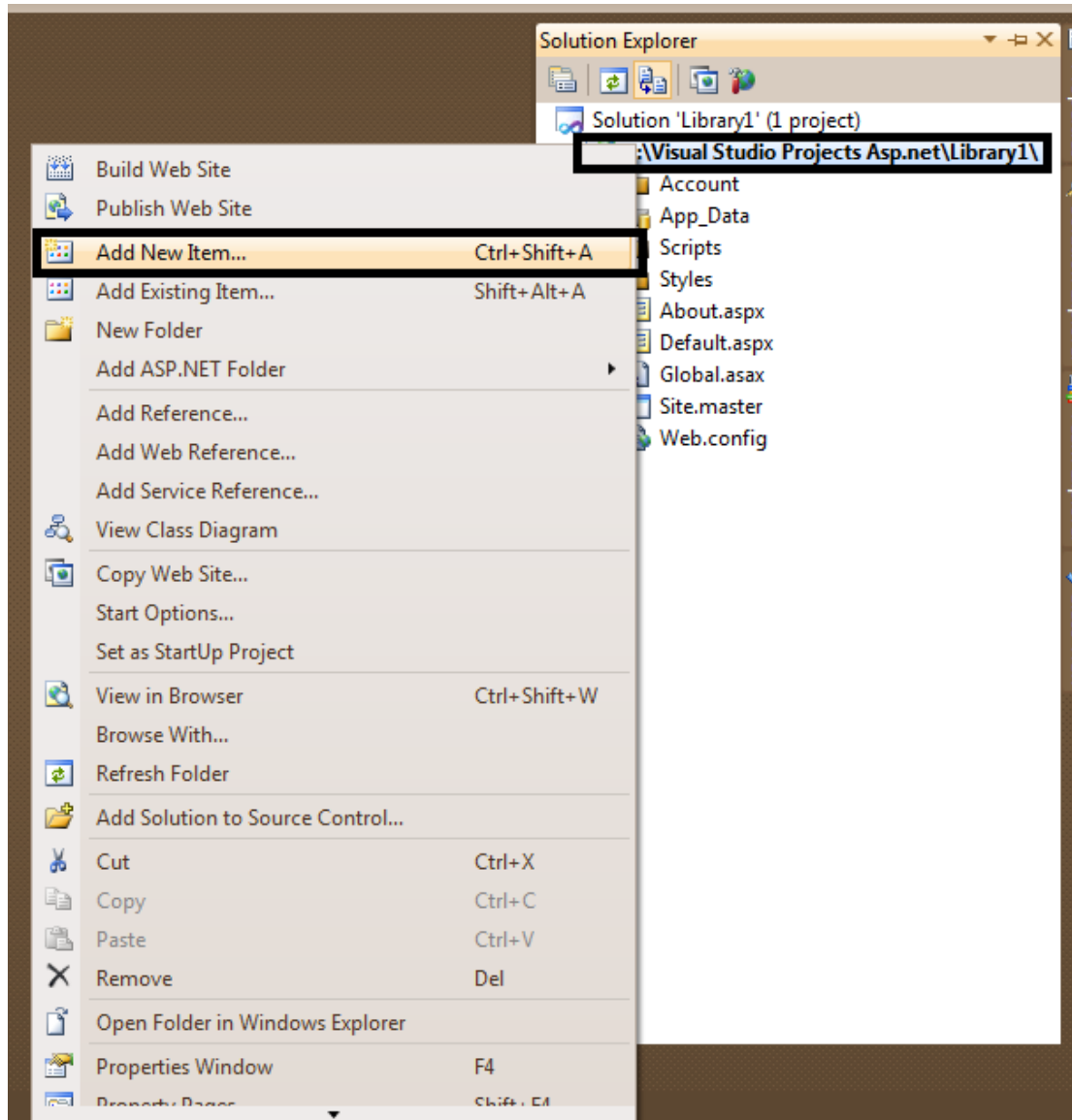


ثم اغلق الصفحة واحفظ المخطط (Diagram) .

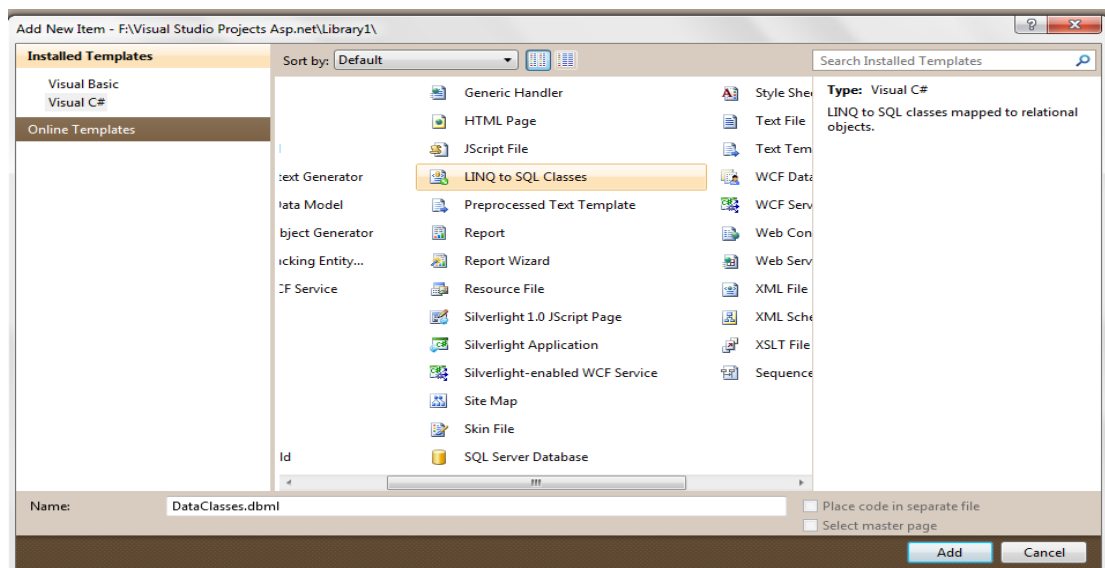
الان انتهينا من قاعدة البيانات .

الان جاء دور Linq ولاحظ معي السهولة وقارنها مع استخدامك ل ( Ado.net )

:

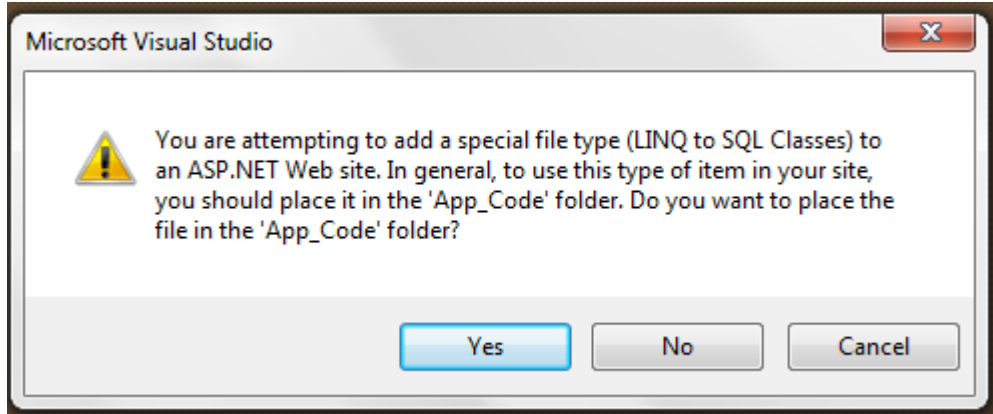


ثم نختار Linq To SQL Classes :

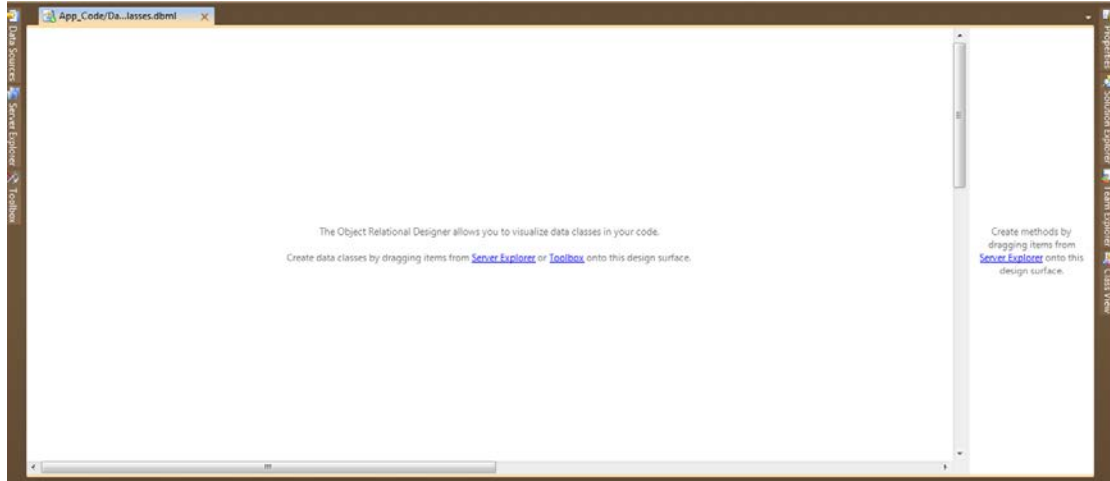




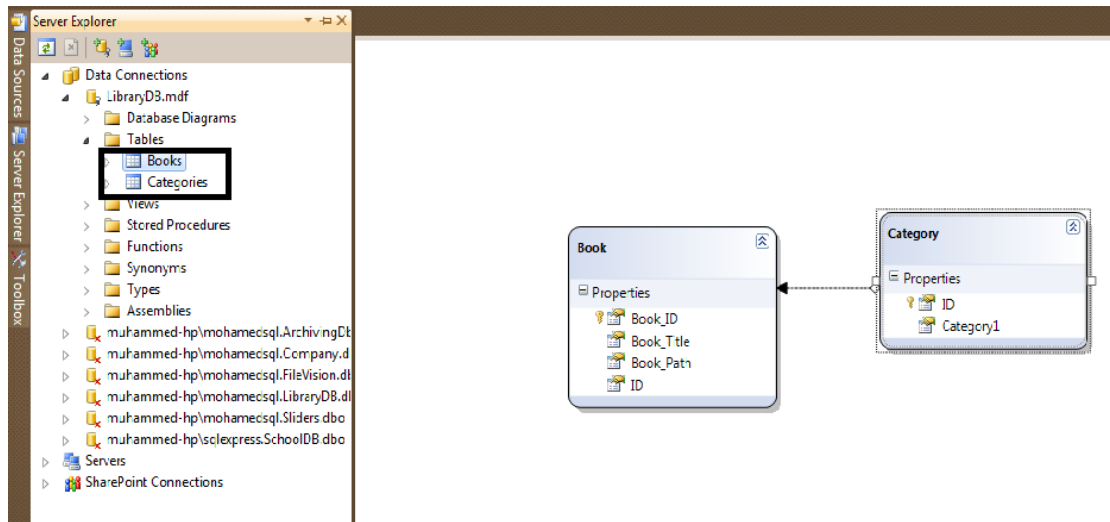
ستظهر لك الشاشة ادناه تخبرك هل تريد ان تضع الكلاس في مجلد App\_Code  
أختر Yes.



سيفتح لك الملف (DataClass.dbml) فارغ كما في الصورة أدناه



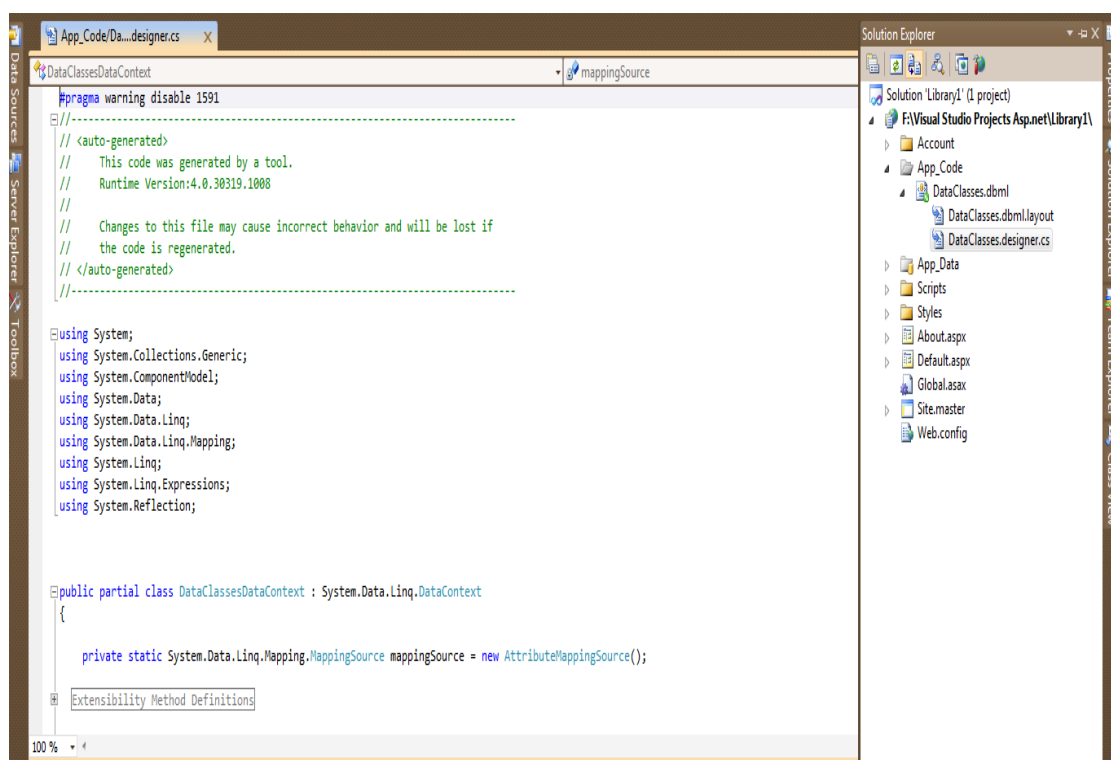
ثم من Server Explorer نبدأ بسحب الجدولين :



إذا تذكر اننا سمينا الجدولين ( Books ) و ( Categories ) وفي الصورة اعلاه أخذ الاسم المفرد لهما ( Book ,Category ) ولأن الجدول الاول اسمه ( Category ) لذلك اعاد تسمية الحقل ( Category ) الى ( Category1)!!!

عموما انقر على Save .

ولو فتحنا الملف ( DataClass.Designer.cs ) ستجد انه مثل الجداول في قاعدة البيانات على شكل كائنات (Class) وهذه هي الفائدة الرئيسية من الـ Linq To SQL



```
App_Code\DataClasses.designer.cs
DataClassesDataContext
mappingSource

#pragma warning disable 1591
//-----
// <auto-generated>
// This code was generated by a tool.
// Runtime Version:4.0.30319.1008
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
// </auto-generated>
//-----

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.Linq;
using System.Data.Linq.Mapping;
using System.Linq;
using System.Linq.Expressions;
using System.Reflection;

public partial class DataClassesDataContext : System.Data.Linq.DataContext
{
    private static System.Data.Linq.Mapping.MappingSource mappingSource = new AttributeMappingSource();
}

Extensibility Method Definitions
```

الى هنا انتهينا من بناء طبقة ( Business Layer ) هذه الطبقة وسيطة بين قاعدة البيانات والواجهات .

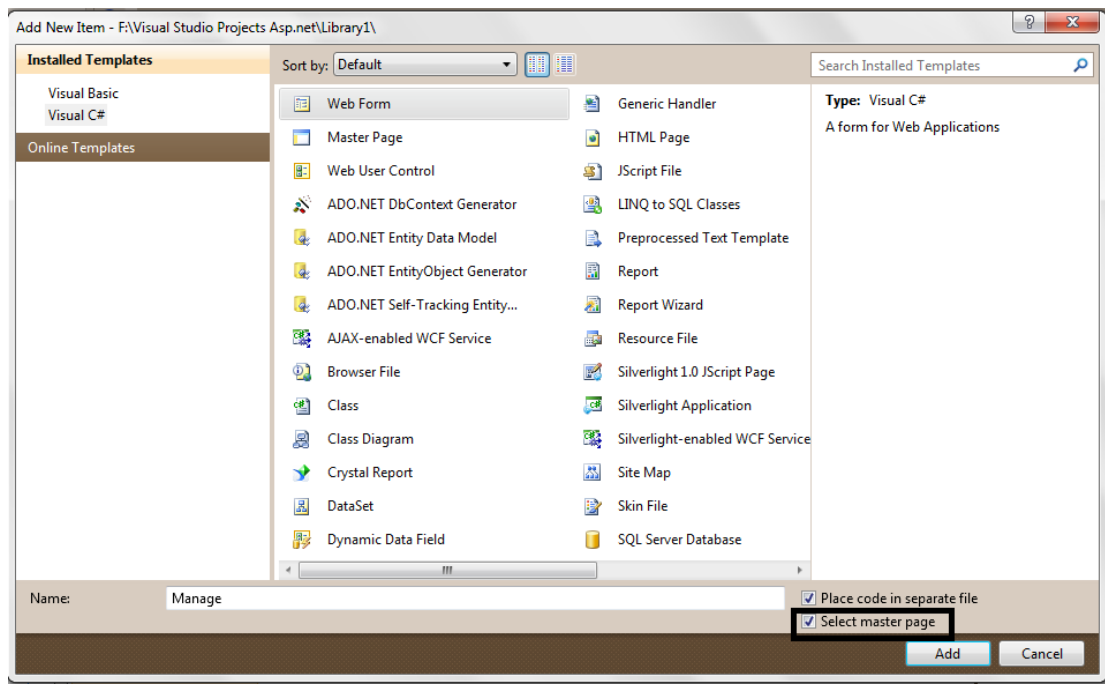
نفتح الصفحة Default.aspx وهي ستكون لعرض التصنيفات ونمسح الكلام الذي في داخلها :

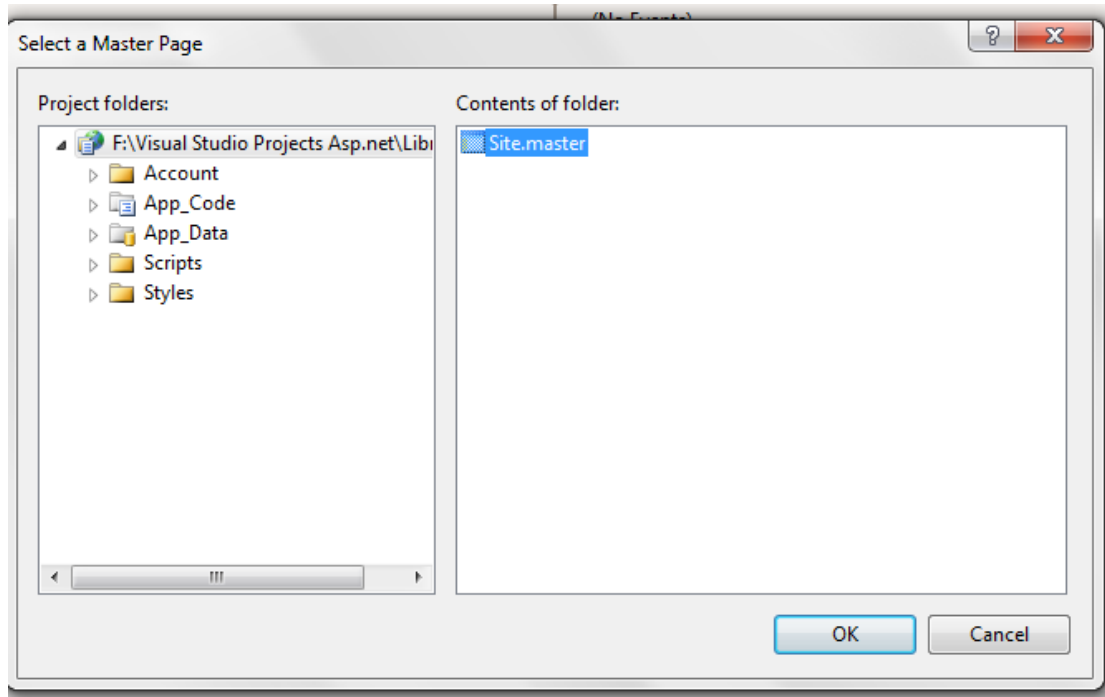
```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
<h2>
Welcome to ASP.NET!
</h2>
<p>
To learn more about ASP.NET visit <a href="http://www.asp.net" title="ASP.NET Website">www.asp.net</a>.
</p>
<p>
You can also find <a href="http://go.microsoft.com/fwlink/?LinkID=152368&cid=0x409"
title="MSDN ASP.NET Docs">documentation on ASP.NET at MSDN</a>.
</p>
</asp:Content>
```

ملاحظة : امسح فقط الكلام المحدد كما في الصورة اعلاه .

ثم نضيف صفحة أخرى اسمها ( Manage.aspx ) واجعلها تتوارث من الماستر  
بيج ( Site.master ) :





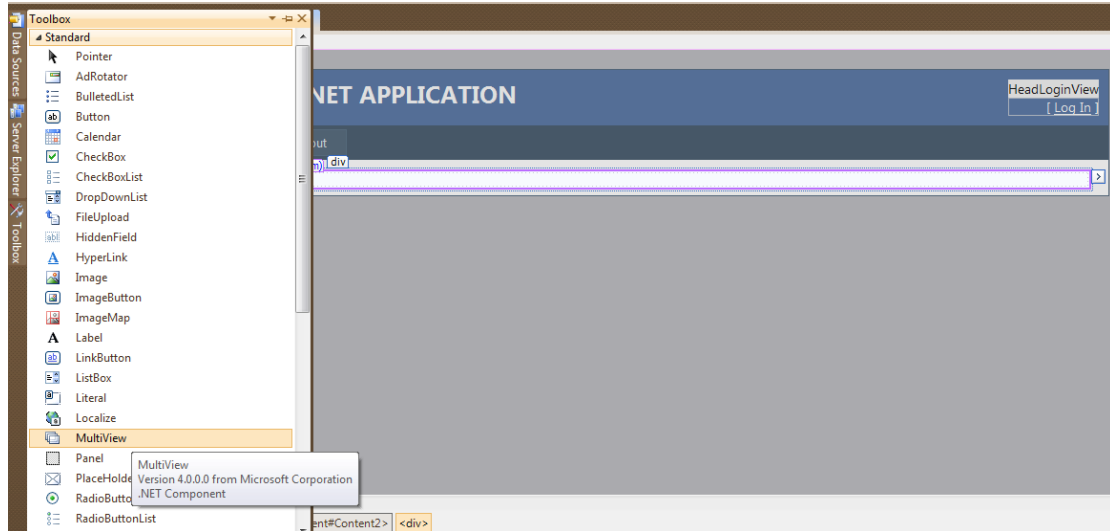
ثم على Ok انقر :

داخل الصفحة Manage.aspx نضيف ( Div ) ونجعل اتجاه من اليمين الى اليسار من اجل اللغة العربية :

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true" CodeFile="Manage.aspx.cs" Inherits="Manage" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<div dir="rtl" >
</div>
</asp:Content>
```

ونضيف داخله العنصر MultiView كما في الصورة أدناه :



وداخله نضيف العنصر View لغرض إضافة اصناف جديدة .

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true" CodeFile="Manage.aspx.cs" Inherits="Manage" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<div dir="rtl" >
    <asp:MultiView ID="MultiView1" runat="server"
">
        <asp:View ID="View1" runat="server">
        </asp:View>
    </asp:MultiView>
</div>
</asp:Content>
```

وداخل View1 نضيف جدول :



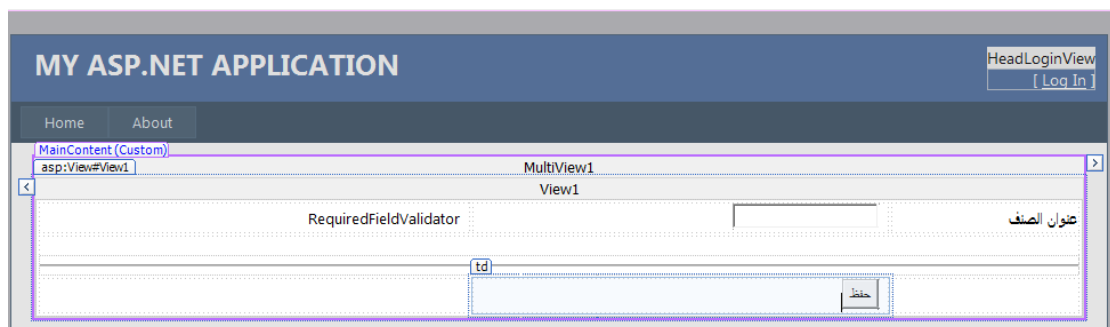
```

        &nbsp;
    </td>
    <td>
        &nbsp;
    </td>
</tr>
<tr>
    <td>
        &nbsp;
    </td>
    <td>
        &nbsp;
    </td>
    <td>
        &nbsp;
    </td>
</tr>
</table>
</asp:View>
</asp:MultiView>

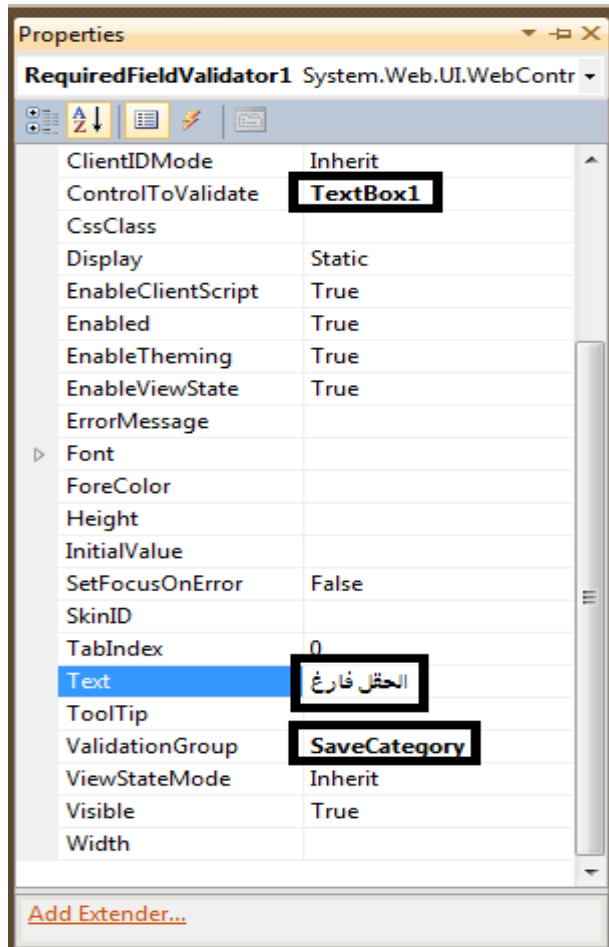
</div>
</asp:Content>

```

نبدأ بإضافة العناصر المهمة كما في الصورة أدناه :

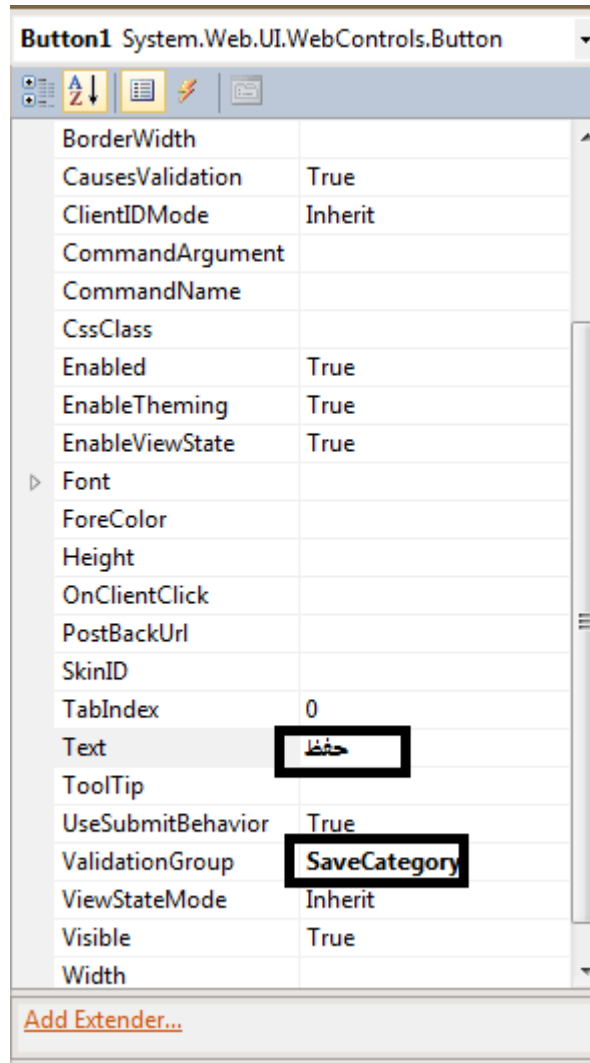


العنصر ( RequiredFieldValidator )



اما Button :





الغاية من استخدام Validation Group في البوتون والريكوارد فاليديتير هي ان الفاليديتير سيعمل مع اي عنصر يسبب PostBack وبما اننا سننظف أكثر من بوتون حيث ان البوتون يسبب PostBack لذلك عندما استخدمنا هذه الخاصية اصبح الفالديتير يعمل فقط مع هذا البوتون .

وهذا كود الصفحة

---

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true" CodeFile="Manage.aspx.cs" Inherits="Manage" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<div dir="rtl" >
```

---

---

```

<asp:MultiView ID="MultiView1" runat="server
">
    <asp:View ID="View1" runat="server">
        <table style="width: 100%;">
            <tr>
                <td>
                    الصنف عنوان
                </td>
                <td>
                    &nbsp;
                    <asp:TextBox ID="TextB
1" runat="server"></asp:TextBox>
                </td>
                <td>
                    &nbsp;
                    <asp:RequiredFieldValida
tor ID="RequiredFieldValida
tor ID="RequiredFieldValidator1" runat="server"
ControlToValidate="T
extBox1" ValidationGroup="SaveCategory">فارغ الحقـل
</asp:RequiredFieldValidator>
                </td>
            </tr>
            <tr>
                <td colspan="3">
                    &nbsp; &nbsp; &nbsp;
                    <hr />
                </td>
            </tr>
            <tr>
                <td>
                    &nbsp;
                </td>
                <td>
                    &nbsp;
                </td>
                <td>
                    &nbsp;
                    <asp:Button ID="Button1"
runat="server" Text="حفظ"
ValidationGroup="Sav
eCategory" />
                </td>
            </tr>
            <tr>
                <td>
                    &nbsp;
                </td>
            </tr>
        </table>
    </asp:View>
</asp:MultiView>

```

---

```
</tr>
</table>
</asp:View>
</asp:MultiView>
```

```
</div>
</asp:Content>
```

لو فتحت ملف Web.Config ستجده اضاف نص الاتصال مع قاعدة البيانات بأسم  
: ( LibraryDBConnectionString)

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <connectionStrings>
    <add name="ApplicationServices" connectionString="data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDbFilename=|DataDirectory|\aspnetdb.mdf;User Instance=true"
      providerName="System.Data.SqlClient" />
    <add name="LibraryDBConnectionString" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\LibraryDB.mdf;Integrated Security=True;User Instance=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="false" targetFramework="4.0">
    <assemblies>
```

نعود الى الصفحة Manage.aspx ونقرتين على البوتون حتى يفتح صفحة الكود الخلفي وداخل الحدث ( Click ) للبوتون نضيف الكود أدناه :

```
protected void Button1_Click(object sender, EventArgs e)
{
    using (DataClassesDataContext ctx = new DataClassesDataContext() )
    {
        ctx.Connection.ConnectionString = System.Configuration.ConfigurationManager.ConnectionStrings["LibraryDBConnectionString"].ConnectionString;
        Category categ = new Category() ;
        categ.Category1 = TextBox1.Text ;
        ctx.Categories.InsertOnSubmit(categ );
        ctx.SubmitChanges();
    }
}
```

---

---

الكود اعلاه عرفنا فيه DataContext وهي عبارة عن كائن يدير الاتصال مع قاعدة البيانات من خلال فتح وغلق الاتصال وكذلك تنفيذ الاستعلامات.

واعطيناه نص الاتصال ثم عرفنا كائن اخر من كائن الجدول Category واسندنا القيمة من التيكس بوكس الى الخاصية Category1 الموجودة في الكائن Category ( يعني حقل Category في الجدول Categories ) ثم طبقنا استعلام الاضافة و ثم حفظ المتغيرات التي اجريناها على DataContext هذا كل ما في الامر جرب ....

ولا تنسى ان تقوم بجعل المولتيفيو يستعرض View1 عند تحميل الصفحة :

---

```
protected void Page_Load(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = 0;
}
```

---

نكمل ان شاء الله في الدرس الثاني باقي العمليات مثل استرجاع البيانات ( Select ) والحذف والتعديل .

تحياتي لكم

محمد الساعدي

<http://mohamediddan.wordpress.com/>