

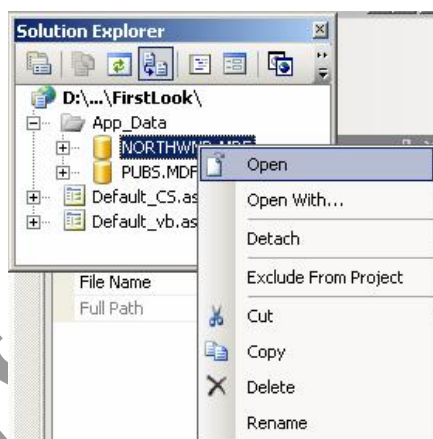
الفصل الرابع : قواعد البيانات و أدوات عرض البيانات.

Database, ADO.net and data show tools.

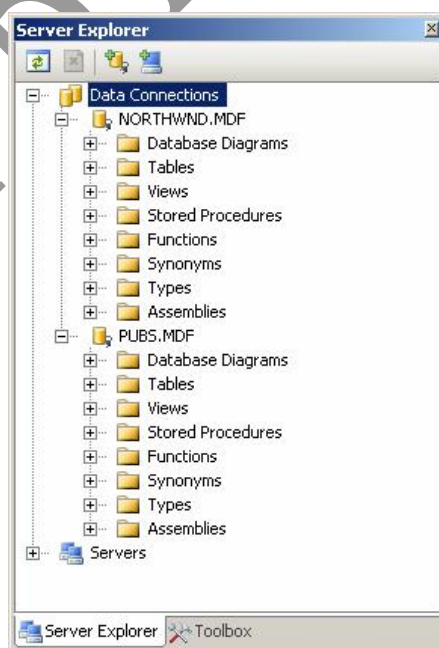
فى هذا الفصل و الذي أعتبرة من أهم الفصول التى تهتم كل مطور يعمل فى السوق و يدري ما هى متطلبات السوق، و الآن سوف نبدأ ببعض الأدوات الجديدة لعرض البيانات و الأمكانيات الجديدة فى هذه الأدوات و بعد ذلك سوف نرى ما هو الجديد فى ADO.net و الآن لنبدأ مع أدوات التعامل مه البيانات الجديدة،



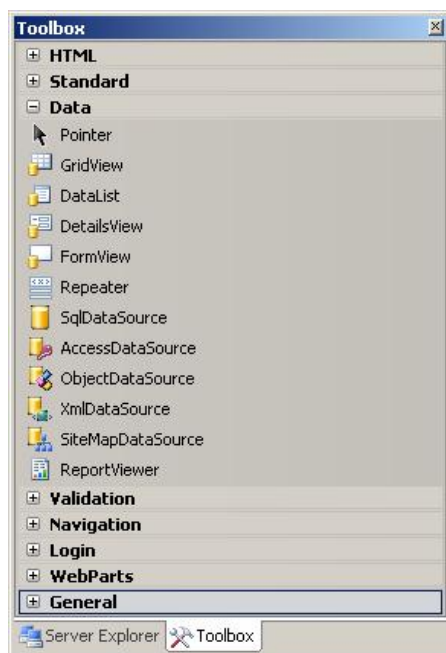
الشكل ١ أولاً عليك البدء بمشروع و تضع الملفات كما هو موضح أعلاه



الشكل ٢ نقوم بفتح قاعدة البيانات للتأكد من أن الاتصال سليم

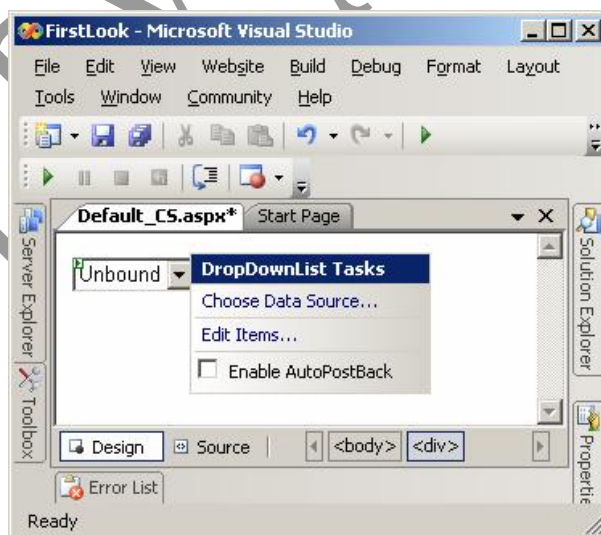


الشكل ٣ عند بدأ الاتصال سوف نرى أن كل قاعدة بيانات تظهر فى مجلد بنفس أسم قاعدة البيانات

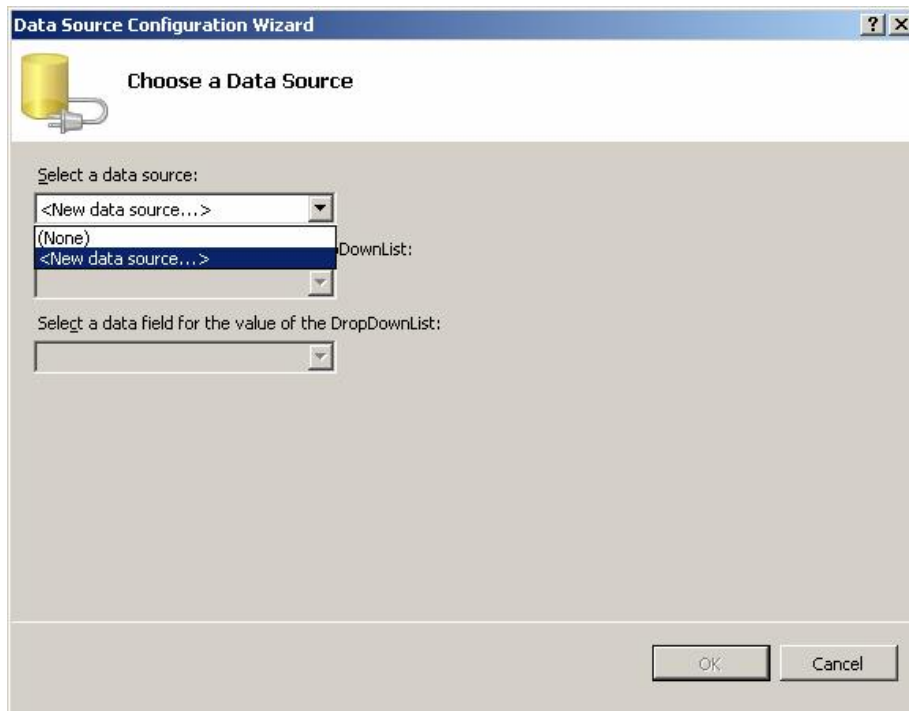


الشكل ٤ : هذه هي مجموعة الادوات الخاصة بالتعامل مع قواعد البيانات – لنن نتعامل مع كل الادوات و لكن سوف نتعامل مع الأدوات التي تحتوى على شئ جديد و الان لنبدأ المثال

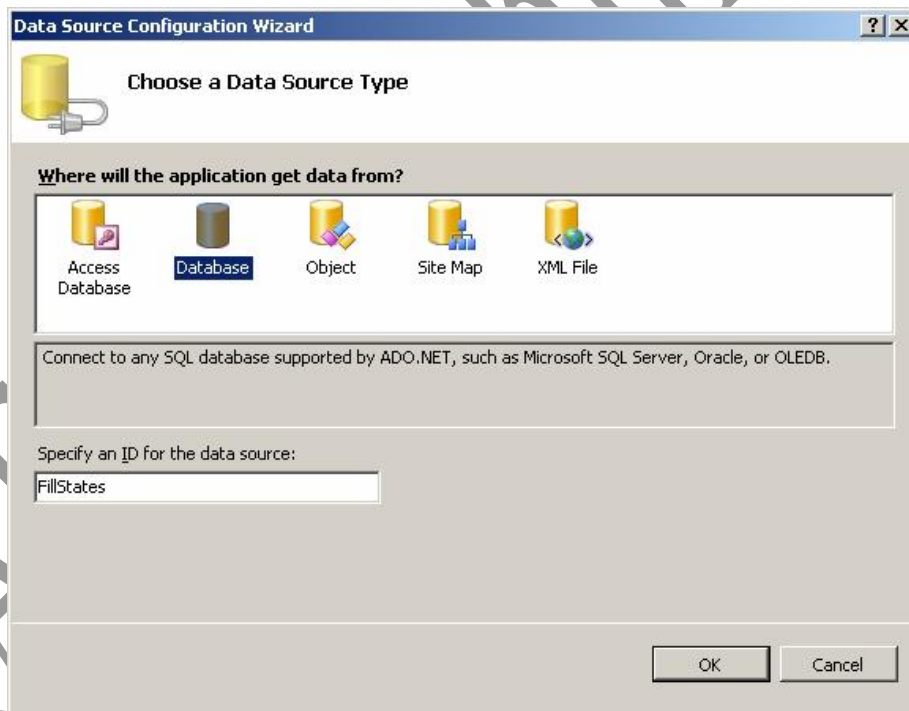
المثال الذى سوف نقوم بتطبيقه الان أولا سوف يتم على قاعدة البيانات Pubs و هي قاعدة بيانات خاصة بدور النشر و المؤلفين و الكتب و ما الى ذلك، فى المثال التالى سوف نوضح كيفية التعامل مع الاداة الجديدة Grid View و هذه الاداة تعتبر بديل للـ Data Grid و هي أداة أكثر من رائعة بالإضافة الى الاداة الجديدة Details View و الان لنبدأ المثال.



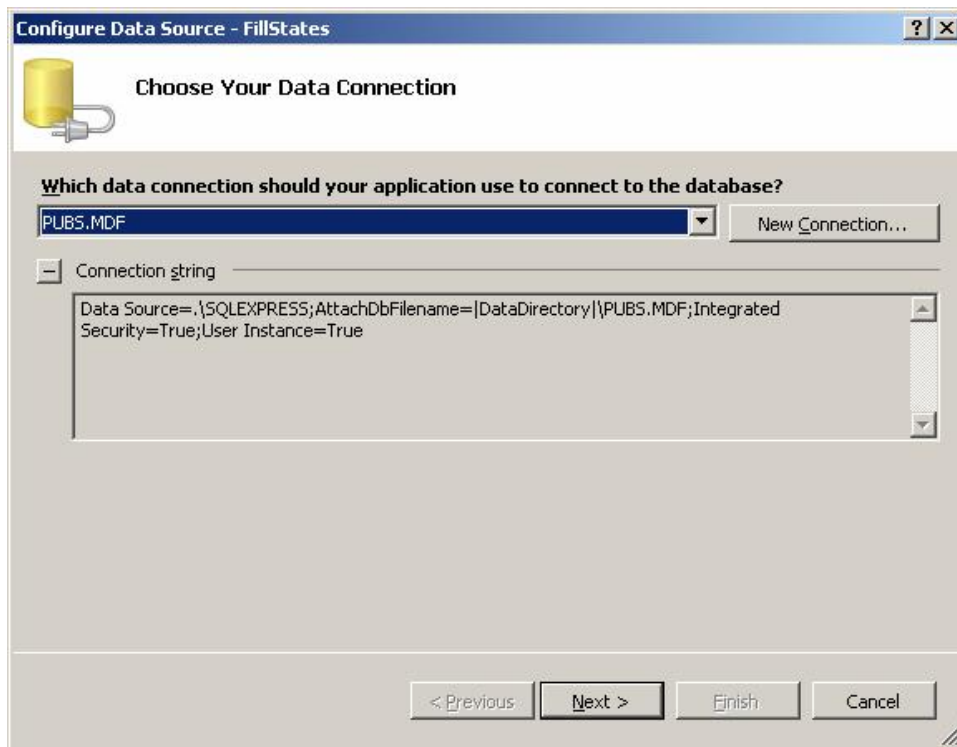
الشكل ٥ : أولا نقوم بوضع DropDownList كما هو موضح و نضغط على الـ Smart Tag و منه نختار Choose Data Source



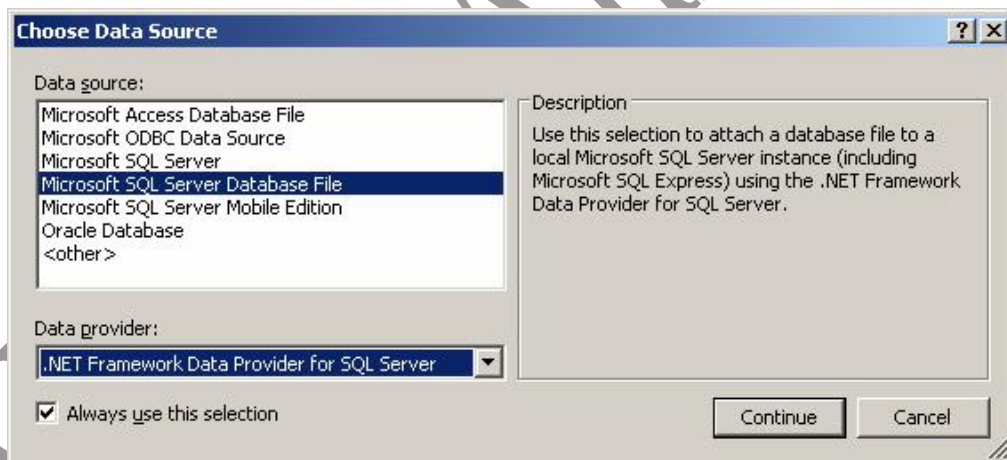
الشكل 6 سوف تظهر لنا شاشة معالج ضبط و أعداد مصدر البيانات و كما نرى الآن سوف نختار مصدر بيانات جديد New Data source



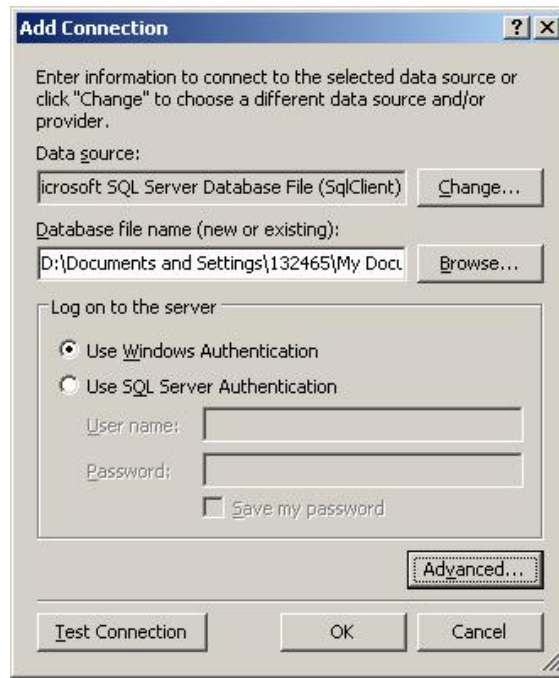
الشكل 7 نقوم باختيار نوع قاعدة البيانات و هذا طبقاً لمحرركات قواعد البيانات الموجودة Database Provider ثم بعد ذلك نكتب اسم مصدر البيانات و نضغط على OK



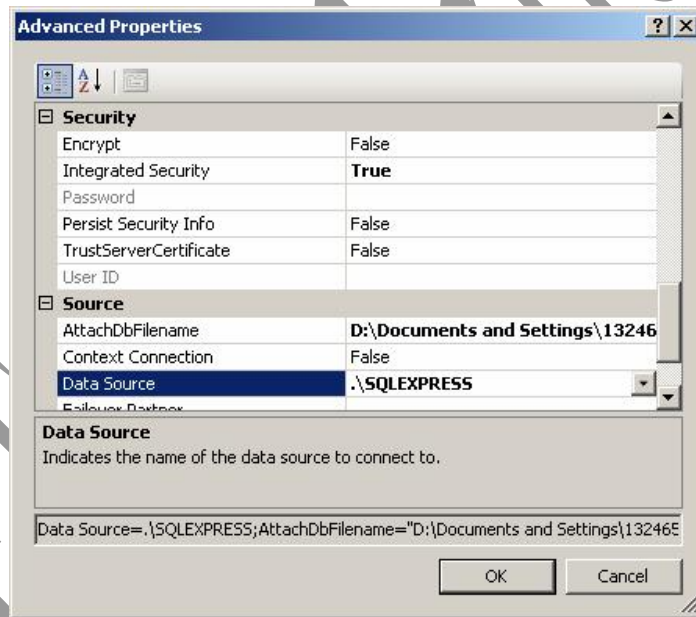
الشكل ٨ الآن سوف ننشء جملة الاتصال بكل بساطة اختر اسم قاعدة البيانات و سوف يقوم المعالج بأكمل الموضوع دون قلق أما اذا كنت تود ضبط بعض الإعدادات الخاصة أضغط على **New Connection**



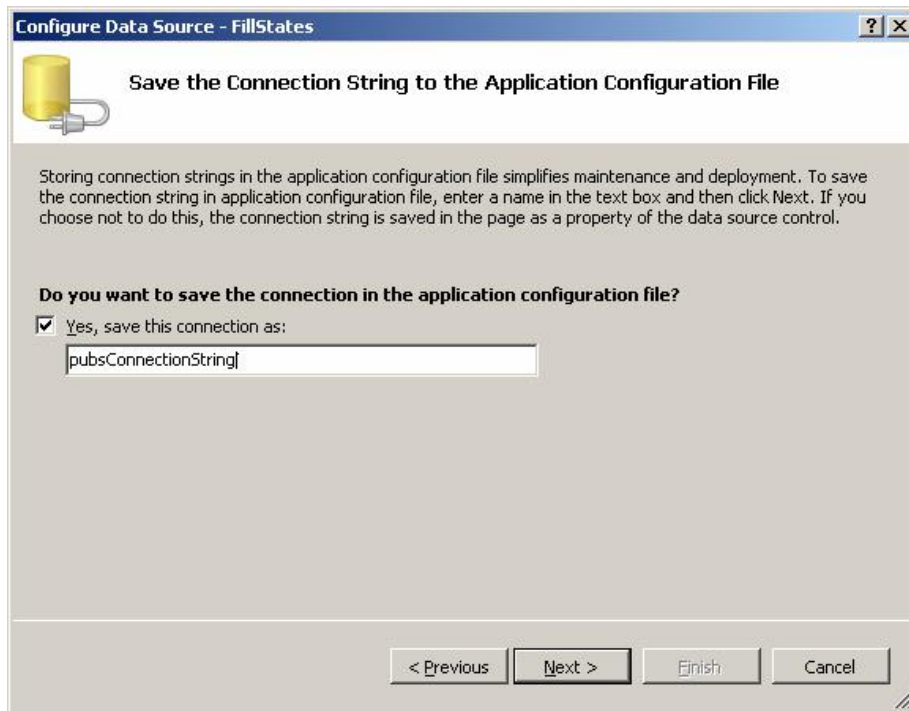
الشكل ٩ سوف تظهر هذه النافذة و التي سوف تختار منها محرك قاعدة البيانات الذي تود التعامل معه و يعد ذلك أضغط على **Continue**



الشكل ١٠ هنا تختار ملف قاعدة البيانات و بعد ذلك تقوم بالضغط على **Advanced** لضبط الإعدادات المتقدمة



الشكل ١١ و هنا شاشة خصائص الإعدادات المتقدمة بعد الانتهاء اضغط على **OK** سوف تعود الى شاشة **Add Connection** و منها اضغط على **Test Connection** للتأكد من سلامة الاتصال و عدم وجود مشاكل



الشكل 12 تأتي هذه الشاشة بميزه جديدة و جبارة و توفير كبير للوقت و هي أن المعالج يخبرك بين حفظ جملة الاتصال في ملف Web.config أو أن يقوم بتركها في خصائص أداة مصدر البيانات، قم باختيار Yes و بعد ذلك أدخل الاسم التي سوف تحفظ به

```
<connectionStrings>
  <add name="pubsConnectionString"
        connectionString="Data Source=.\SQLEXPRESS;
        AttachDbFilename=|DataDirectory|\PUBS.MDF;
        Integrated Security=True;User Instance=True"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

هذا هو شكل جملة الاتصال في الملف Web.config

```
<asp:SqlDataSource
  ID="FillStates"
  runat="server"
  ConnectionString="<%$ ConnectionStrings:pubsConnectionString %>"
  SelectCommand="SELECT DISTINCT [state] FROM [authors] ORDER BY [state]">
</asp:SqlDataSource>
```

هذا هو عرض Source view للصفحة و هذا الجزء للأداة SQL Data Source و فيها خاصية جملة الاتصال و كما نرى أنها ConnectionStrings:pubsConnectionString و هذا يعني أنه سوف يحضرها من الملف Web.config ليقوم بعملية الاتصال.

Configure Data Source - FillStates

Configure the Select Statement

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name:

Columns:

☐ *	☐ city
☐ au_id	☒ state
☐ au_lname	☐ zip
☐ au_fname	☐ contract
☐ phone	
☐ address	

☒ Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:

SELECT DISTINCT [state] FROM [authors]

< Previous Next > Finish Cancel

الشكل ١٣ و الآن سوف نكمل بعد أن قمنا بأعداد أداة الاتصال- الآن نقوم بأعداد البيانات التي سوف تعرضها الأداة و كما نرى أننا اخترنا أن نقوم بعض الأعمدة من الجداول و الآن شوف نختار العمود المراد عرضه كما نرى هو State و سوف نقوم بتفعيل Return only unique row و هذا لمنع التكرار في النتائج بعد ذلك قم بالضغط على Order by

Add ORDER BY Clause

Specify the columns you would like to order by.

Sort by: ☒ Ascending ☐ Descending

Then by: ☒ Ascending ☐ Descending

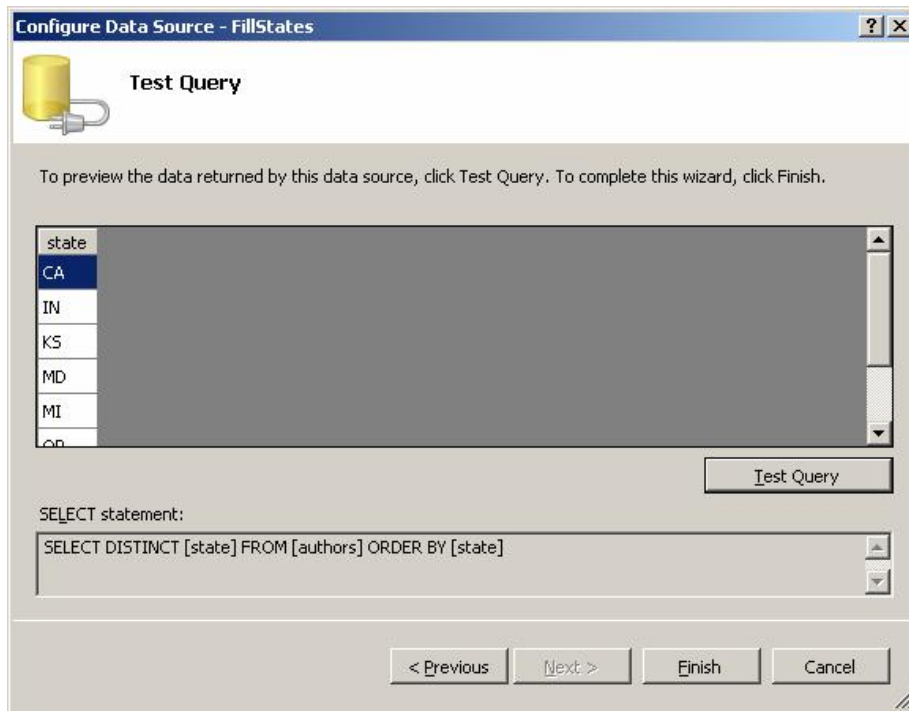
Then by: ☒ Ascending ☐ Descending

SELECT statement:

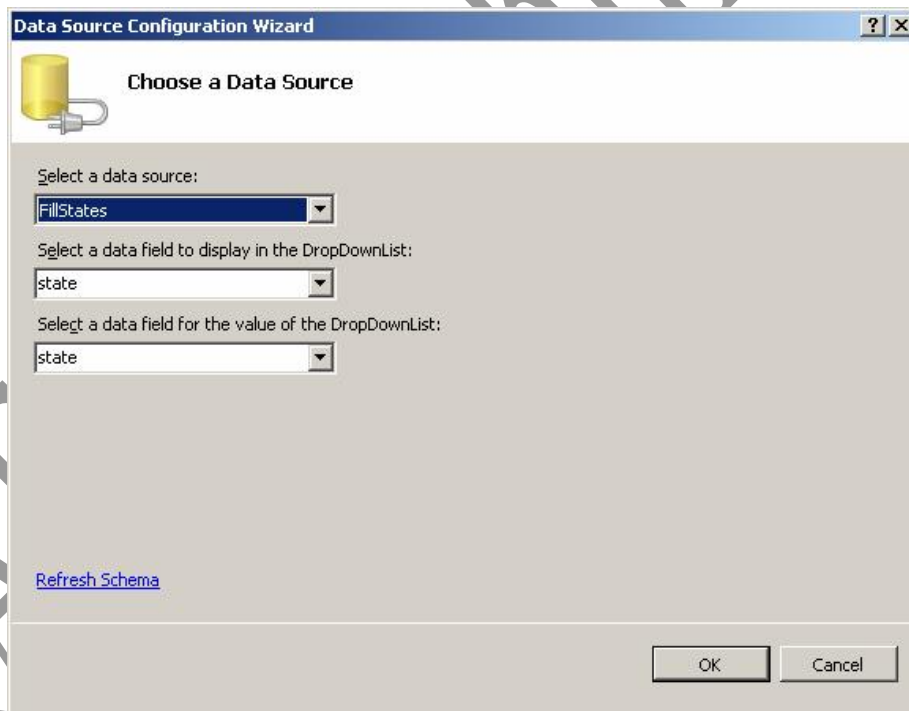
SELECT DISTINCT [state] FROM [authors] ORDER BY [state]

OK Cancel

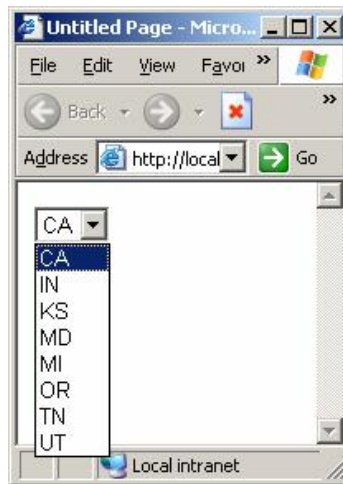
الشكل ١٤ هنا نختار ترتيب عرض البيانات أضغط على OK سوف تعود الى الشكل السابق ثم أضغط على Next



الشكل 15 و الآن اختيار جملة الاستعلام للتأكد من سلامتها



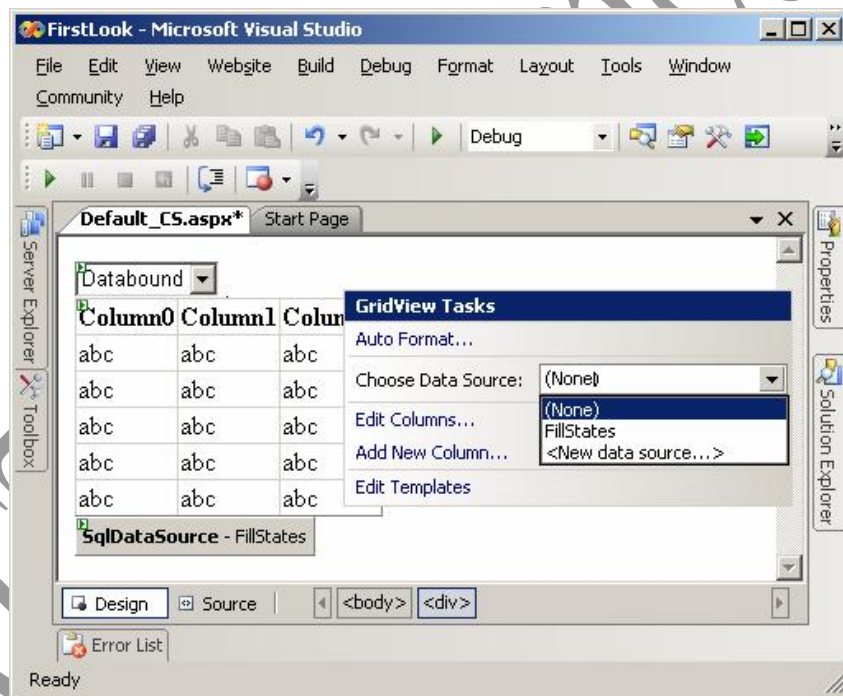
الشكل 16 و هنا يتم تحديد كلا من العمود الذي سوف يتم بع عرض البيانات **Displayed value** و العمود الذي سوف يتم فيه وضع القيمة **Value of selected item**



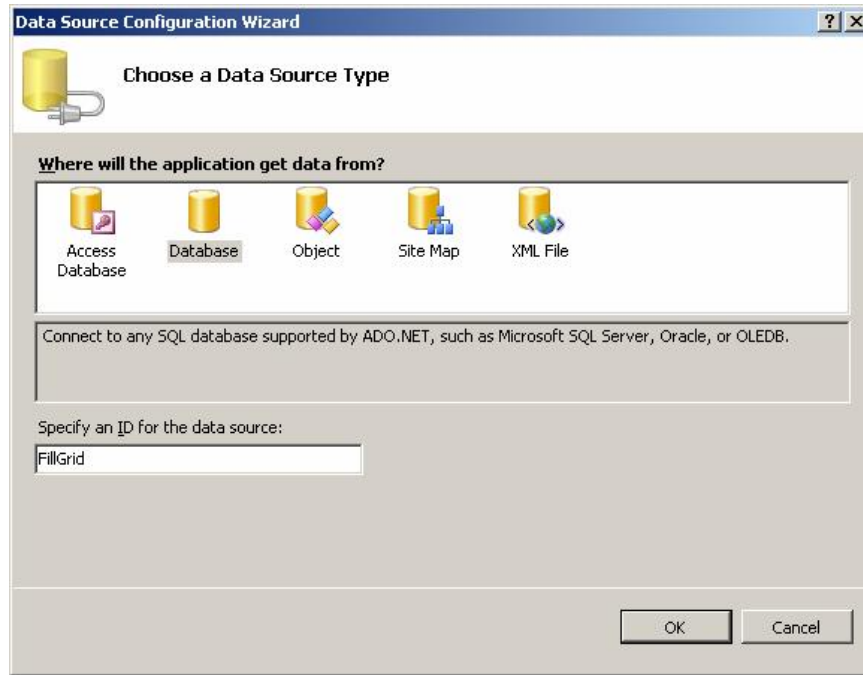
الشكل ١٧ و هذا هو المثال بعد التنفيذ



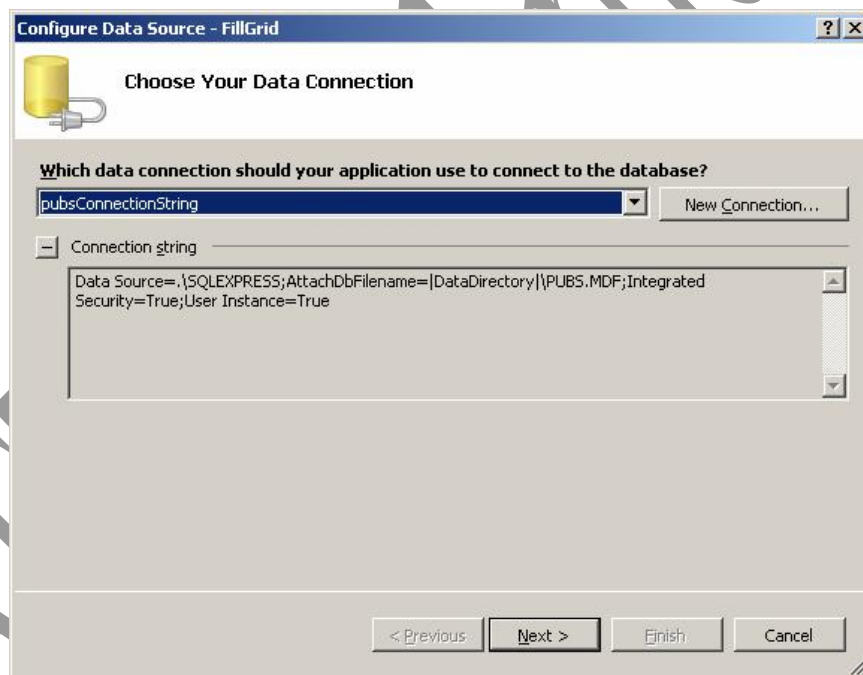
الشكل ١٨ و الآن نقوم بوضع الأداة GridView على الصفحة لنبدأ التجربة الفعلية لأدوات البيانات الجديدة



الشكل ١٩ نقوم بفتح Smart Tag و نختار من Choose data source مصدر بيانات جديد New Data source كما هو موضح في الشكل



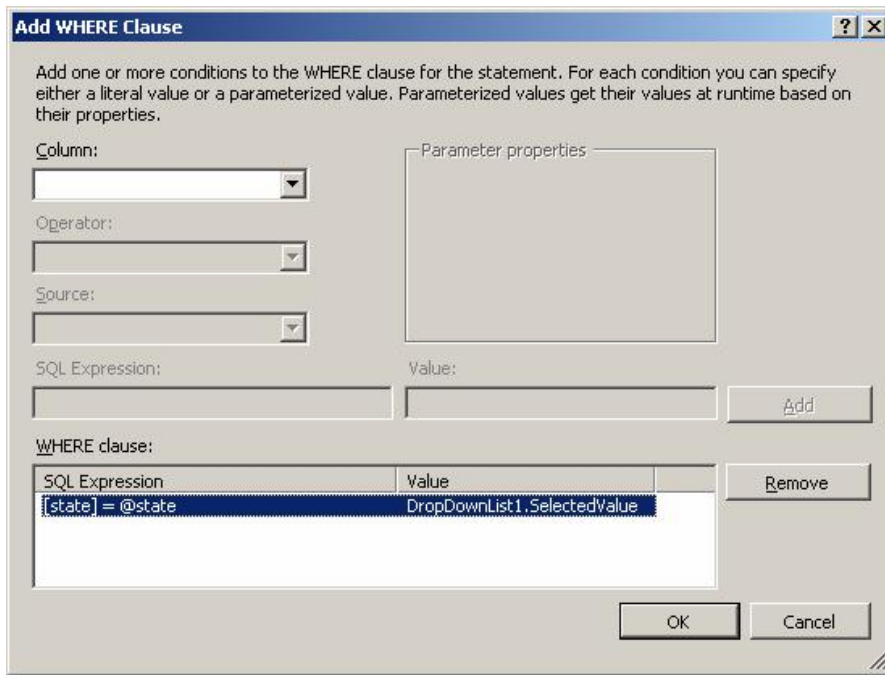
الشكل ٢٠ نقوم باختيار نوع المحرك و أسم أداة مصدر البيانات



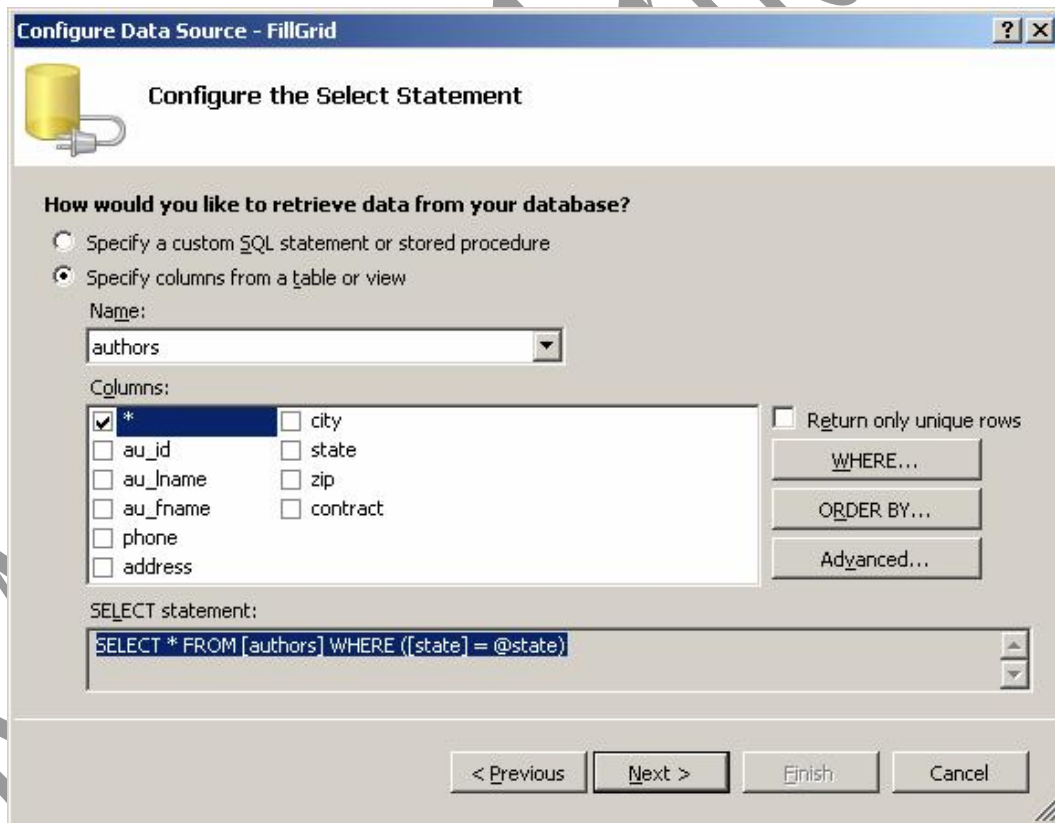
الشكل ٢١ هنا سوف نستعمل جملة الاتصال المحفوظة من قبل في Web.config

الشكل ٢٢ هنا نختار الأعمدة التي سيتم عرضها بعد ذلك نضغط على Where

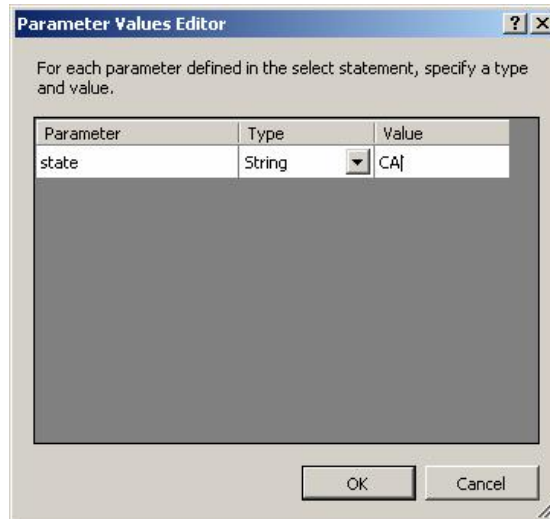
الشكل ٢٣ و الآن هذه النافذة سوف توفر علينا الكثير جدا جدا من الوقت حيث أننا الآن سوف نقوم بربط DropDownList بالGridView و بكل بساطة أختار العمود الذي سوف يتم عليه الشرط ثم أختار مصدر قيمة العمود Source و بعد ذلك قم بأختيار الأداة التي سوف يتم أخذ القيمة منها بعد ذلك أضغط على ADD



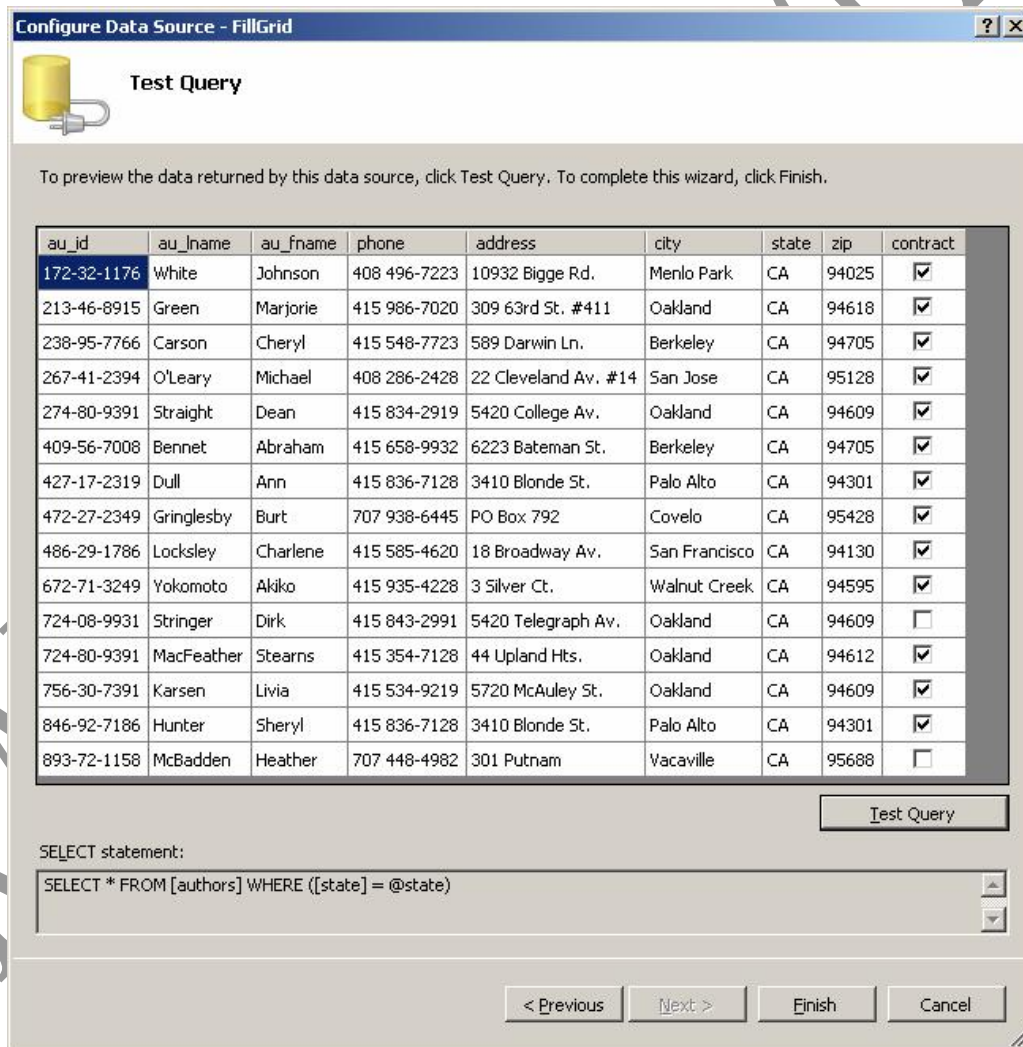
الشكل ٢٤ أضغط على OK و بذلك تكون قد قمت ببناء الشرط و ربطه بالأداة



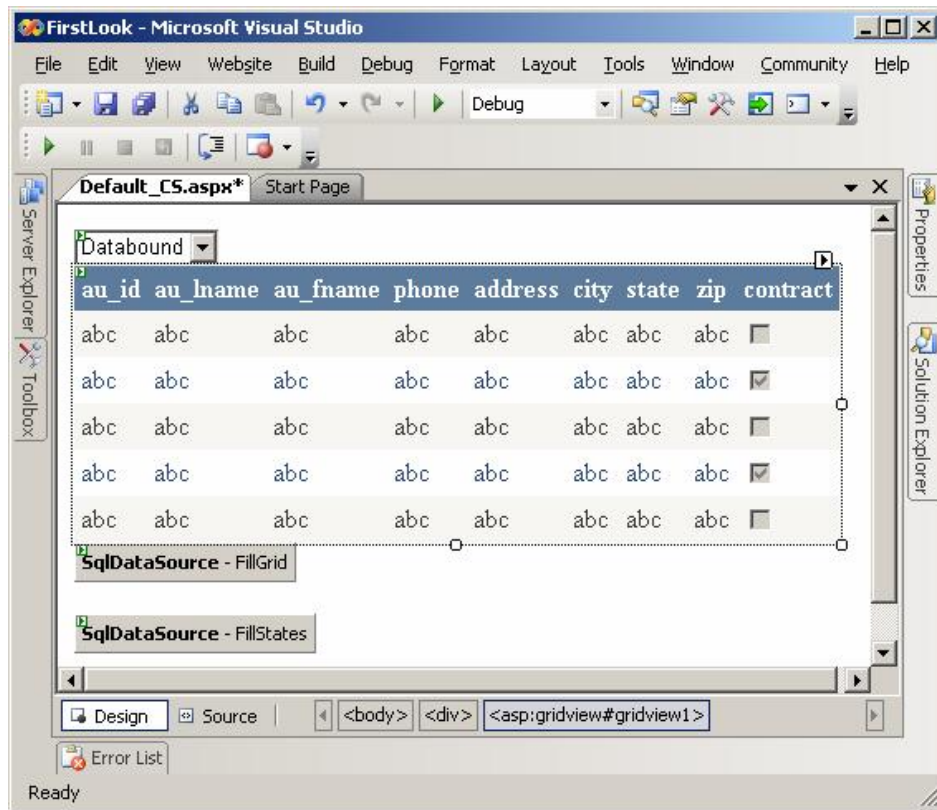
الشكل ٢٥ هذا هو الشكل النهائي بعد تحديد الشرط و الاداة مصدر قيمة الشرط أضغط Next



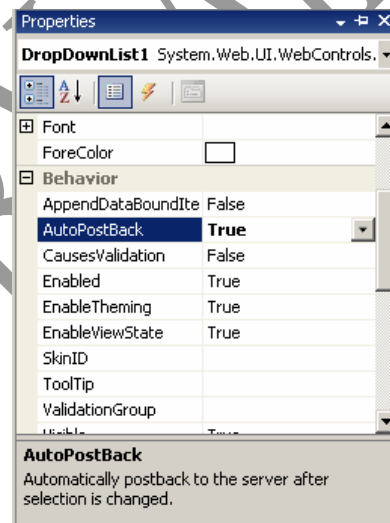
الشكل ٢٦ هذه الشاشة تقوم بادخال قيمة لأختبار جملة الاستعلام



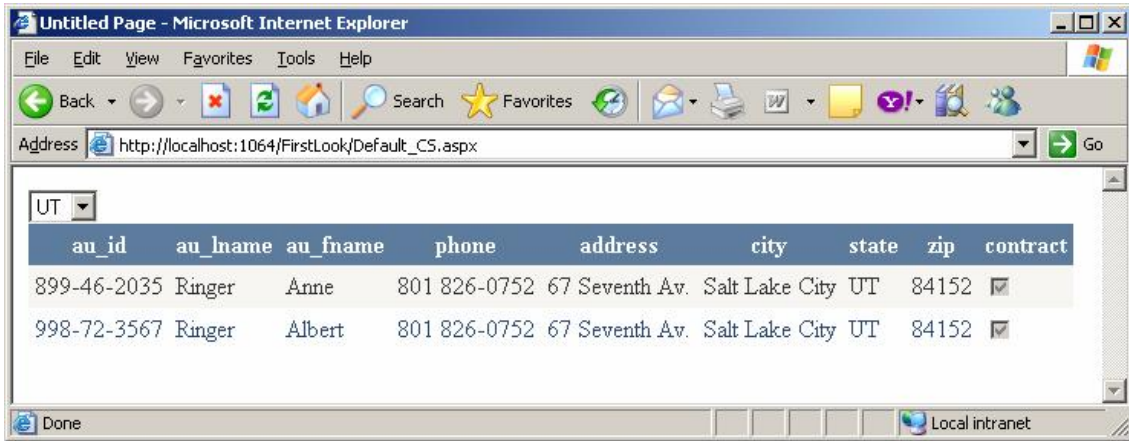
الشكل ٢٧ كما نرى هذه هي نتيجة الأختبار الحمد لله تمت بنجاح



الشكل ٢٨ هذا هو شكل الصفحة بعد تعديل الشكل Auto Format

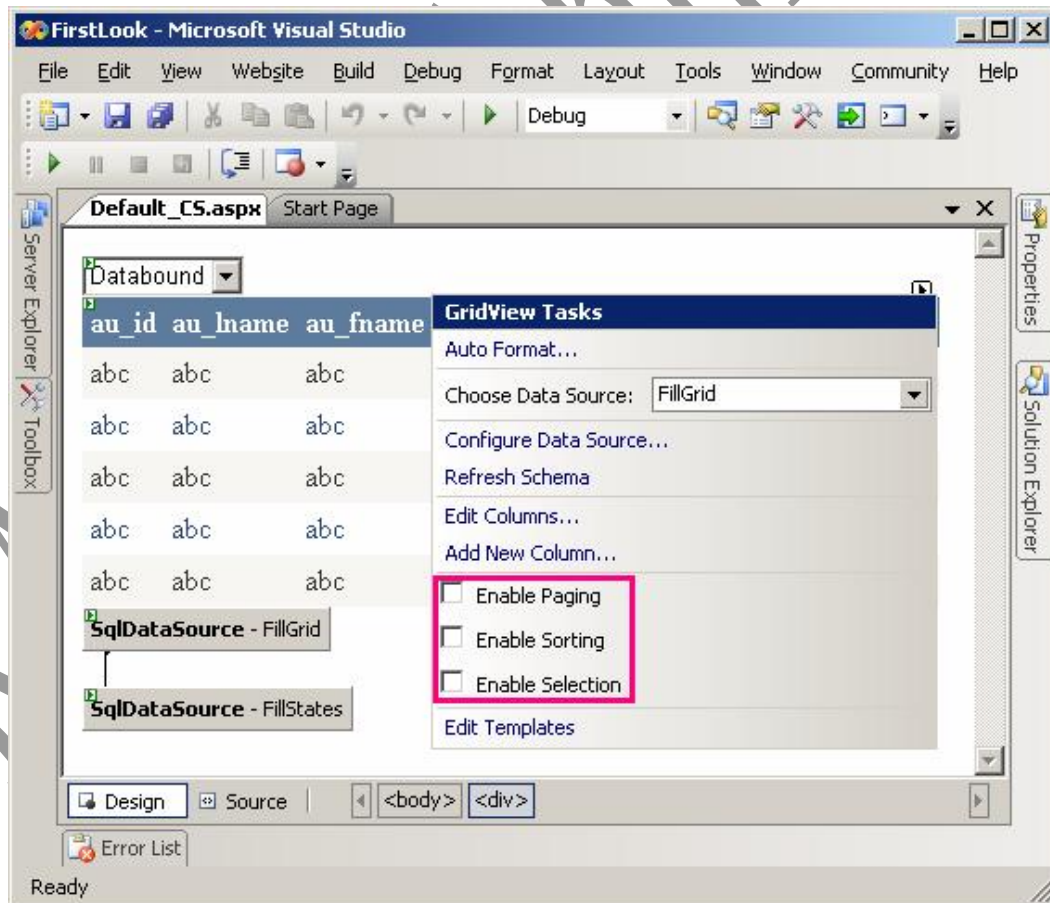


الشكل ٢٩ ولا تنسى تفعيل الخاصية AutoPostBack حيث أن هذا الحدث هو الذي سوف يقوم بتفعيل المثال كله

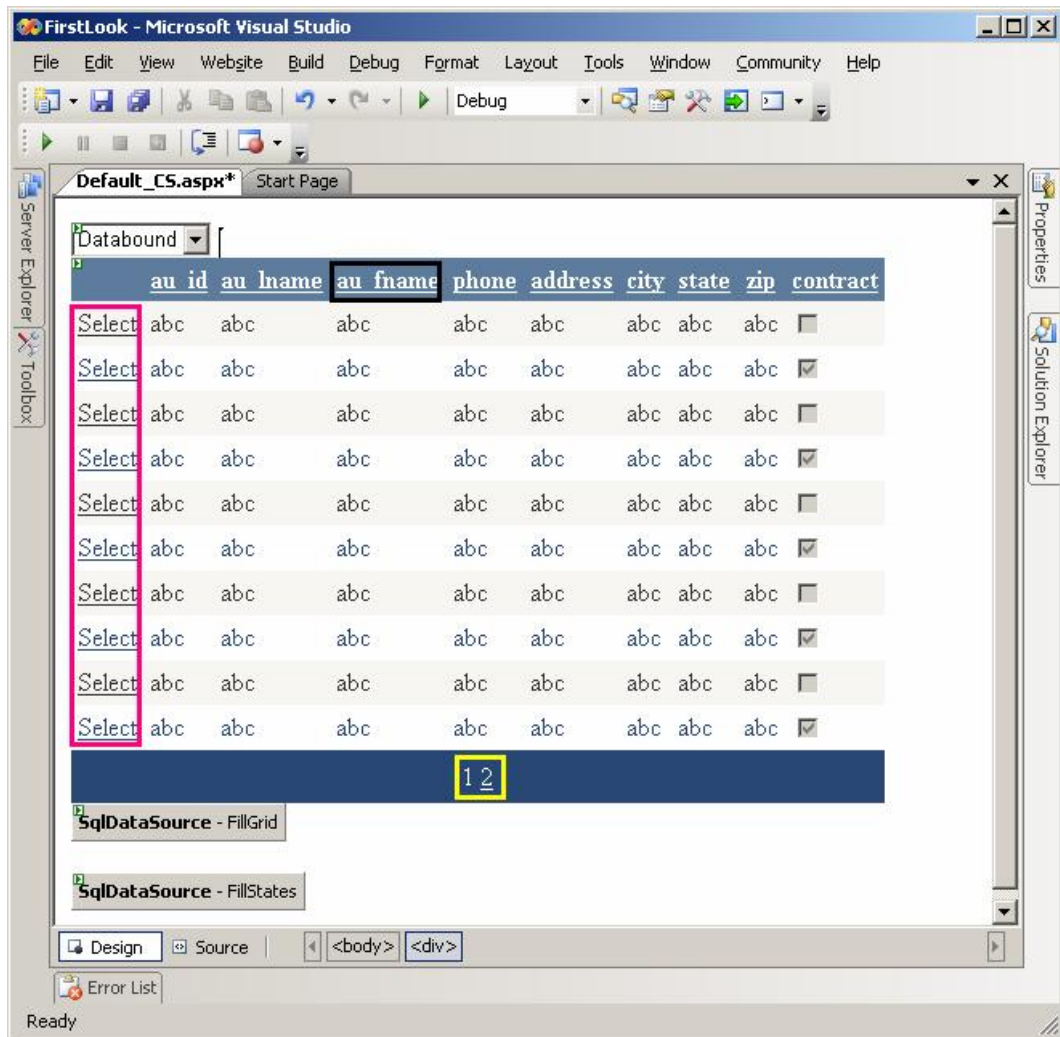


الشكل ٣٠ هذه كما نرى هي النتيجة عند تغيير القيمة في DropDownList يقوم بعرض البيانات التابعة لها

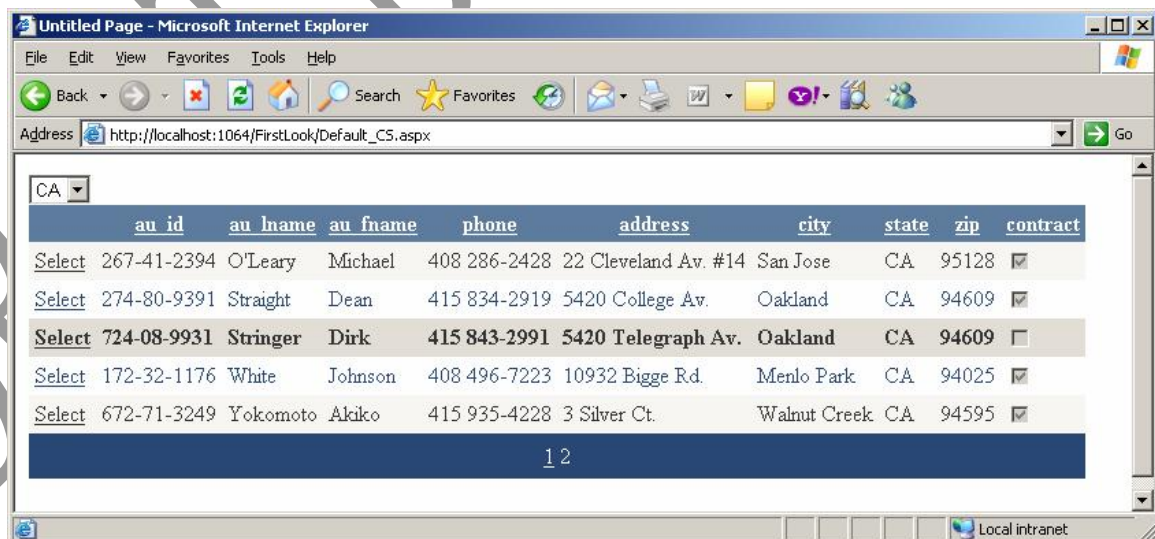
ماذا لو أردنا أن نقسم البيانات لأن كميتها كبيرة جدا. وماذا أيضا لو أردنا أن نعطي امكانيه للمستخدم في الترتيب و اختيار الصف الذي يريده، لا داعي للقلق نهائيا لأن كل هذا موجود و بدون سطر كود واحد في الأصدارات السابقة من فيجوال ستوديو كنا نضطر الى كتابة كود يقوم بهذه الخواص الآن مع الأصدار ٢٠٠٥ لن تكتب أى حرف كود واحد كله سوف يتم تشغيله ببساطة، و الآن لنكمل المثال.



الشكل 31 كما نرى أن هناك ٣ إمكانيات يمكن تفعيلها وهما التقسيم الى صفحات ، إعطاء المستخدم إمكانية ترتيب العناصر في الجدول ، إعطاء المستخدم إمكانية اختيار عنصر



الشكل 32 المربع الأسود يشير الى وصلات الترتيب و التي عند الضغط عليها سوف ترتب العناصر ضيقا لهذا العمود، و المربع الأحمر يشير الى وصلة اختيار العنصر ، و المربع الأصفر يشير الى الصفحات التي تحتوى على بعض البيانات بعد التقسيم

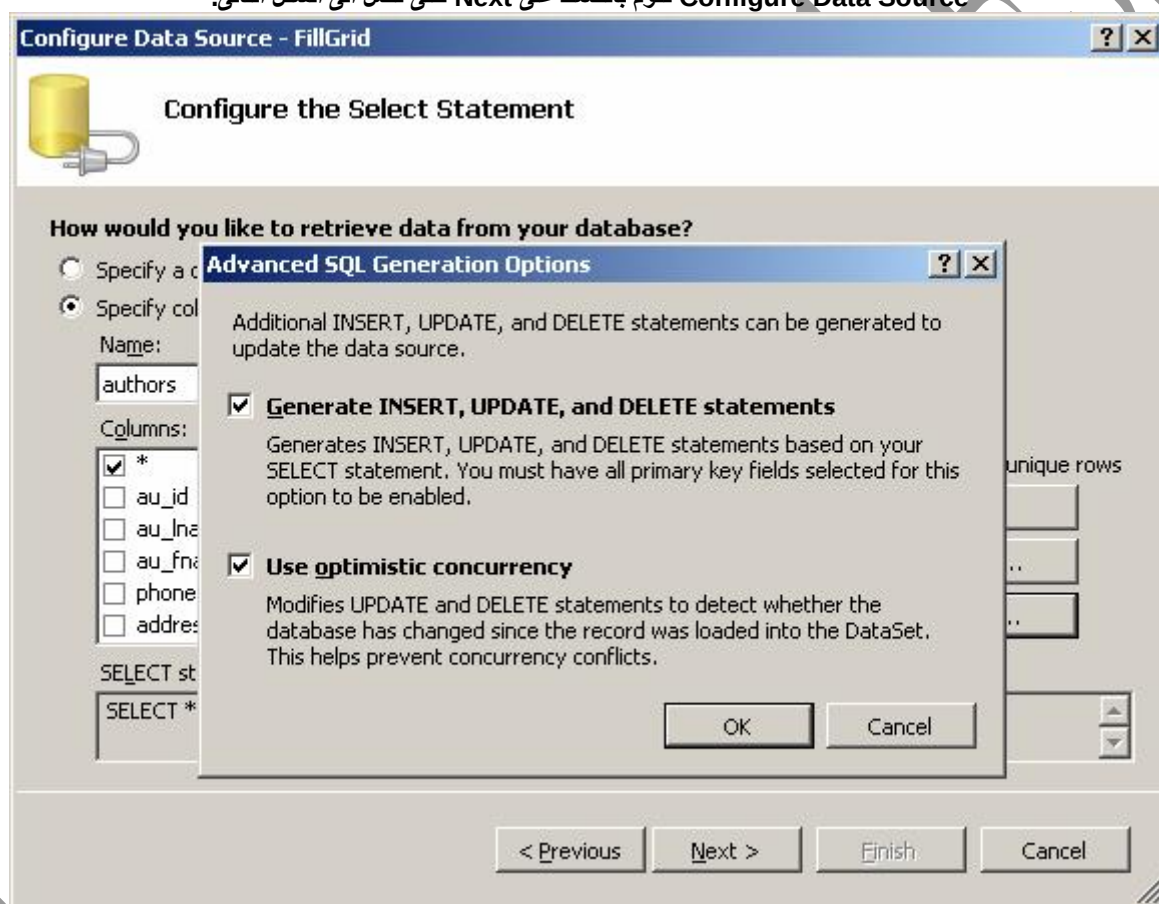


الشكل 33 هذا هو الشكل النهائي للمثال بعد ضبط الخصائص و تفعيلها

هناك شئ آخر ماذا لو أردت أن أقوم بحذف أو تعديل البيانات كيف يمكنني ذلك، بكل بساطة سوف نرى في تكملة المثال و الذي سوف نقوم فيه بأضافة خاصيتي التعديل و الأضافة كما كنا نفعل في الإصدارات القديمة و لكن بدون سطر كود. في الإصدارات السابقة كان هناك الكثير من الكود الذى يقوم بحل هذه المشكلة و هذا الكود كان يزيد كلما زاد عدد الأعمدة في حالة التعديل، و لأن نرى بقية المثال



الشكل ٣٤ نقوم بإظهار Smart Tag الخاص بالـ Datasource التابع لـ GridView و المسمى بـ FillGrid و بعد ذلك نضغط على **Configure Data Source** نقوم بالضغط على **Next** حتى نصل الى الشكل التالى.



الشكل 35 و الان نضغط على **Advanced** و سوف تظهر هذه النافذة لك و به اختياران علشك أن تقوم بتفعيلهم و هما كالاتى

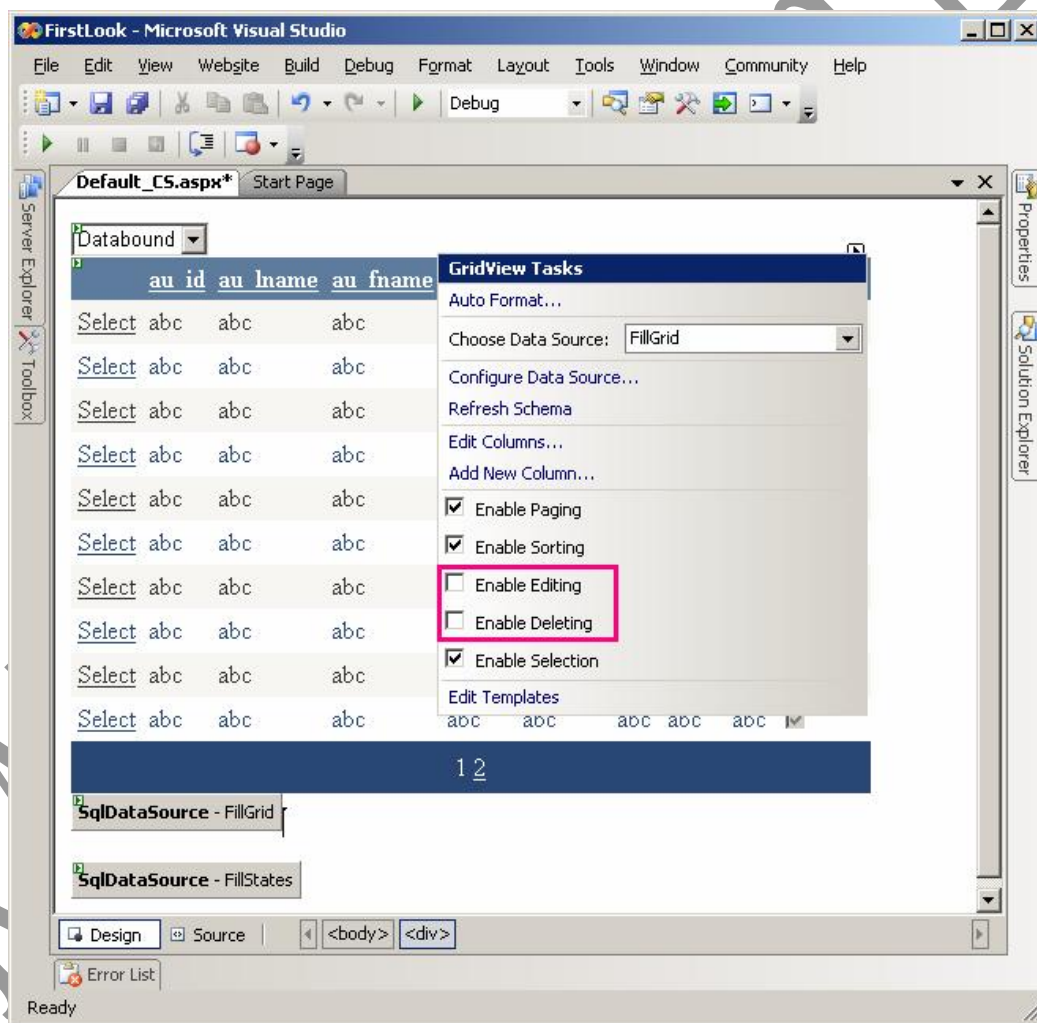
الأختيار الاول Generate Insert, Update and Delete statements

و هذا الأختيار يقوم عنك بأضافة خاصية الأضافة و التعديل و الحذف و هذا فى السابق كما ذكرت كان يتطلب عمل كثير و لكن الآن كل هذا سوف يتم عنك.

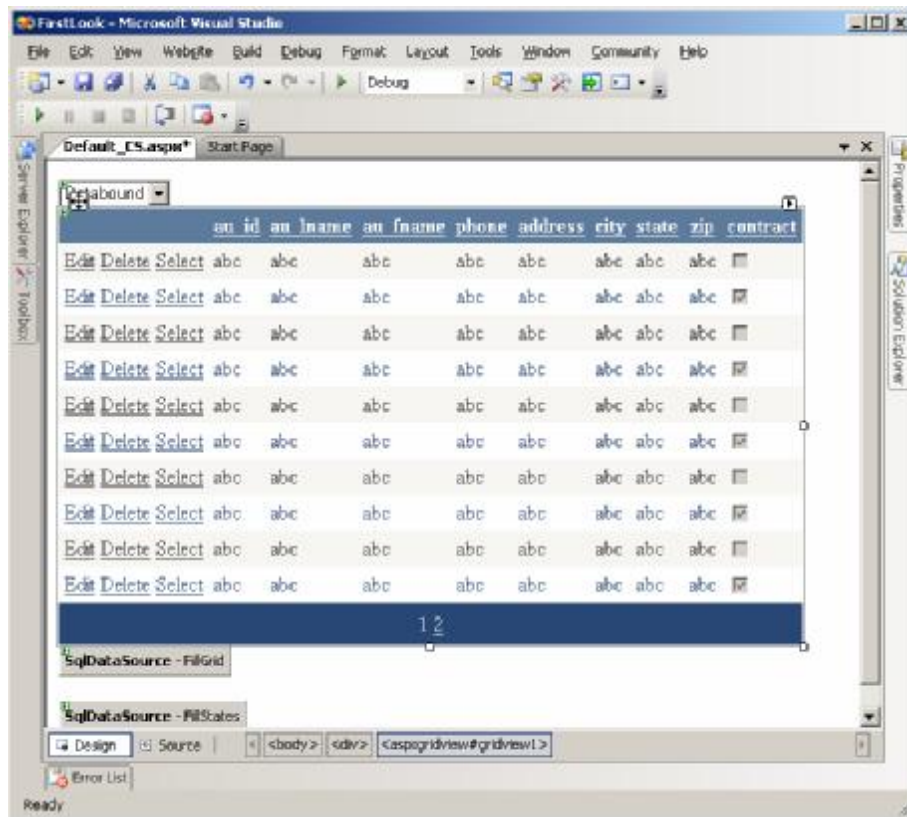
الأختيار الثانى و هو Use optimistic Concurrency

كان هناك شئ لابد من عمله و هو تأمين التعامل و تعديل البيانات لان هناك العديد من المستخدمين لقاعدة البيانات و الآن لنعرض مثال واقعى و سهل الفهم و لنفرض أن هناك مستخدمين يقوموا بنفس الوقت بتعديل بيانات معينة و

تصادف أن الأثنين يعدلان نفس البيانات و قام المستخدم الأول بحفظ البيانات ثم قام المستخدم الثاني بحفظ البيانات التي عدلها إذا فإن آخر تعديل هو التعديل الذي قام به المستخدم الثاني، حاول المستخدم الأول استخدام البيانات فوجد ان البيانات ليس كما حفظها، هنا تكمن المشكلة و هي لابد أن يكون هناك طريقة لضبط التعامل بين المستخدمين بعضهم البعض بمعنى أنه عندما يقوم مستخدم بتعديل البيانات و حاول الحفظ أن يكون هناك خطوة قبل الحفظ و هي التأكد من ان البيانات المخزنة لم تتغير و هذا ما يسمى بالـ **Optimistic Concurrency** في الإصدارات السابقة كان المطور يقوم بعمل هذا الموضوع و كان هذا بمساعدة الـ **Dataset** حيث كان هناك ما يسمى بالـ **Data version** و هي تحتوى على البيانات بكل إصداراتها البيانات الأصلية (البيانات عند عرضها قبل التعديل) و البيانات الحالية (فى قاعد البيانات) و كان يقوم المطور بمقارنة البيانات للتأكد من أن البيانات لم تتغير فيقوم بالحفظ أما فى حالة أن البيانات تم تغييرها كان يقوم المطور بعرض رسالة تحذيرية للمستخدم، الآن تقوم هذه الاداة الرائعة بهذا العمل عنك و بالتالى توفر الكثير من الوقت على المطور.



الشكل ٣٦ و الآن بعد أن قمت بتفعيل الخواص فى الـ **Datasource** قم بإظهار الـ **Smart Tag** الخاص بالـ **GridView** و سوف تلاحظ ظهور خاضيتين قم بتفعيلهم و هما كما ترى **Enable Editing & Enable Deleting**



الشكل 37 و كما نرى ظهور كلا من الوصلتين Delete & Edit و التي سوف تقوم كلا منهما بمهمة التعديل و الحذف

```
<asp:SqlDataSource ID="FillGrid" runat="server" ConnectionString="<%"$
ConnectionString:pubsConnectionString %>"
SelectCommand="SELECT * FROM [authors] WHERE ([state] = @state)"
ConflictDetection="CompareAllValues"
DeleteCommand="DELETE FROM [authors] WHERE [au_id] = @original_au_id AND
    [au_lname] = @original_au_lname AND [au_fname] = @original_au_fname AND
    [phone] = @original_phone AND [address] = @original_address AND
    [city] = @original_city AND [state] = @original_state AND
    [zip] = @original_zip AND [contract] = @original_contract"
InsertCommand="INSERT INTO [authors] ([au_id], [au_lname], [au_fname], [phone],
    [address], [city], [state], [zip], [contract])
    VALUES (@au_id, @au_lname, @au_fname, @phone, @address,
    @city, @state, @zip, @contract)"
OldValuesParameterFormatString="original_{0}"
UpdateCommand="UPDATE [authors] SET [au_lname] = @au_lname, [au_fname] = @au_fname,
    [phone] = @phone, [address] = @address, [city] = @city, [state] = @state,
    [zip] = @zip, [contract] = @contract WHERE [au_id] = @original_au_id AND
    [au_lname] = @original_au_lname AND [au_fname] = @original_au_fname AND
    [phone] = @original_phone AND [address] = @original_address AND
    [city] = @original_city AND [state] = @original_state AND
    [zip] = @original_zip AND [contract] = @original_contract">
<SelectParameters>
```

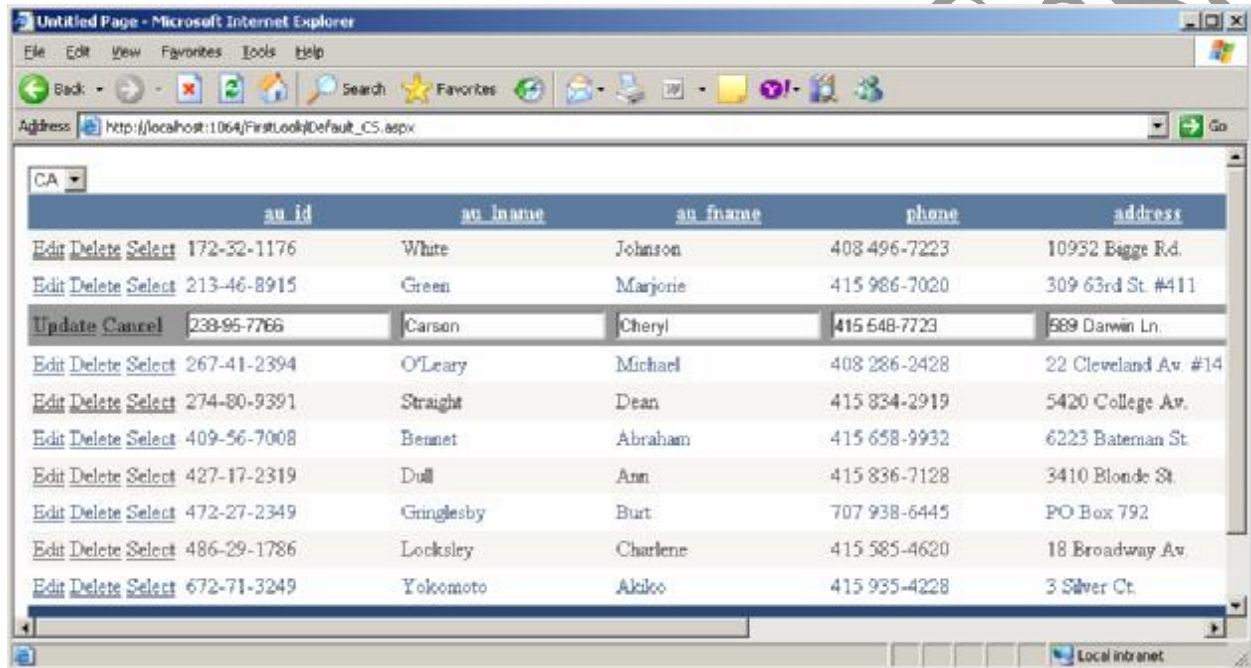
```

        <asp:ControlParameter ControlID="DropDownList1" Name="state"
PropertyName="SelectedValue"
        Type="String" />
    </SelectParameters>
    <DeleteParameters>
        <asp:Parameter Name="original_au_id" Type="String" />
        <asp:Parameter Name="original_au_lname" Type="String" />
        <asp:Parameter Name="original_au_fname" Type="String" />
        <asp:Parameter Name="original_phone" Type="String" />
        <asp:Parameter Name="original_address" Type="String" />
        <asp:Parameter Name="original_city" Type="String" />
        <asp:Parameter Name="original_state" Type="String" />
        <asp:Parameter Name="original_zip" Type="String" />
        <asp:Parameter Name="original_contract" Type="Boolean" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="au_lname" Type="String" />
        <asp:Parameter Name="au_fname" Type="String" />
        <asp:Parameter Name="phone" Type="String" />
        <asp:Parameter Name="address" Type="String" />
        <asp:Parameter Name="city" Type="String" />
        <asp:Parameter Name="state" Type="String" />
        <asp:Parameter Name="zip" Type="String" />
        <asp:Parameter Name="contract" Type="Boolean" />
        <asp:Parameter Name="original_au_id" Type="String" />
        <asp:Parameter Name="original_au_lname" Type="String" />
        <asp:Parameter Name="original_au_fname" Type="String" />
        <asp:Parameter Name="original_phone" Type="String" />
        <asp:Parameter Name="original_address" Type="String" />
        <asp:Parameter Name="original_city" Type="String" />
        <asp:Parameter Name="original_state" Type="String" />
        <asp:Parameter Name="original_zip" Type="String" />
        <asp:Parameter Name="original_contract" Type="Boolean" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="au_id" Type="String" />
        <asp:Parameter Name="au_lname" Type="String" />
        <asp:Parameter Name="au_fname" Type="String" />
        <asp:Parameter Name="phone" Type="String" />
        <asp:Parameter Name="address" Type="String" />

```

```
<asp:Parameter Name="city" Type="String" />
<asp:Parameter Name="state" Type="String" />
<asp:Parameter Name="zip" Type="String" />
<asp:Parameter Name="contract" Type="Boolean" />
</InsertParameters>
</asp:SqlDataSource>
```

هذا هو كود في الـ **Source View** و هذا الكود كما نرى به كل تفاصيل عملية التعديل و الحذف من أول جملة الاستعلام SQL الى مدخلات التي سيتم حفظها.

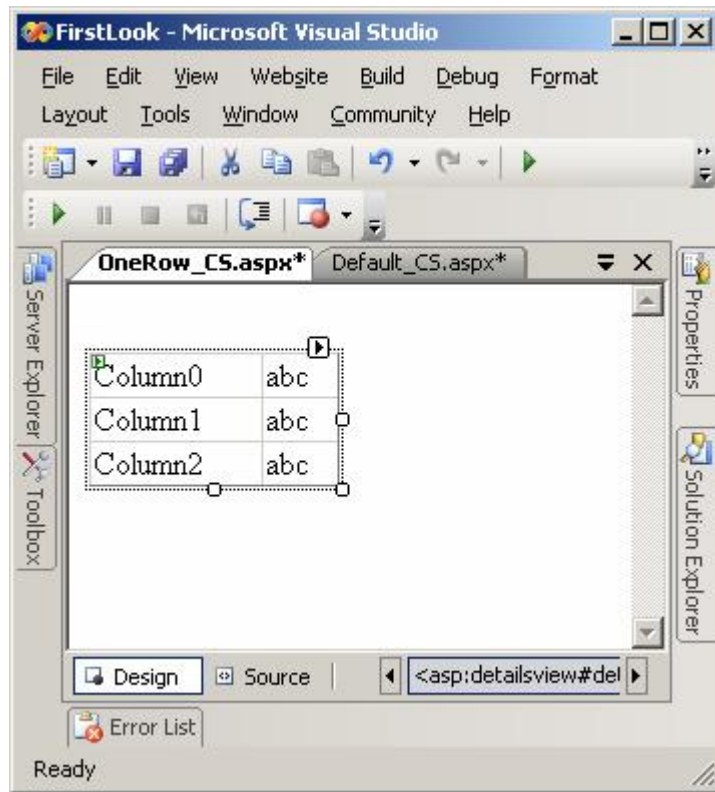


الشكل 38 هذا هو الشكل النهائي في إحدى المحاولات للتعديل.

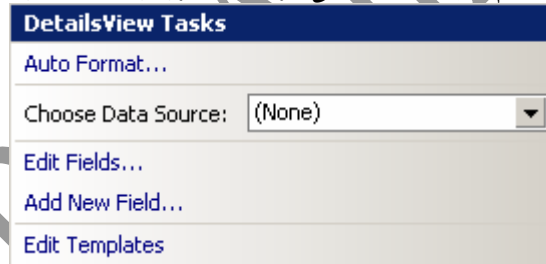
على ما أظن أن بعضنا كان في بعض الأحيان لا يتعامل مع الـ **Data Grid** و إنما كان يعرض البيانات بشكل مختلف عن ذلك وهو ان الحاجة كانت تتطلب عرض صف واحد و كان هذا في بعض الأحيان يعتبر معناه لسبب بسيط نفرض انك تود عرض ٧٠ بيان من قاعدة البيانات كان هناك كثير من الكود لعمل هذا الموضوع و لكن الآن مع الأداة **Details View** تم حل هذه المشكلة و الان لنرى كيف نتعامل مع هذه الأداة و لنجرب مثال يساعدنا على فهم الفكرة بشكل أكبر.



الشكل ٣٩ سوف نستعمل الأداة **Details View** من صندوق الأدوات و هي أداة كما سنرى مخصصة للعرض الفردي أو كما يسمونها للعرض التفصيلي ،



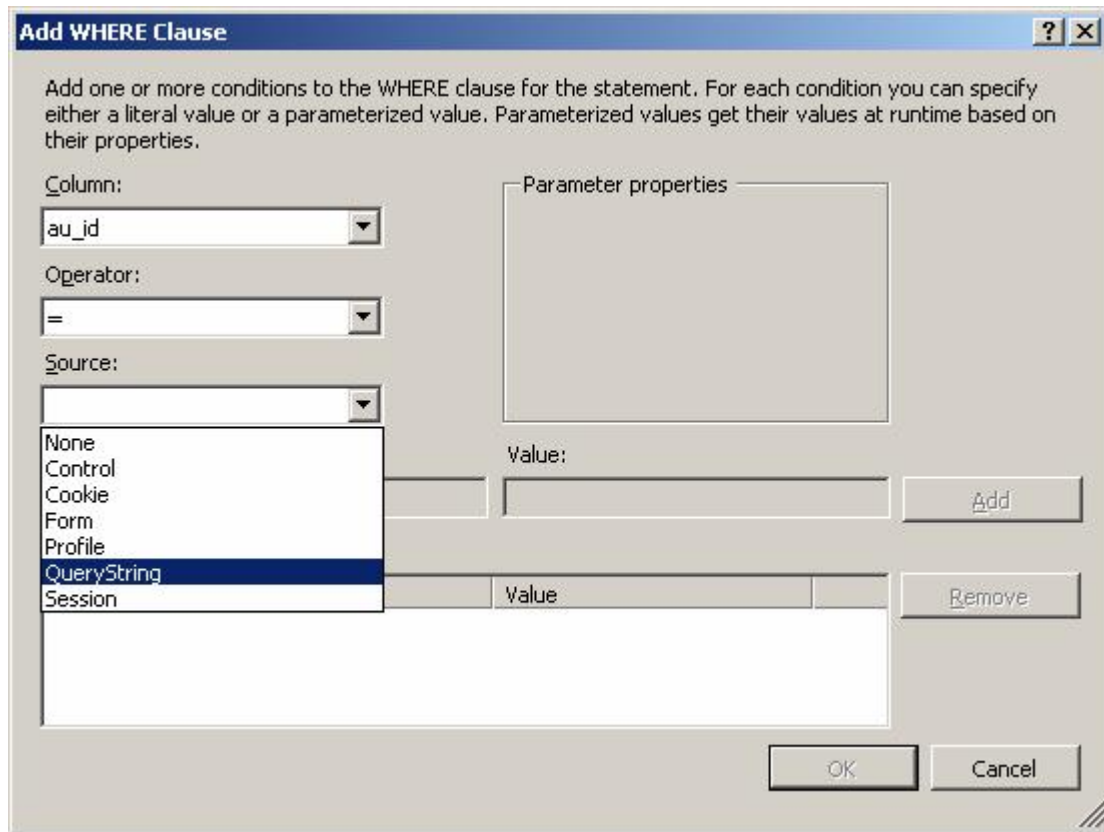
الشكل 40 هذا هو شكل الأداة و الآن لنقم بضبط الإعدادات التي منها سوف تظهر قوة هذه الأداة و الآن لنضغط على الـ Smart Tag



الشكل 41 سوف نختار Choose Data Source



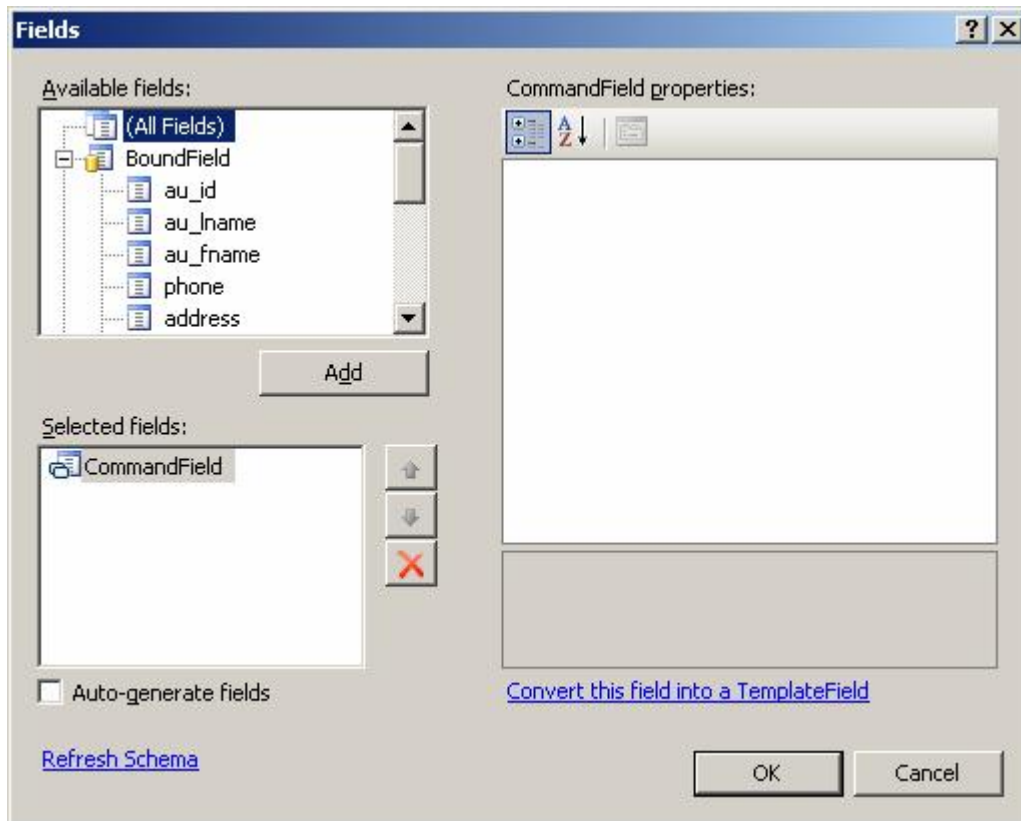
الشكل 42 و الآن نختار New Data source



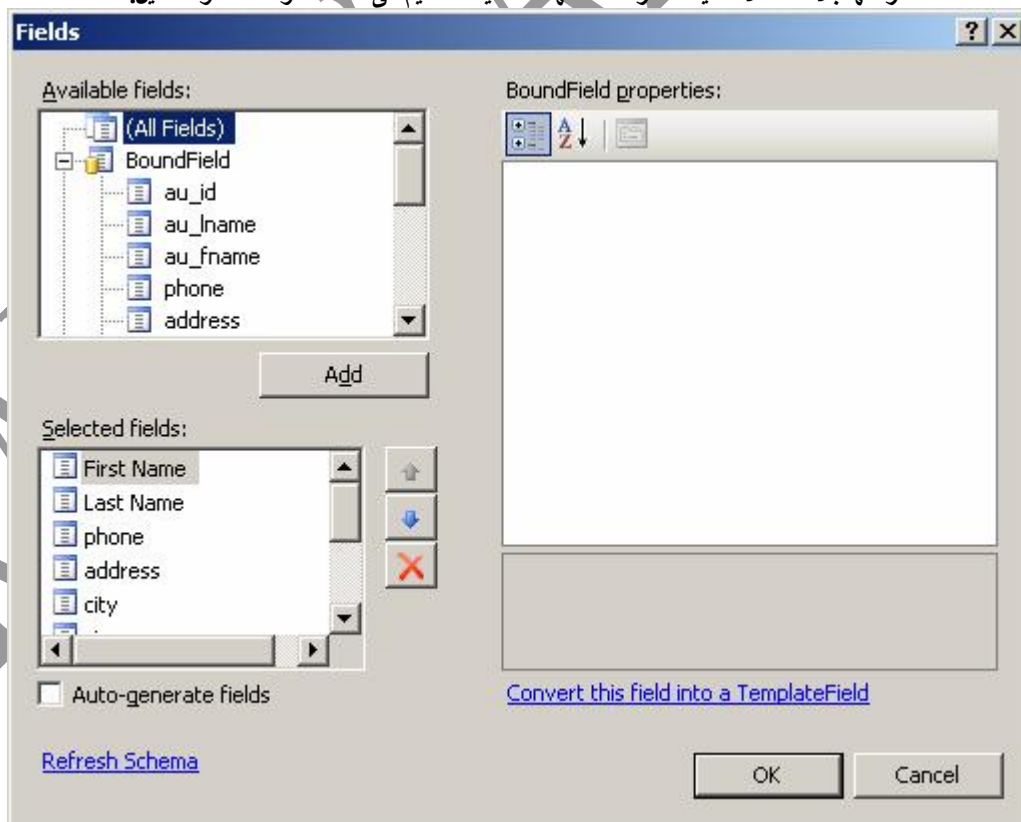
الشكل 43 لن أذكر طريقة أعداد Datasource لأننا قمنا بها أكثر من مرة، و الآن نختار جدول Authors ثم نضغط على زر Advanced سوف تظهر هذه النافذة و التي منه شوف نختار مصدر فلترة البيانات بمعنى آخر هنا سوف نختار مصدر الشرط Where في جملة الاستعلام و كما نرى أنه من الممكن أن يكون مصدر الشرط أحد هذه الأنواع الموجودة في القائمة المنزلة Source ، بعد ذلك قم بتحديد أسم Query String التي سوف تحمل قيمة الشرط،

الشكل 44 كما هو موضح هنا أننا قمنا بتحديد حقل مصدر البيانات بعد ذلك نقيم بالضغط على Add ثم بعد ذلك OK ونكمل بقية خطوات معالجة مصدر البيانات Data source wizard

الشكل ٤٥ مرة أخرى نظهر الـ Smart Tag الخاص بالـ Detail view ونقوم بتفعيل كل الخيارات كما نرى ولكن لا بد وأن تكون حزين لأن الخاصيتين الخاصتين بالتعديل والحذف لا بد وأن تكون مفعلة في الـ Data Source كما وضعنا من قبل.



الشكل 46 من الـ Smart Tag الخاص بالـ Detail view نقوم باختيار الـ Edit Fields و هنا سوف نختار البيانات المراد عرضها و التي لن يتم عرضها بالإضافة الأماكن المراد إضافتها كأمكانيه التقسيم الى صفحات و الحذف و التعديل.



الشكل 47 بعد اختيار البيانات المراد عرضها و ذلك عن طريق اختيار الحقل مراد عرضه ثم Add بعد الانتهاء من هذه العملية كل ما عليك هو ان تضغط على OK

First Name	abc
Last Name	abc
phone	abc
address	abc
city	abc
zip	abc
contract	☐
Edit	
Delete	
New	
1 2	

SqlDataSource - OneRow

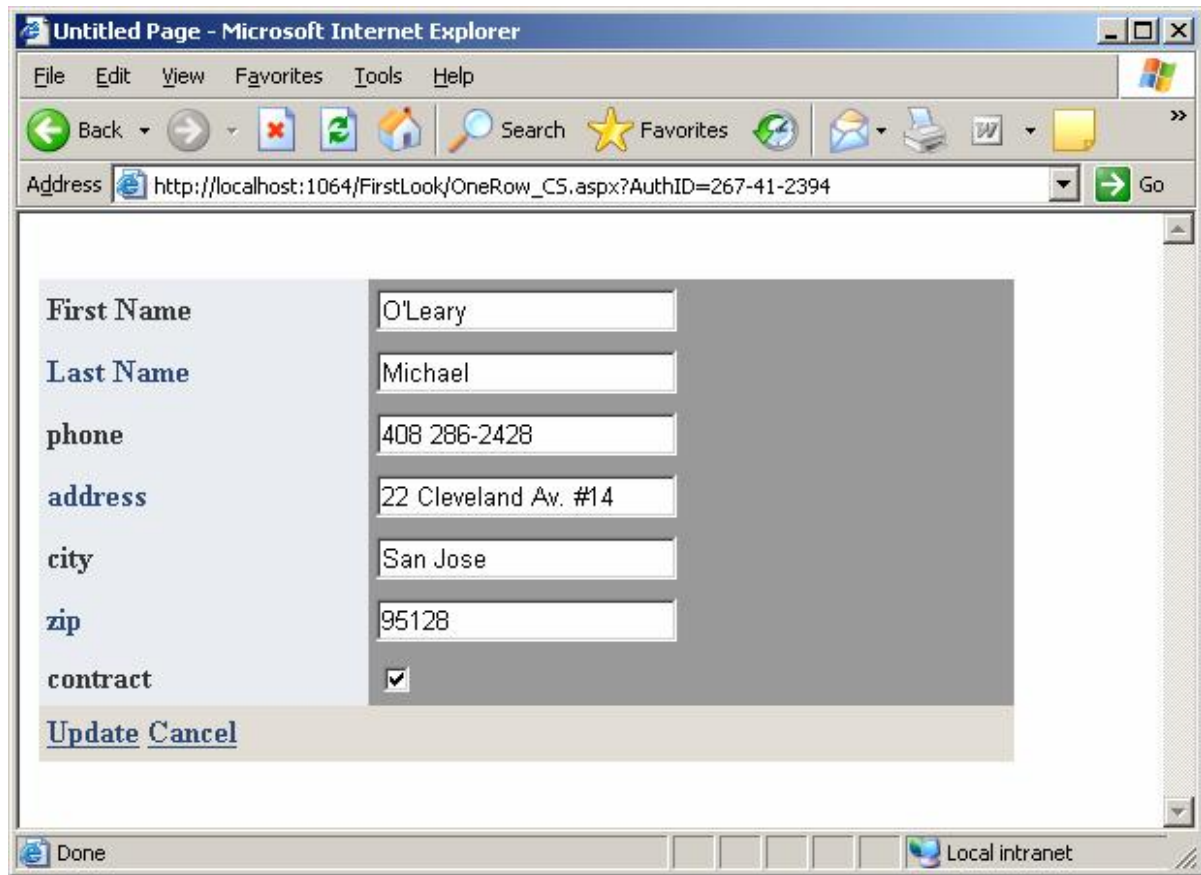
الشكل 48 هذا هو شكل الأداة بعد الانتهاء من إعدادها و طبعا بعد تعديل الـ **Auto Format** الخاص بالأداة

Microsoft Internet Explorer window showing the web application. The address bar displays the URL: `http://localhost:1064/FirstLook/OneRow_CS.aspx?AuthID=267-41-2394`. The page content includes a form with the following fields and values:

First Name	O'Leary
Last Name	Michael
phone	408 286-2428
address	22 Cleveland Av. #14
city	San Jose
zip	95128
contract	☒

Below the form, there are three links: [Edit](#), [Delete](#), and [New](#).

الشكل 49 و هذا هو الشكل النهائي أثناء عمل الأداة و كما نلاحظ في الـ **Address bar** شريط العنوان أن عنوان الصفحة يحتوي على **Query string** و هي كما قمنا بأعدادها **AuthID** و كما نرى المثال يعمل بشكل رنع



Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print W Go

Address http://localhost:1064/FirstLook/OneRow_CS.aspx?AuthID=267-41-2394

First Name O'Leary

Last Name Michael

phone 408 286-2428

address 22 Cleveland Av. #14

city San Jose

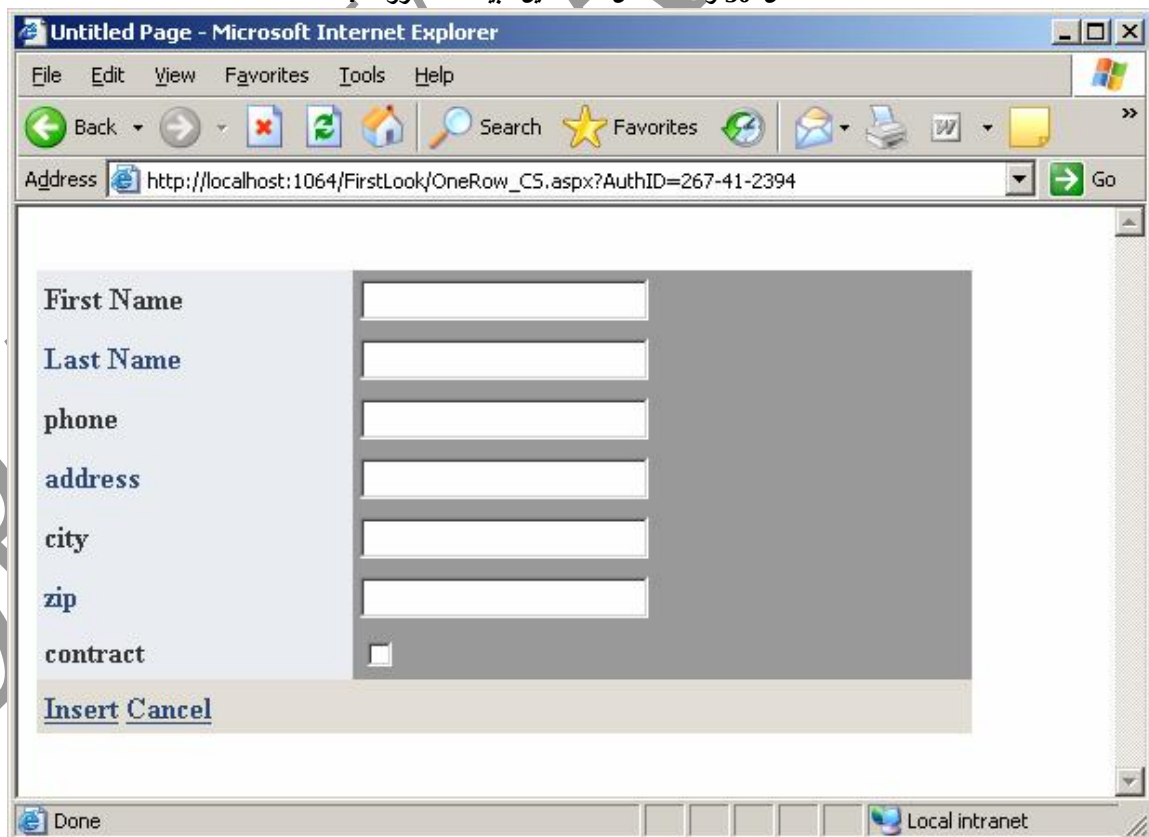
zip 95128

contract ☒

[Update](#) [Cancel](#)

Done Local intranet

الشكل 50 و هذا الشكل عند تعديل البيانات المعروضة.



Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print W Go

Address http://localhost:1064/FirstLook/OneRow_CS.aspx?AuthID=267-41-2394

First Name

Last Name

phone

address

city

zip

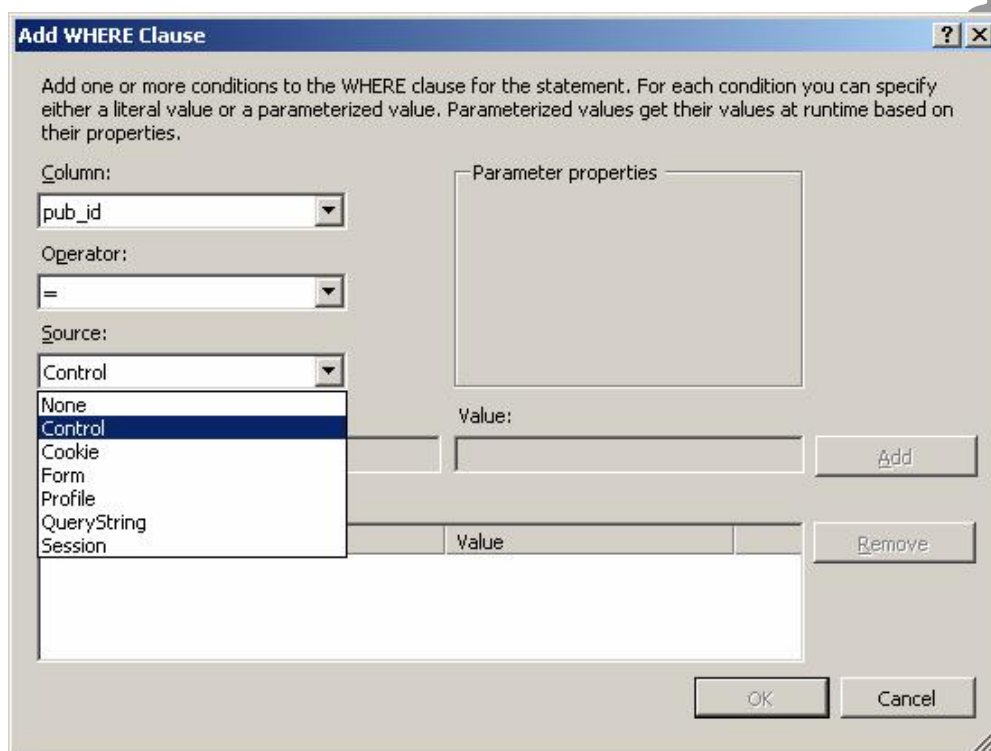
contract ☐

[Insert](#) [Cancel](#)

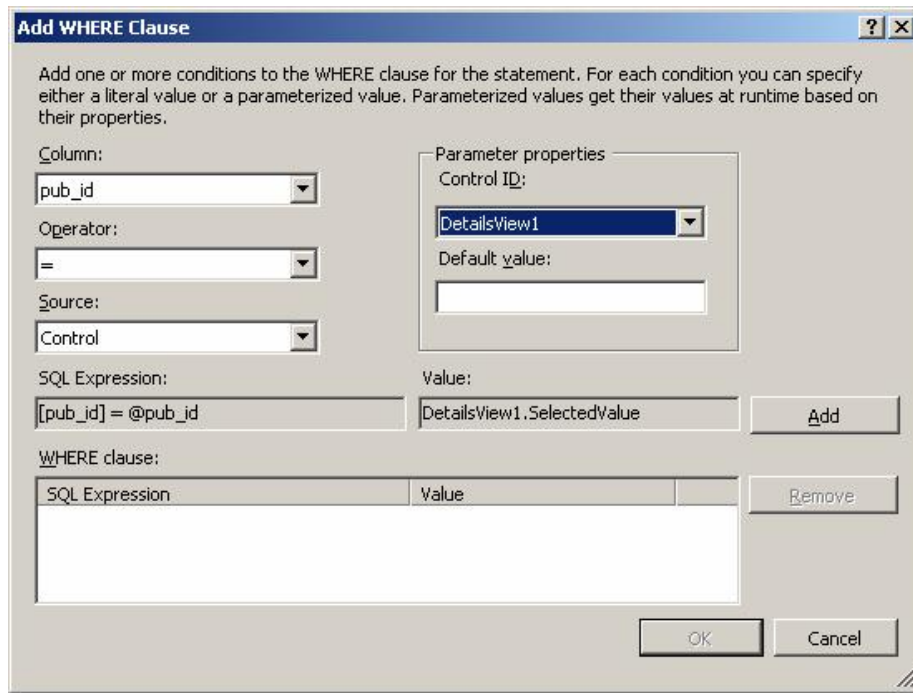
Done Local intranet

الشكل 51 و هذا هو الشكل عند طلب إضافة بيان جديد.

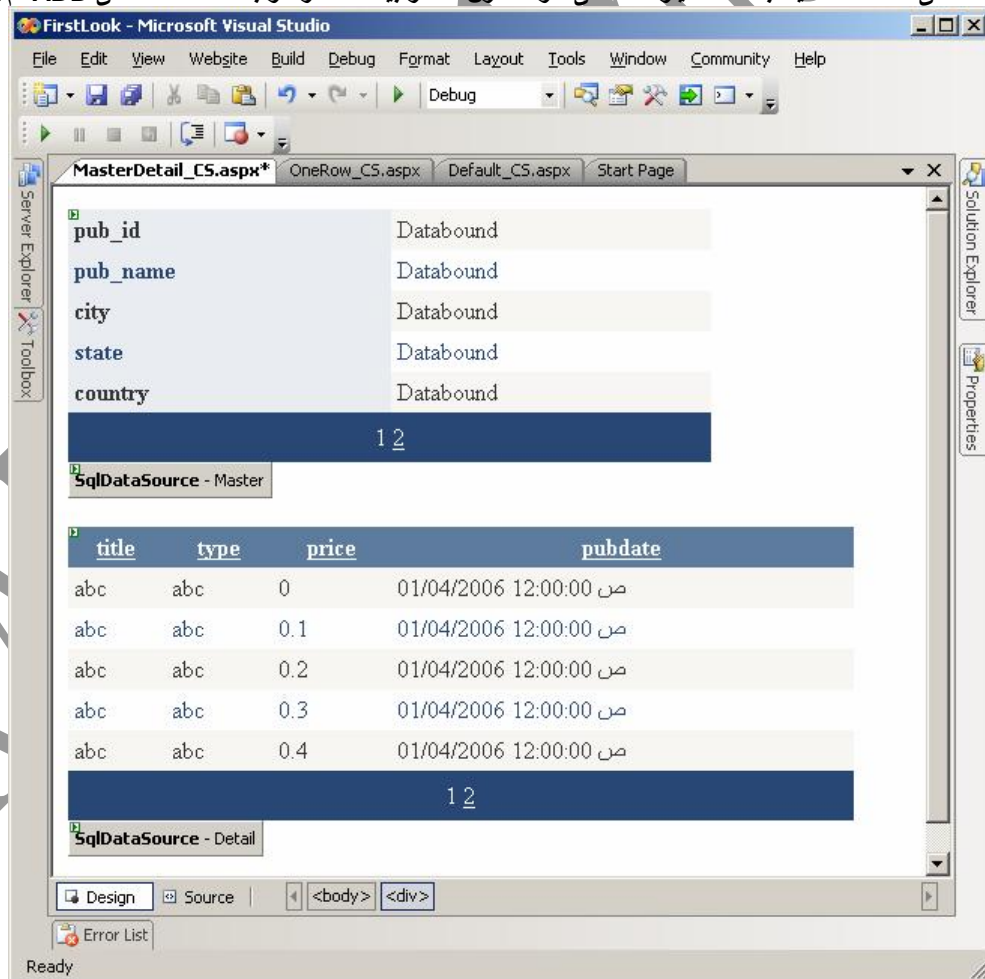
ماذا لو أردنا أن نقوم بعمل نموذج Master & Details هذا النموذج كلنا نعرفه و أبسط مثال له هو أى فاتورة حيث يوجد بيانات المشتري فى أول الفاتورة ثم يأتي بعد ذلك جدول بالأصناف، و لنوضح هذه المعلومة بشكل أكبر، المثال التالى عبارة عن بيانات الناشرين و فى النصف الآخر من الصفحة بيانات الكتب التى نشرها الناشر، سوف نحتاج الى وضع Grid view و Detail View فى Detail View سوف تعرض محتويات الجدول Publisher ثم بعد ذلك فى Grid View سوف تعرض الـ Titles و لكن فى الجزء الخاص بالـ Where أو بمعنى آخر جملة الاستعلام، نقوم بالآتى،



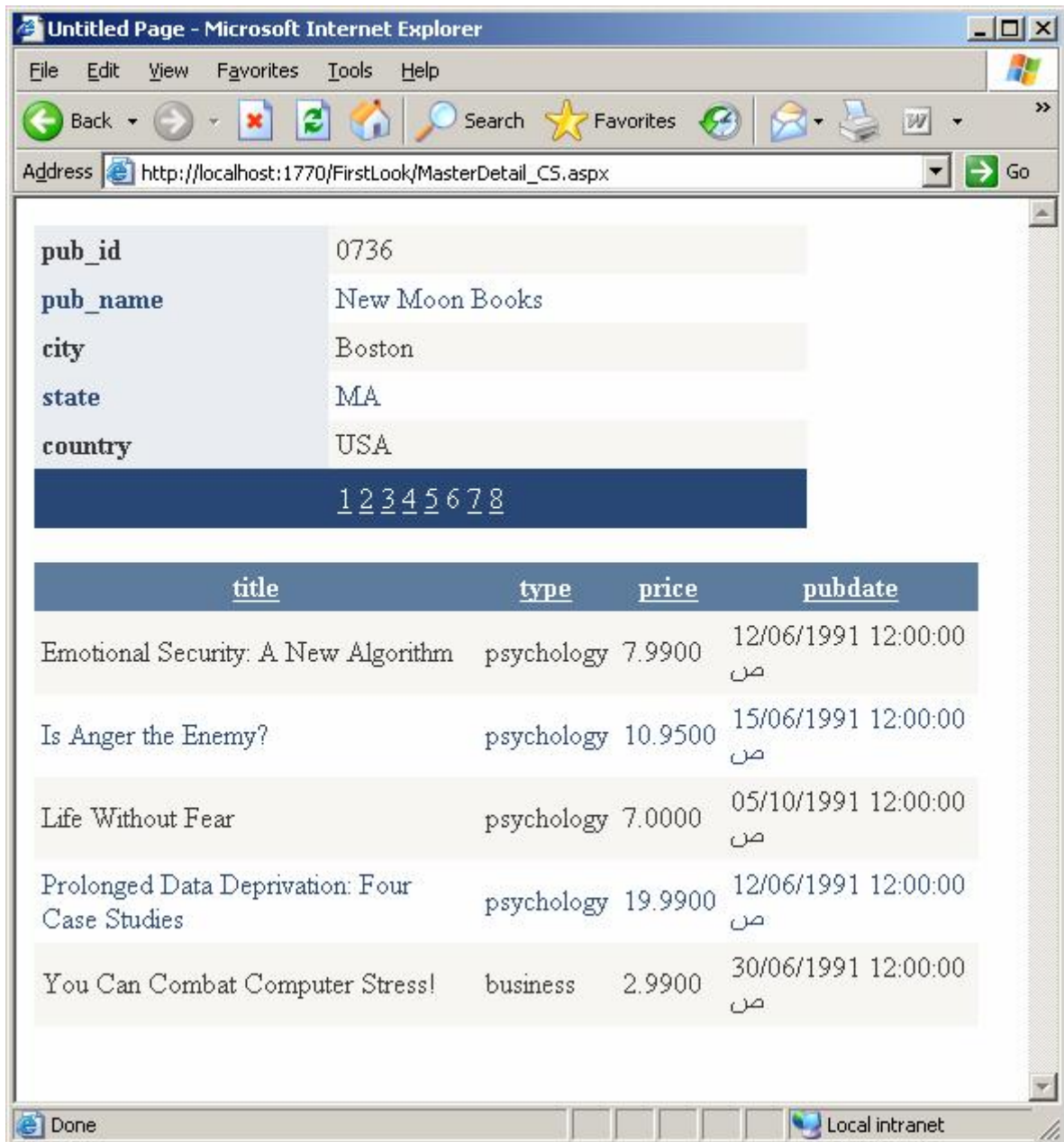
الشكل 52 أولا نختار الحقل الذي سوف يتم عليه تطبيق الشرط Column و بعد ذلك نختار نوع الشرط Operator و بعد ذلك مصدر الشرط و سوف نختار من مصدر الشرط النوع Control.



الشكل ٥٣ في هذه الشاشة يطلب منك اختيار الاداة التي سوف تكون مصدر بيانات الشرط و بعد ذلك أضغط على ADD ثم OK



الشكل ٥٤ هذا هو الشكل النهائي للمثال كما نرى في الجزء علوى Detail view و فى الجزء السفلى Grid View



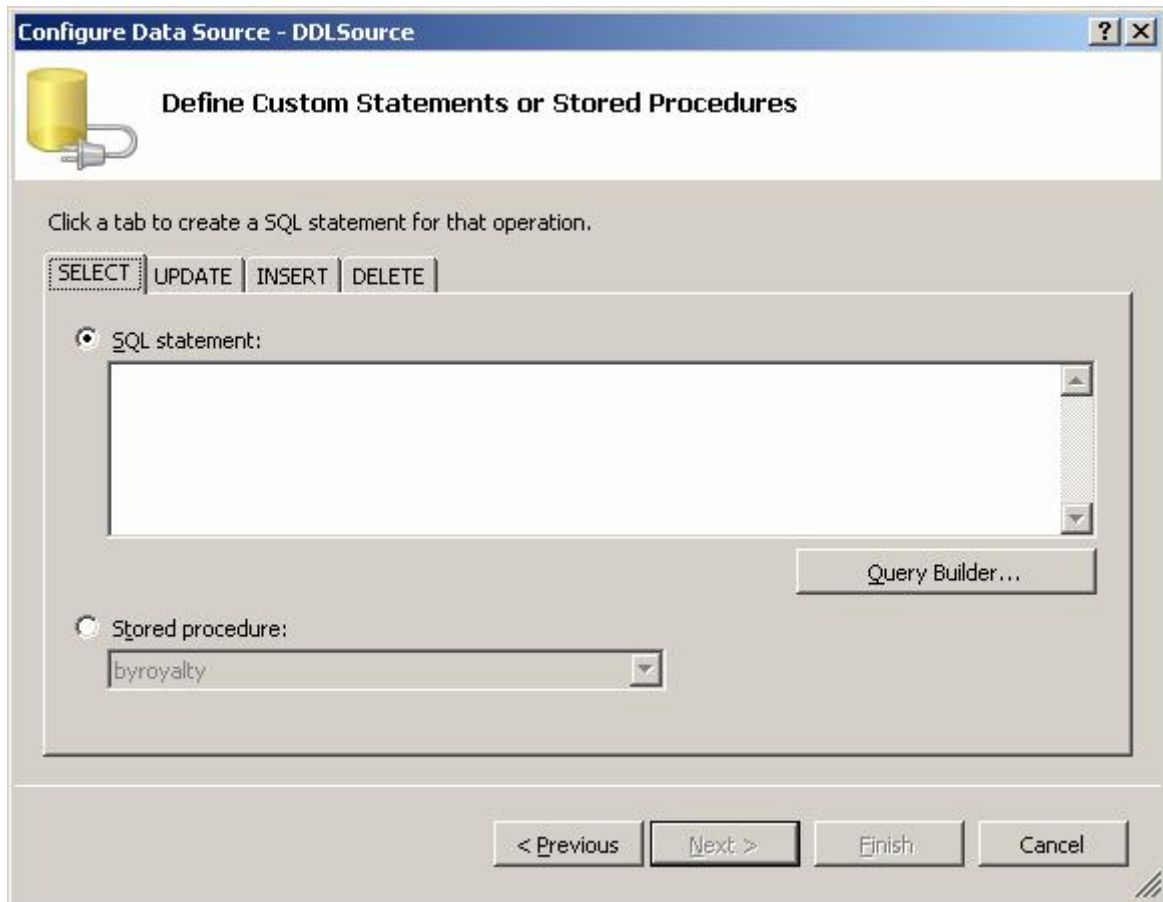
الشكل 55 و هذا هو المثال أثناء التجربة.

و الآن و قد أنتهينا من طريقة استخدام بعض أدوات التعامل مع قواعد البيانات، من المؤكد أن كثير منا قد يخطر بباله أن هناك موضوع آخر لم نتطرق اليها على سبيل المثال عرض البيانات المخزنة لتخفيف الضغط على قاعدة البيانات و هذا ما يسمى بالـ **Caching**، و الآن لنستعرض معا كيفية تفعيل هذه الخاصية و التي قد توفر الكثير من الوقت و الأداء الذي سوف يحسب لصالح المطور،

عملية الاتصال بقاعدة البيانات و أحضار البيانات منها عملية تحتاج الى وقت ، على سبيل المثال أنك لو حبيت تحضر ١٠٠ صف من البيانات سوف تستهلك بعض الوقت في إجراء اتصال التأكد من هوية المتصل بقاعدة البيانات من أجل حماية البيانات و بعد ذلك تطبيق جملة الاستعلام ثم الرجوع بالنتيجة، هذه العملية في حد ذاتها تستهلك الكثير من الوقت، هنا ظهرت الحاجة لما يسمى بتخزين البيانات **Data Caching** و عملية تخزين البيانات هذه نستعملها في بعض الأحيان

لأنه توفر سرعة كبيرة جدا بالإضافة الى أنه تقلل عدد مرات الاتصال بقاعدة البيانات، عملية تخبئة البيانات عادة لا تتم الا على البيانات التي لا تتغير باستمرار على سبيل المثال قوائم المنتجات، نادرا ما تتغير في حالة مثل هذه الحالة نقوم بتخبئة البيانات الخاصة بالقائمة و عند الحاجة اليه نقوم بعرضها بدل من الاتصال بقاعدة البيانات و أستهلاك هذا الوقت، و لكن كل هذا الكلام نظري و نحن كما تعودنا على أنه دائما هناك تطبيق عملي على ما نقوم به، و الآن لنرى مثال يوضح ما نقصده بعملية تخبئة البيانات. و الآن سوف نقوم بأعداد مثال بسيط وهو Drop Down List بها أسماء المؤلفين و عن الاختيار منها تظهر البيانات تفصيلية في Grid View و سوف نظهر مع بيانات حقل سوف نضيفه وهو حقل التاريخ و سوف نرى ما هو المقصود بشكل أوضح مع المثال. كم بوضع كلا من Drop Down List و Detail View و الآن لنقم بأعدادهم و هذه المرة سوف نجد طريقة جديدة للأعداد.

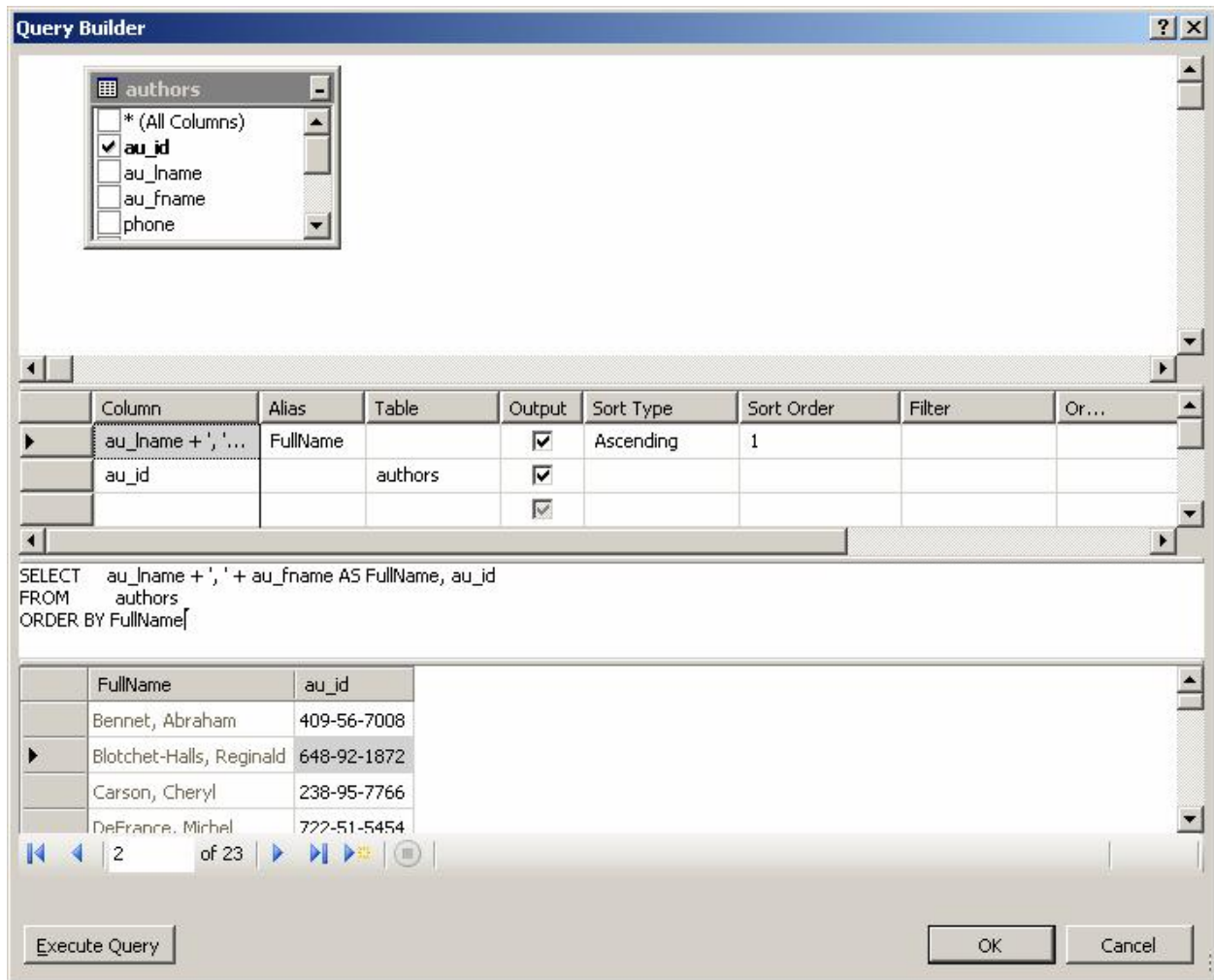
الشكل ٥٦ بعد أعداد جملة الاتصال سوف يظهر لك هذ الشكل المعتاد لأختيار مصدر البيانات هذه المرة لن نأخذ من الجدول مباشرة و لكن سوف نقوم بعمل الاستعلام بأنفسنا، فم بأختيار Specify a custom SQL statement or stored procedure



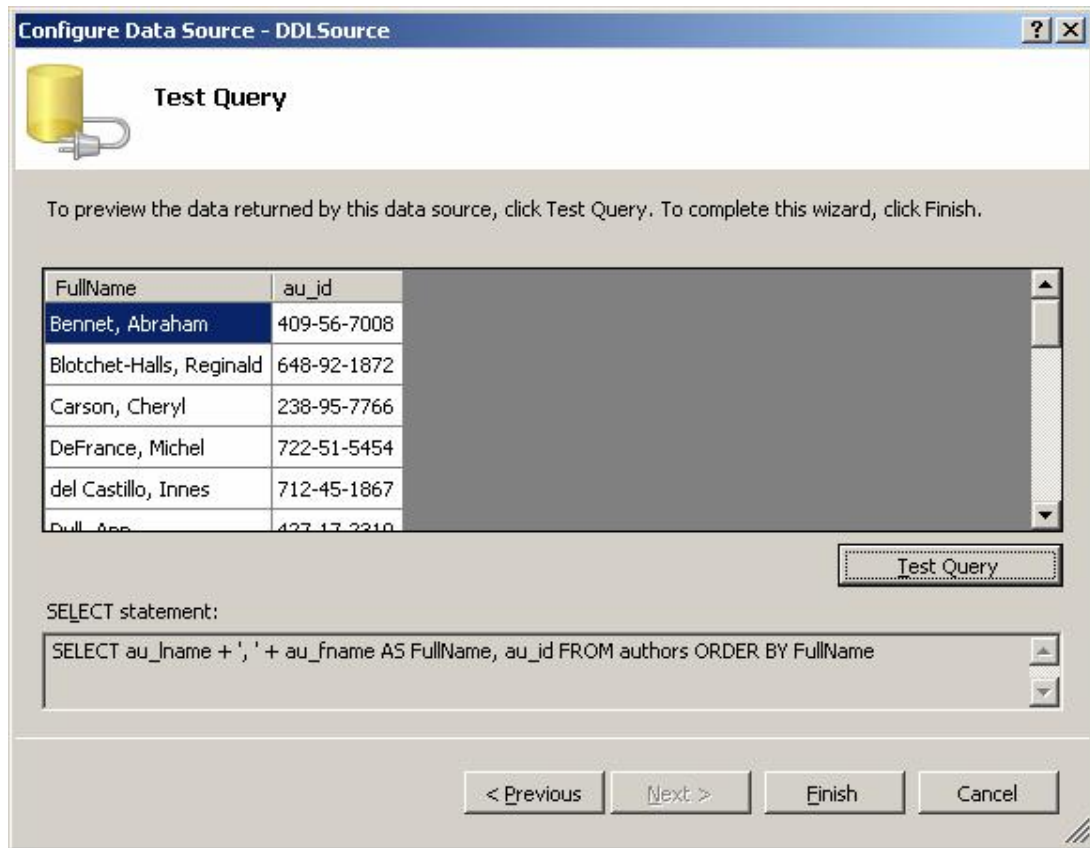
الشكل 57 سوف يظهر لك هذا الشكل قم بالضغط على Query Builder.



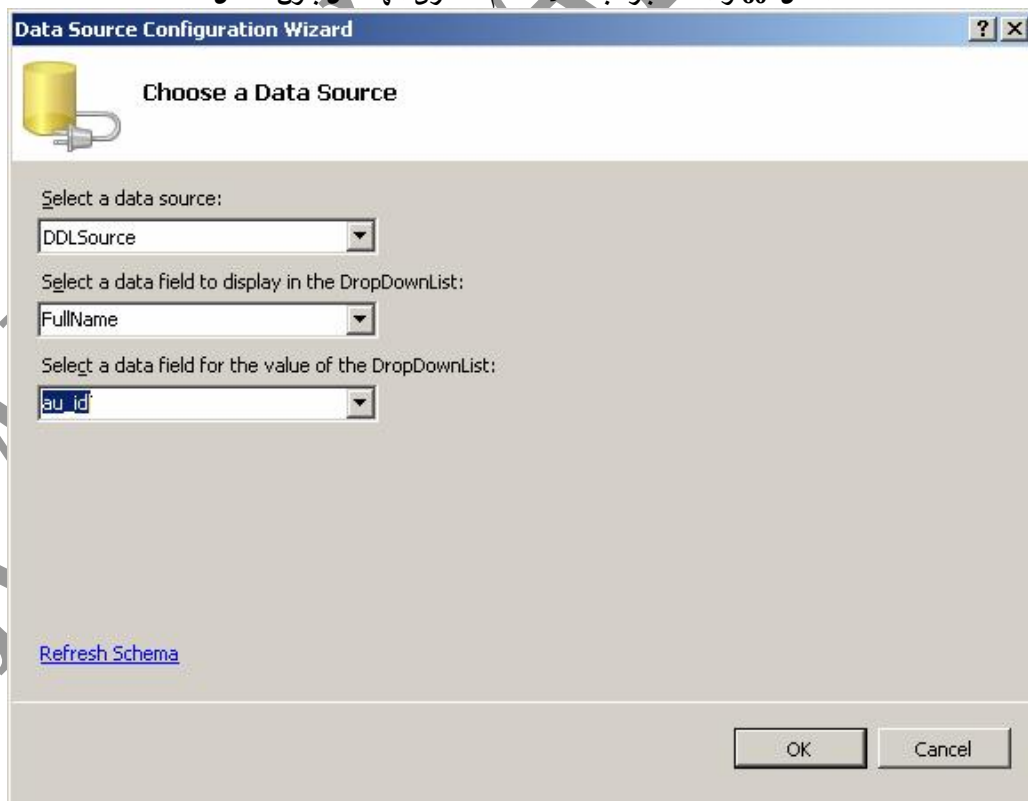
الشكل 58 من هذه النافذة قم باختيار الجداول المشاركة في جملة الاستعلام وفي حالتنا سوف نختار authors و بعد ذلك نضغط على Add ثم close



الشكل 59 سوف تظهر هذه النافذة من هذه النافذة قم باختيار البيانات التي سيتم عرضها ، كما ترى قم باختيار كلاً من au_lname و au_fname و أدمجهم معاً و أختار Au_ID ثم قم بضبط الترتيب حسب الاسم كما هو في المثال



الشكل 60 و هنا اختيار لجملة الاستعلام كما نرى أنها تعمل بدون مشاكل



الشكل ٦١ هنا نقوم باختيار الحقل الذي سوف يظهر و الحقل الذي سوف يكون قيمة للعنصر في الـ Drop Down List.

بعد ذلك نقوم بأعداد الـ Detail View و سوف نقوم بخطوات كما سبق و لكن مع تغيير جملة الاستعلام.

Configure Data Source - DetailSource

Define Custom Statements or Stored Procedures

Click a tab to create a SQL statement for that operation.

SELECT UPDATE INSERT DELETE

☒ SQL statement:

```
SELECT au_lname, au_fname, phone, address, city, state, zip, contract, GETDATE() AS ViewDate
FROM authors
where au_id=@au_id
```

Query Builder...

☐ Stored procedure:

byroyalty

< Previous Next > Finish Cancel

الشكل 62 هذه هي حملة الاستعلام للـ **Detail View** و كما نلاحظ أن هناك حقل اسمه **ViewDate** و أن هناك شرط وهو رقم المعرف للمؤلف

Configure Data Source - DetailSource

Define Parameters

The wizard has detected one or more parameters in your SELECT statement. For each parameter in the SELECT statement, choose a source for the parameter's value.

Parameters:

Name	Value
au_id	

Parameter source:

None

DefaultValue:

[Show advanced properties](#)

SELECT statement:

```
SELECT au_lname, au_fname, phone, address, city, state, zip, contract, GETDATE() AS ViewDate
FROM authors
```

< Previous Next > Finish Cancel

الشكل 63 هنا يتم اختيار مصدر بيانات الشرط. كما نرى و أعدنا أن هناك أكثر من نوع ،

Configure Data Source - DetailSource

Define Parameters

The wizard has detected one or more parameters in your SELECT statement. For each parameter in the SELECT statement, choose a source for the parameter's value.

Parameters:

Name	Value
au_id	DropDownList1.Select...

Parameter source:

ControlID:

DefaultValue:

[Show advanced properties](#)

SELECT statement:

```
SELECT au_name, au_fname, phone, address, city, state, zip, contract, GETDATE() AS ViewDate
FROM authors
```

< Previous Next > Finish Cancel

الشكل 64 نختار مصدر الشرط **Control** و سوف نختار العنصر الذي سوف يكون مصدر البيانات و هو في هذه الحالة **Drop Down List** أو كما هو مسمى في المثال **DropDownList1** ، نلاحظ أن هناك وصلة بلون أزرق **Show Advanced Properties** هذه الوصلة سوف تظهر بعض الخصائص المهم في عملية الربط بين ال**Drop Down list** و ال**Detail View** لنقم بالضغط على هذه الوصلة

Configure Data Source - DetailSource

Define Parameters

The wizard has detected one or more parameters in your SELECT statement. For each parameter in the SELECT statement, choose a source for the parameter's value.

Parameters:

Name	Value
au_id	DropDownList1.Select...

Parameter source: Control

Properties:

ControlID	DropDownList1
ConvertEmptyStringToNull	True
DefaultValue	
Direction	Input
Name	au_id
PropertyName	SelectedValue
Size	0
Type	Empty

PropertyName
A property name indicating the property from which to get the value. If none is specified, the ControlValueProperty attribute of ...

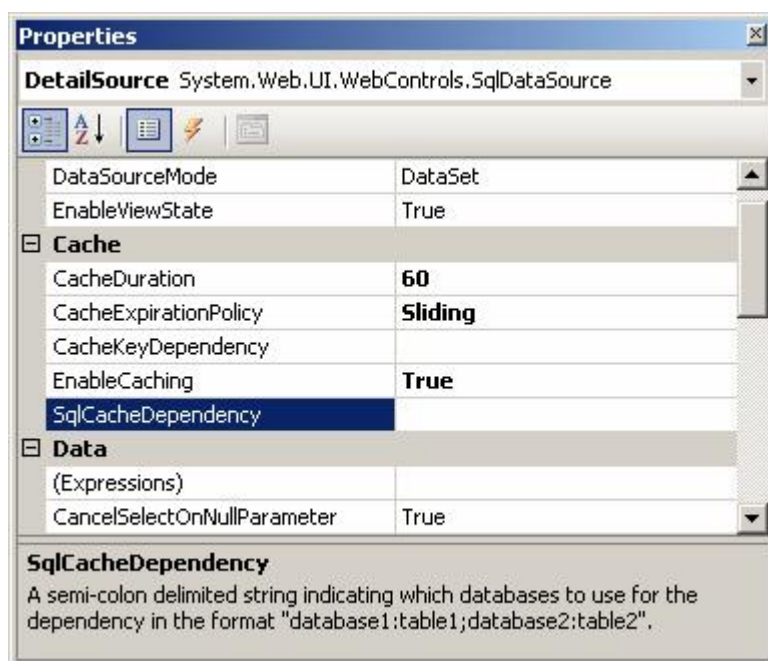
[Hide advanced properties](#)

SELECT statement:

```
SELECT au_name, au_fname, phone, address, city, state, zip, contract, GETDATE() AS ViewDate
FROM authors
```

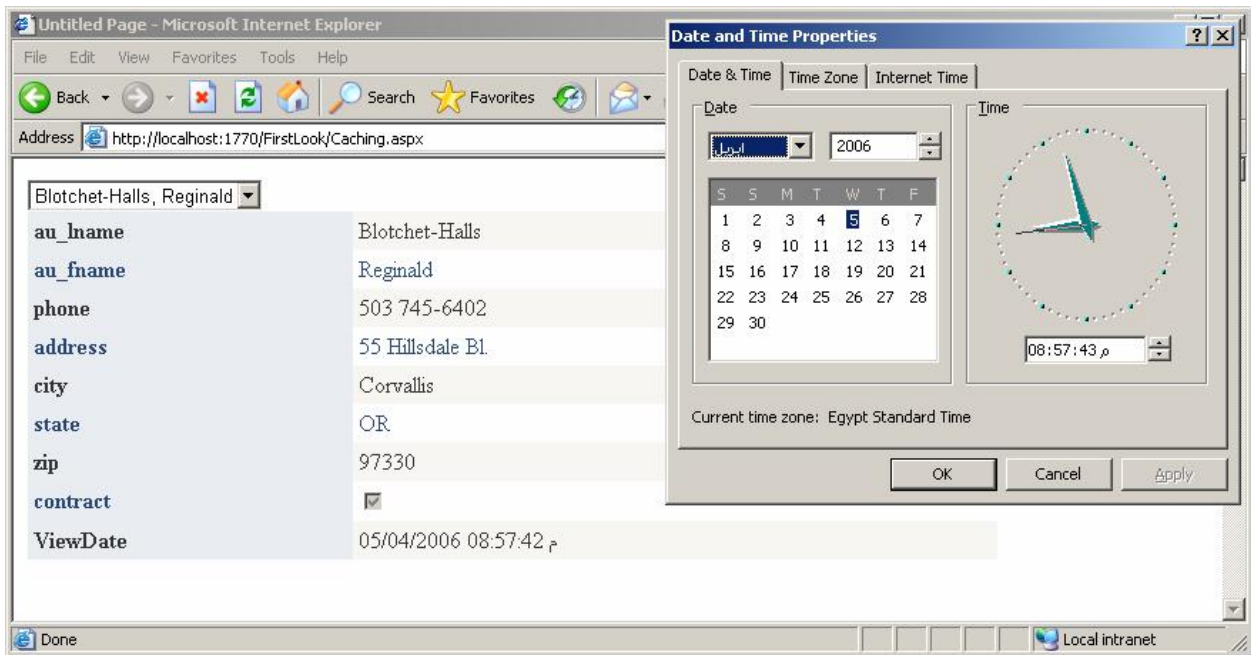
< Previous Next > Finish Cancel

الشكل 65 كما نلاحظ و كما ذكرنا من قبل أن الوصلة بالفعل ظهر مكانها لوحة خصائص مصغرة للأداة Drop Down List و التي منها نقوم بضبط الإعدادات المهمة على سبيل المثال في حالة أن العنصر كانت قيمة فارغة يقوم بتحويلها إلى Null، كما نلاحظ أن هناك وصلة Hide Advanced properties و هذه الخاصية تقوم بإخفاء لوحة الخصائص، و الآن لنقوم بالضغط على Next حتى نصل إلى نهاية المعالج.



الشكل 66 و الآن أهم ما في الموضوع خصائص Caching هذه الخواص سوف نجدها في مصدر البيانات الخاصة بالعنصر و كما نرى أننا سوف نقوم بتعديل مصدر البيانات الخاص بالـ Detail view وهو كما هو مسمى في المثال DetailSource و الآن قم بتعديل كلاً من الخصائص التالية

- الخاصية **CacheDuration** و هنا المقصود بهذه الخاصية هو مدة تخزين البيانات و التي بعدها سوف يقوم مصدر البيانات بأحضار البيانات من جديد، و يتم هنا ضبط الوقت بالثانية.
- الخاصية **EnableCaching** و هذه الخاصية المسأولة عن عملية تفعيل خاصية تخزين البيانات في مصدر البيانات.
- الخاصية **CacheExpirationPolicy** هنا خاصية مهم و من الممكن أن تشكل فارقاً كبيراً و هذه الخاصية لها اختياراً هما **Sliding** و الآخر **Absolute** و الآن لنقوم بشرح كل واحدة منهم
- **Sliding** : في هذه الحالة سوف يقوم مصدر البيانات بتخزين البيانات للفترة المحددة لتخزين البيانات لكن في حالة استدعاء بيانات من مصدر البيانات سوف يتم تمديد وقت التخزين الى المدة التي تم تحديدها، بمعنى أنه إذا قام المستخدم بعمل Refresh للصفحة سوف يقوم مصدر البيانات بحساب وقت التخزين من جديد، إذا قام المستخدم بعمل Refresh في الثانية ٥ ٤ و كانت مدة التخزين ٦٠ ثانية سوف يقوم مصدر البيانات بإعادة الوقت من جديدة أي أن الـ ٤ ثانية سوف تكون صفر ليبدأ العد من جديد.
- **Absolute** : في هذه الحالة سوف تكون تخزين البيانات لمدة محددة ليس أكثر بمعنى تحت أي ظرف من الظروف سوف يقوم مصدر البيانات بأحضار البيانات من جديد بعد الوقت المحدد لتخزين البيانات



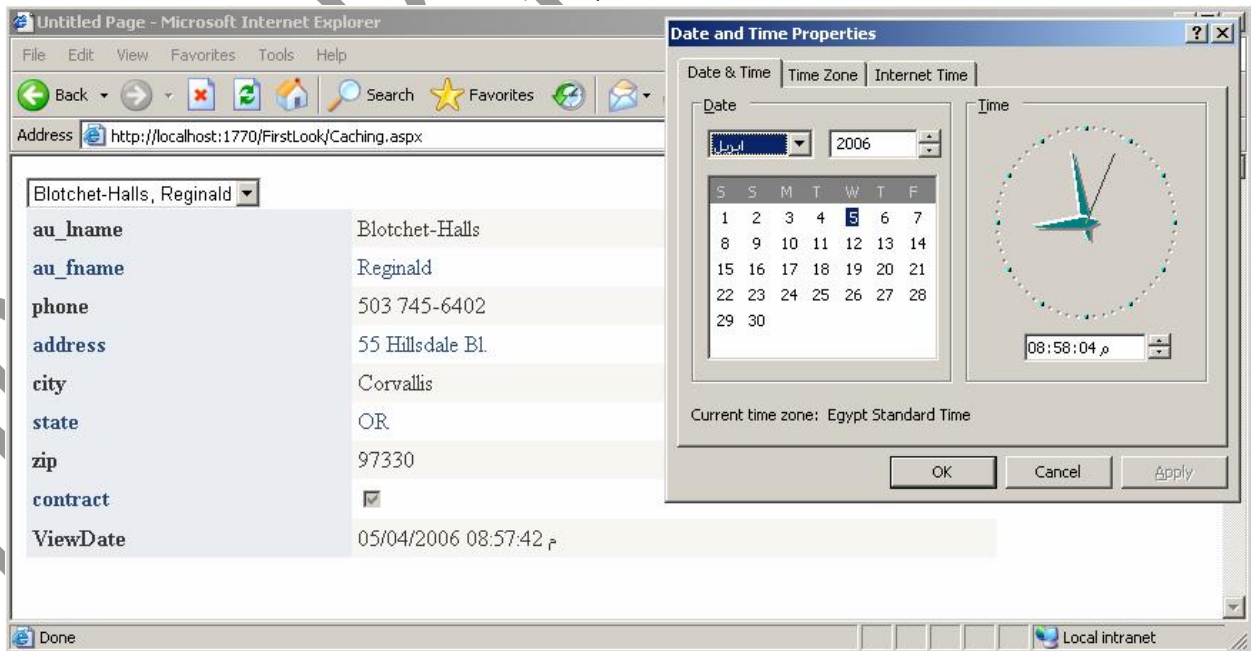
الشكل 67 و هنا المثال بعد تشغيله و كما نرى الوقت الذى تم احضار البيانات فيه

Greene	Morningstar
Blotchet-Halls	Reginald
Ynkamoto	Δkiko

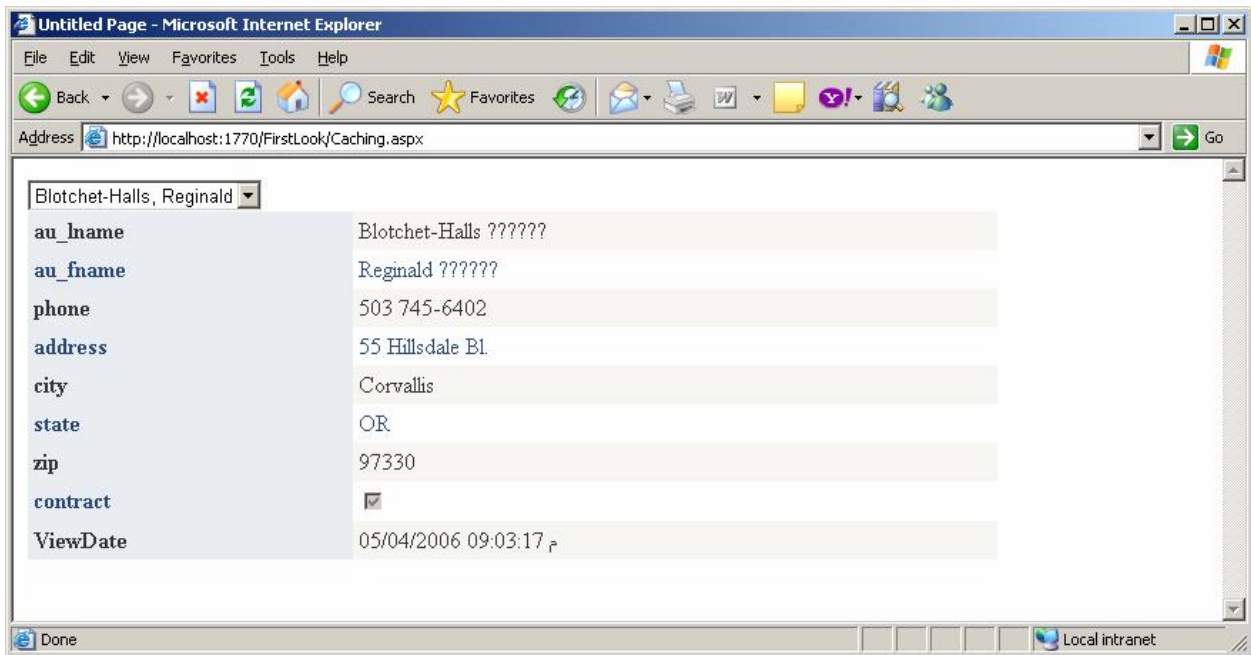
الشكل ٦٨ - نقوم بالدخول على جدول البيانات و نقوم بتعديل اسم المؤلف المعروض

Greene	Morningstar	61
Blotchet-Halls ?????	Reginald ?????	50
Ynkamoto	Δkiko	41

الشكل ٦٩ شكل بعد التعديل



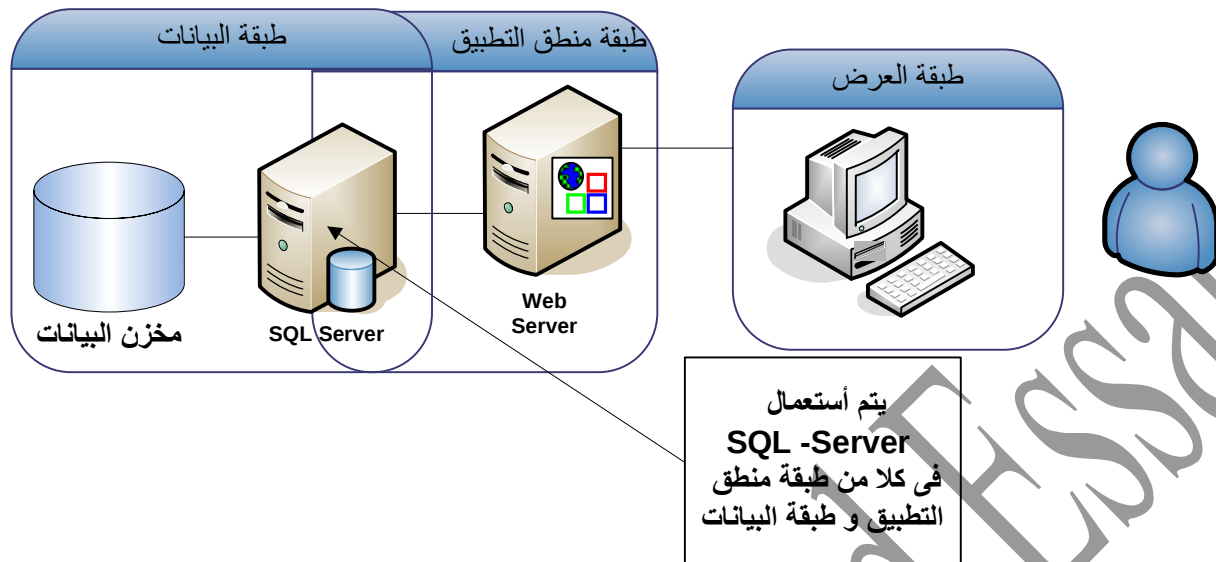
الشكل 70 البيانات لم تتغير مع أننا قمنا بتغيير البيانات و هذا هو نتيجة لتخبة البيانات Caching



الشكل 71 كما نرى أن تحديث البيانات قد ظهر و لكن بعد وقت المحدد لتخبة البيانات.

هل لاحظت أن أسم المؤلف في الـ Drop Down list لم يتغير؟؟؟
السبب بكل بساطة أن الـ Visual Studio.net 2005 يقوم بمعالجة أرتداد الصفحات Post Back كان هذا مطلوب عمله لتوفير الوقت مستهلك في أحضار البيانات بالإضافة الى أنه تحافظ على حالة العنصر عند إعادة عرض الصفحة. هناك نوع آخر من تخبة البيانات يسمى **Caching with Dependency** و معنى هذا المصطلح أنه سوف يتم تخبة البيانات و لكن لن تخبة البيانات هاكذا و لكن سوف يتم التأكد "هل تغيرت البيانات؟" في حالة أن البيانات تغيرت سوف يقوم مصدر البيانات بأحضار البيانات التي تغيرت، هذا الموضوع يتطلب أعدادات خاصة في قاعدة البيانات و هي أنه سوف يتم إنشاء جدول بالعناصر و حالتها و يتم إنشاء حدث **Trigger** يقوم بمراقبة حدوث أى تغيير للجدول معين ثم يقوم الحدث **Trigger** بتخزين وقت آخر تعديل يقوم مصدر البيانات بعد ذلك بالتأكد من أن البيانات لم تتغير من هذا الجدول.

على ما أظن أن هناك بعض المحترفين الذين سوف يقولون أننا لاتعامل مع البيانات بهذا الشكل و أننا نتعامل بمنطق تعدد الطبقات أو كما يسمى **N-Tier** ، و أن كل هذه الأمكانيات لا تنعهم، بكل بساطة مايكروسوفت قامت بتخصيص أداة جديدة لهذا الغرض و لا تقلق من ناحية كيفية الاستعمال لان هناك دائما مثال لتوضيح هذه الأمور، و لكن أولاً لابد من توضيح معنى كلمة "تعدد الطبقات" أو الـ **N-Tier** ، الفكرة بكل بساطة أنه يتم تقسيم البرنامج من ناحية التصميم و البناء الداخلى الى أقسام الأكثر شيوعاً هو البناء المعتمد على 3 طبقات **"3 Tiers Architecture"** ، الطبقة الأولى من البرنامج تقوم طبقة البيانات و فيه يتم وضع كل ما يتعلق بالبيانات "أضافة ، تعديل، حذف ، عرض" و لا يكن به أكثر من ذلك، الطبقة الثانية هي طبقة منطق التطبيق **Application logic or Business tier** و هنا يكون موجود المحرك الحقيقي للبرنامج بمعنى أن كل قوانين عمل البرنامج تكون موضوعة فى هذه الطبقة ، الطبقة الأخيرة هي طبقة العامل مع المستخدم و هذه الطبقة هي الوصلة بين منطق التطبيق **Business logic** و المستخدم، الشكل التالى يوضح العلاقة بين الطبقات و بعضها.



الشكل ٧٢ كما نرى هنا أن التطبيق مقسم الى عدة طبقات و كل طبقة لها وظيفة فكما نرى أن المستخدم يتعامل مع طبقة العرض "طبقة المستخدم" و هذه الطبقة تتعامل مع طبقة منطق التطبيق و طبقة المنطق تتعامل مع طبقة البيانات.

كما نلاحظ أننا اعتبرنا SQL server في طبقتي منطق التطبيق و طبقة البيانات، لماذا؟
لانه يمكنك من داخل الـ SQL Server أن تضع جزء من منطق التطبيق الخاص بك على شكل دوال Functions و إجراءات مخزنة Stored procedure و أحداث Trigger هذه الأماكن يمكنك من أن تضع جزء كبير من طبقة منطق التطبيق في الـ SQL Server، و الآن لنبدأ مثال الى حد ما كبير أو بمعنى آخر أننا سوف نقوم لأول مرة بكتابة أكواد منذ بداية هذا الفصل.

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public class DBManagement
{
    private DBManagement()
    {
    }

    private static string GetConnectionString()
    {
        return
        ConfigurationManager.ConnectionStrings["pubsConnectionString"].ConnectionStri
        ng;
    }

    private static DataSet ExecuteProcedure(string strCommandName, params
    SqlParameter[] Inputs)
    {
        SqlConnection con = new SqlConnection(GetConnectionString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText = strCommandName;
        cmd.CommandType = CommandType.StoredProcedure;
        con.Open();
        cmd.Connection = con;
        foreach (SqlParameter tempParam in Inputs)
        {
            cmd.Parameters.Add(tempParam);
        }
    }
}
```

```

    }

    SqlDataAdapter adbt = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    adbt.Fill(ds);

    cmd.Dispose();
    adbt.Dispose();
    con.Close();
    return ds;
}
public static DataSet GetAuthorInformation()
{
    return ExecuteProcedure("spGetAuthors");
}
public static DataSet GetAuthorInformationWithPermission(int
intPremission)
{
    if (intPremission==1)
        return ExecuteProcedure("spGetAuthors");
    else
        return ExecuteProcedure("spGetAuthorByIDWithPermission");
}
}

```

كود التعامل مع قاعدة البيانات بصيغة الـ C#

```

Imports Microsoft.VisualBasic
Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Public Class DBLayer_VB
    Private Sub New()

    End Sub
    Private Shared Function GetConnectionString() As String
        Return
        ConfigurationManager.ConnectionStrings("pubsConnectionString").ConnectionStri
        ng
    End Function
    Private Shared Function ExecuteProcedure(ByVal strCommandName As String,
Optional ByVal Inputs As SqlParameter() = Nothing) As DataSet
        Dim con As New SqlConnection(GetConnectionString())
        Dim cmd As New SqlCommand
        cmd.CommandText = strCommandName
        cmd.CommandType = CommandType.StoredProcedure
        For Each tempParam As SqlParameter In Inputs
            cmd.Parameters.Add(tempParam)
        Next
        Dim adpt As New SqlDataAdapter(cmd)
        Dim ds As New DataSet
        adpt.Fill(ds)
        cmd.Dispose()
        adpt.Dispose()
        con.Close()
        Return ds
    End Function
    Public Shared Function GetAuthorInformation(ByVal strAuthorID As String)

```

```

As DataSet
    Return ExecuteProcedure("spGetAuthors")
End Function

Public Shared Function GetAuthorInformationWithPermission(ByVal
intPermission As Integer) As DataSet
    If intPermission = 1 Then
        Return ExecuteProcedure("spGetAuthors")
    Else
        Return ExecuteProcedure("spGetAuthorByIDWithPermission")
    End If
End Function
End Class

```

كود التعامل مع قاعدة البيانات بصيغة الـ VB.net

```

CREATE PROCEDURE dbo.spGetAuthors AS
select * from authors
RETURN

```

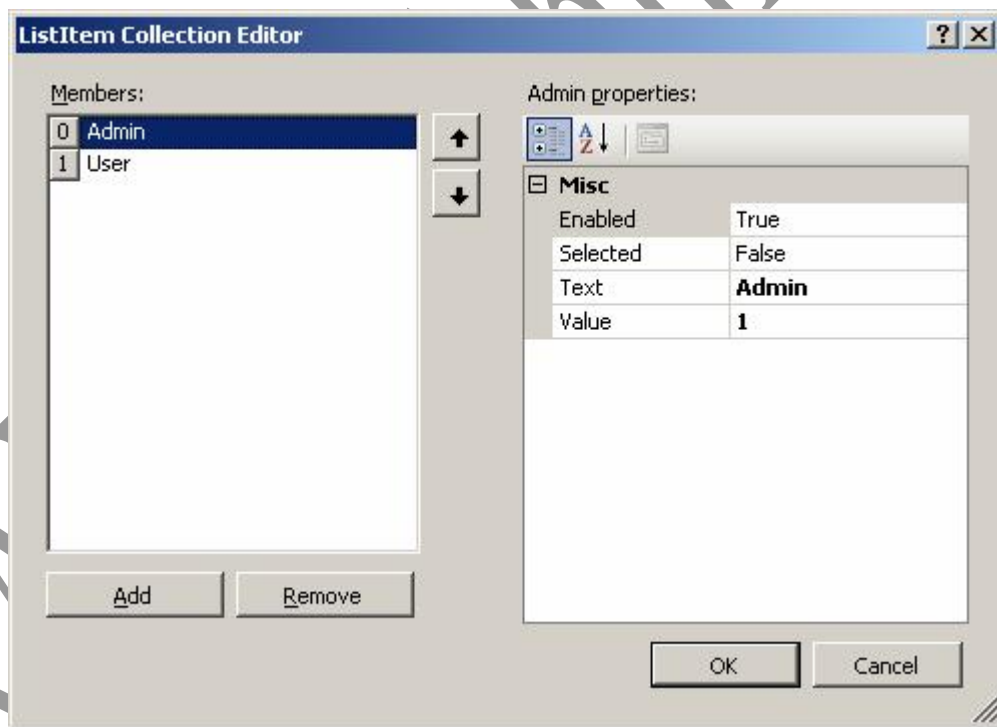
الأجرائة المخزنة GetAuthors

```

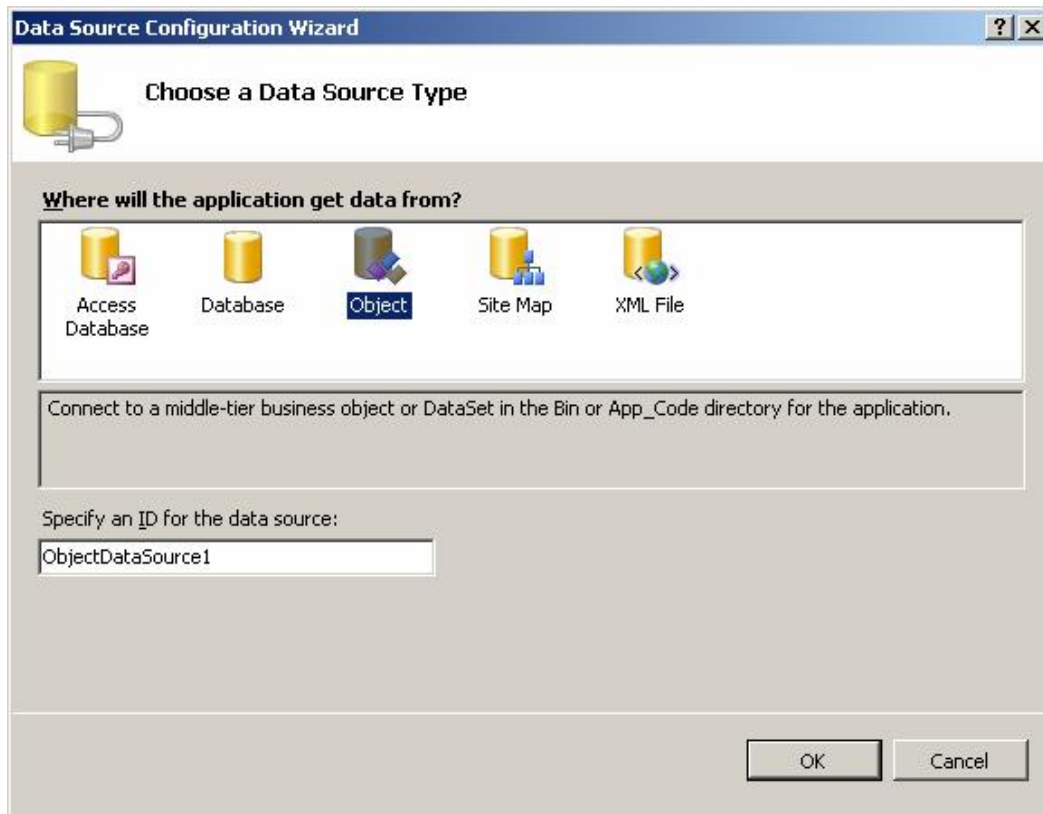
CREATE PROCEDURE dbo.spGetAuthorByIDWithPermission AS
select au_fname, au_lname from authors
RETURN

```

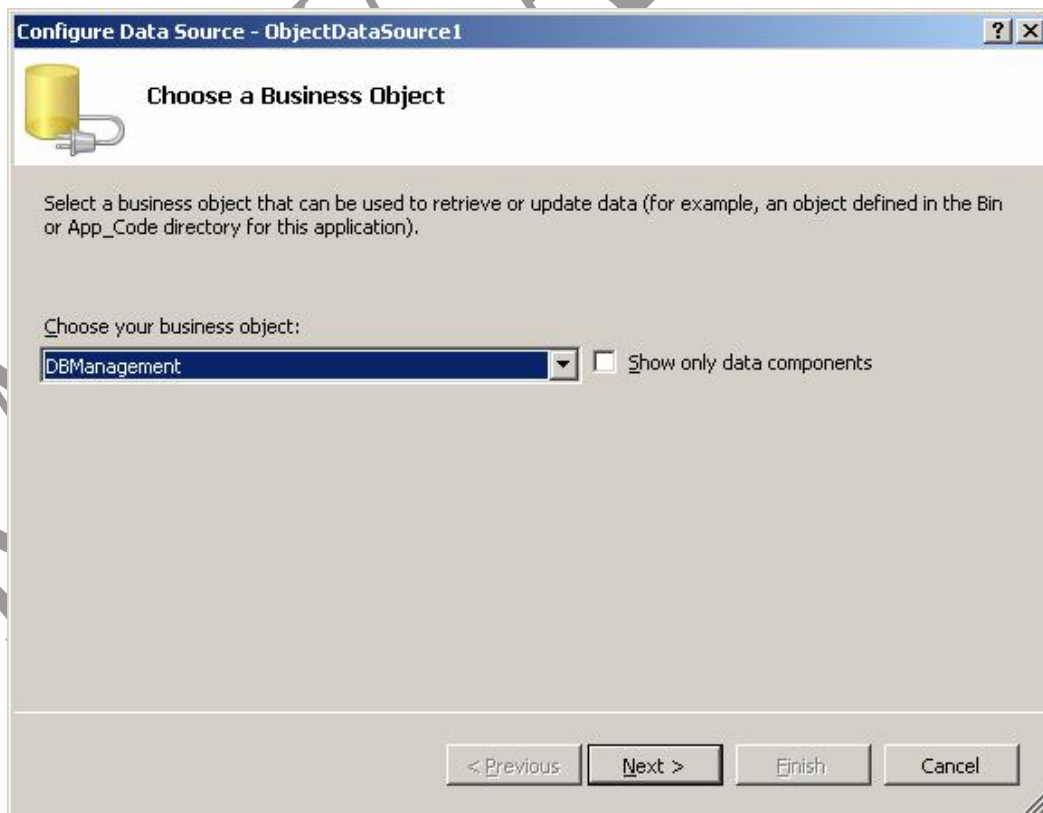
الأجرائة المخزنة GetAuthorByIDWithPermission



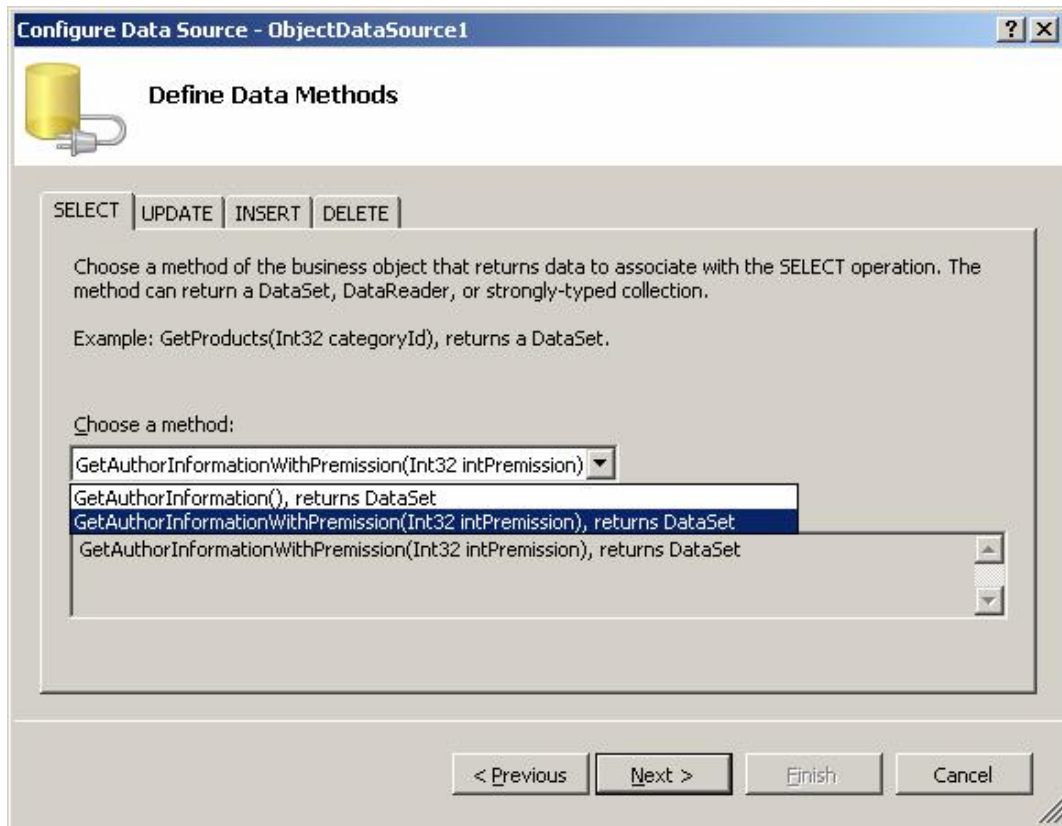
الشكل 73 نقوم بإضافة DropDownList ونقوم بأعداد محتوياتها كما نرى و هي Admin و القيمة الخاصة به هي ١ و User و القسم الخاص به صفر



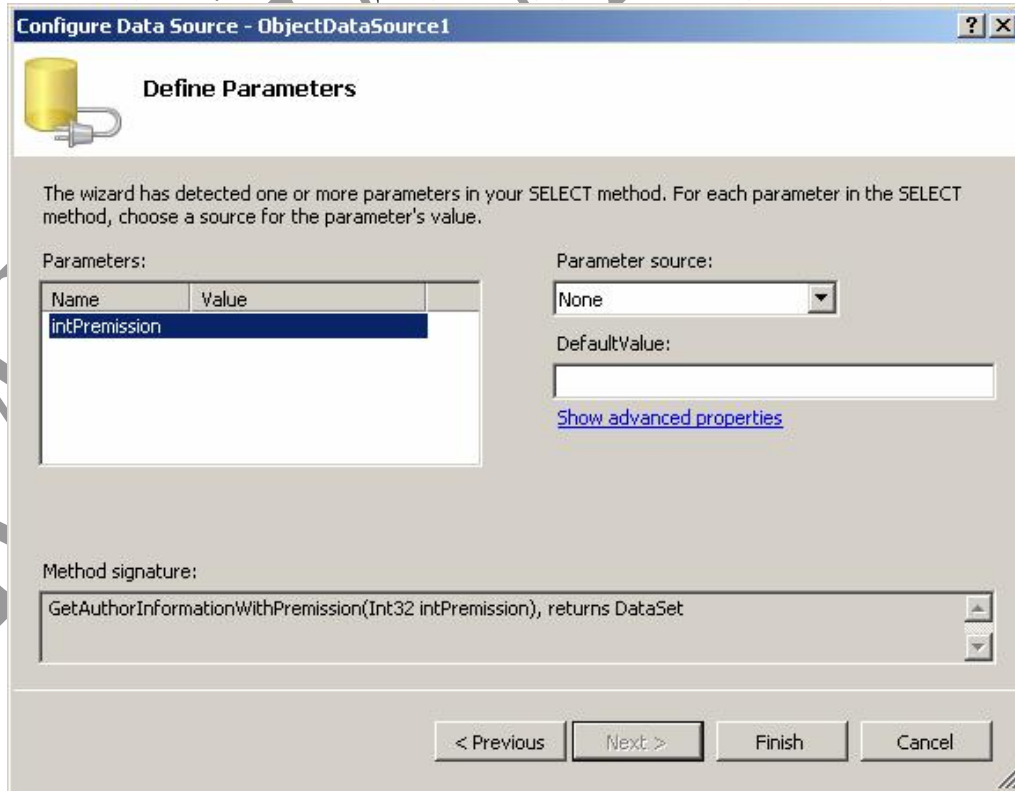
الشكل 74 نقوم بوضع Grid View على الصفحة و بعد ذلك نقوم بأعداد مصدر البيانات و كما نرى و على غير العادة أننا سوف نغير مصدر البيانات من Database الى Object و نقوم بتسمية ثم نضغط على OK



الشكل 75 من هنا سوف نختار المكتبة التي تتعامل مع البيانات أو بمعنى أصح الـ Business Tier التي سوف تمتد الـ Interface بالبيانات



الشكل ٧٦ و هنا يتم اختيار الدالة التي سوف ترجع لنا البيانات أو الدالة التي تحتوي على منطق البرنامج و هنا في حالتنا هنا وضعنا الكود الخاص بالمنطق البرنامج في الدالة **GetAuthorInformationWithPermission** وهذه الدالة تستقبل متغير من نوع رقم Integer ، السؤال هنا كيف سوف تستقبل هذا المتغير أو من أين سوف يتم أمداد هذه الدالة بهذا المتغير؟



الشكل ٧٧ الأجابة عن السؤال السابق في هذه النافذة و كما نرى أن هناك قائمة في يسار الشاشة تحتوي على المتغيرات التي تحتاجها الدالة و على اليمين قائمة بالمصادر التي يمكن أن نأخذ منها البيانات للدالة.

Configure Data Source - ObjectDataSource1

Define Parameters

The wizard has detected one or more parameters in your SELECT method. For each parameter in the SELECT method, choose a source for the parameter's value.

Name	Value
intPermission	DropDownList1.Select...

Parameter source: Control

ControlID: DropDownList1

DefaultValue:

[Show advanced properties](#)

Method signature:

GetAuthorInformationWithPermission(Int32 intPermission), returns DataSet

< Previous Next > Finish Cancel

الشكل ٧٨ و الآن نقوم باختيار القائمة المنزلة Drop Dow List و التي تحتوي على نوع المستخدم ثم الضغط على Finish

GridView Tasks

Auto Format...

Choose Data Source: ObjectDataSource1

Configure Data Source...

Refresh Schema

Edit Columns...

Add New Column...

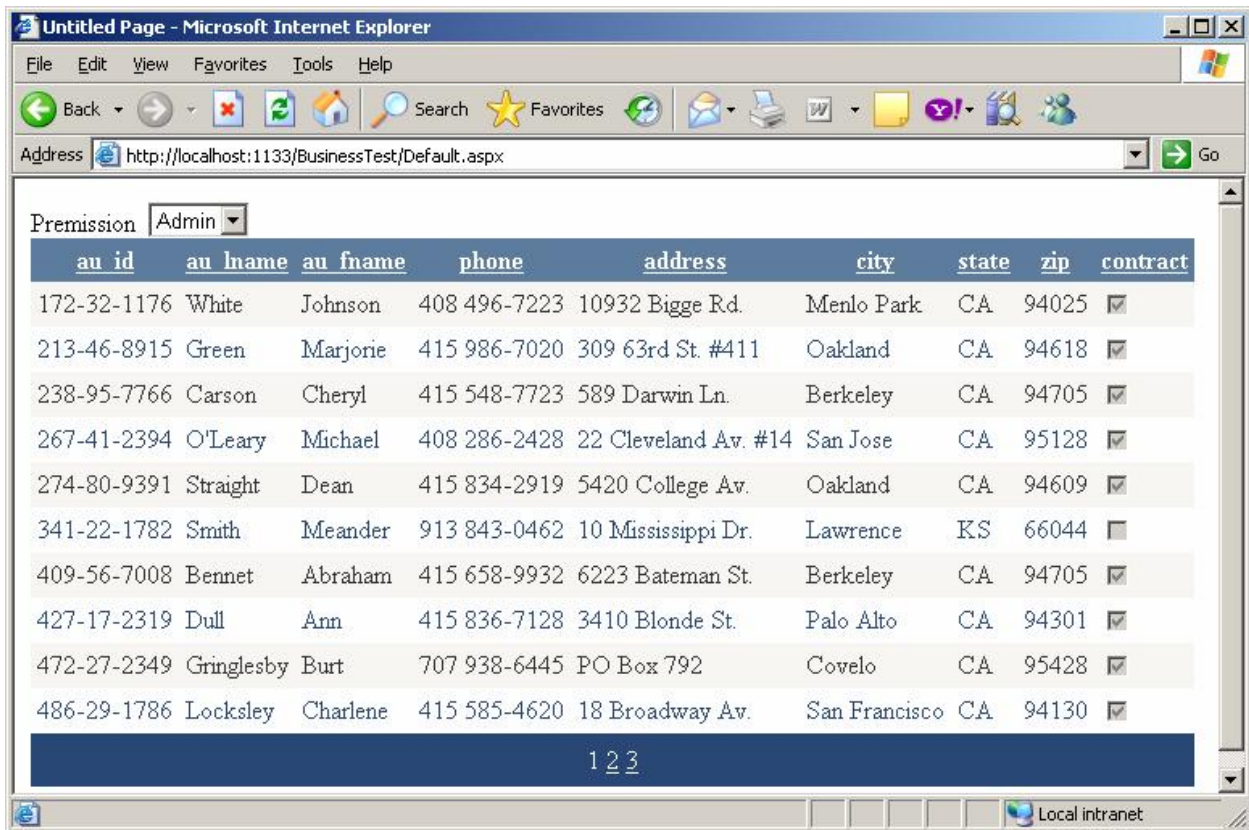
☒ Enable Paging

☒ Enable Sorting

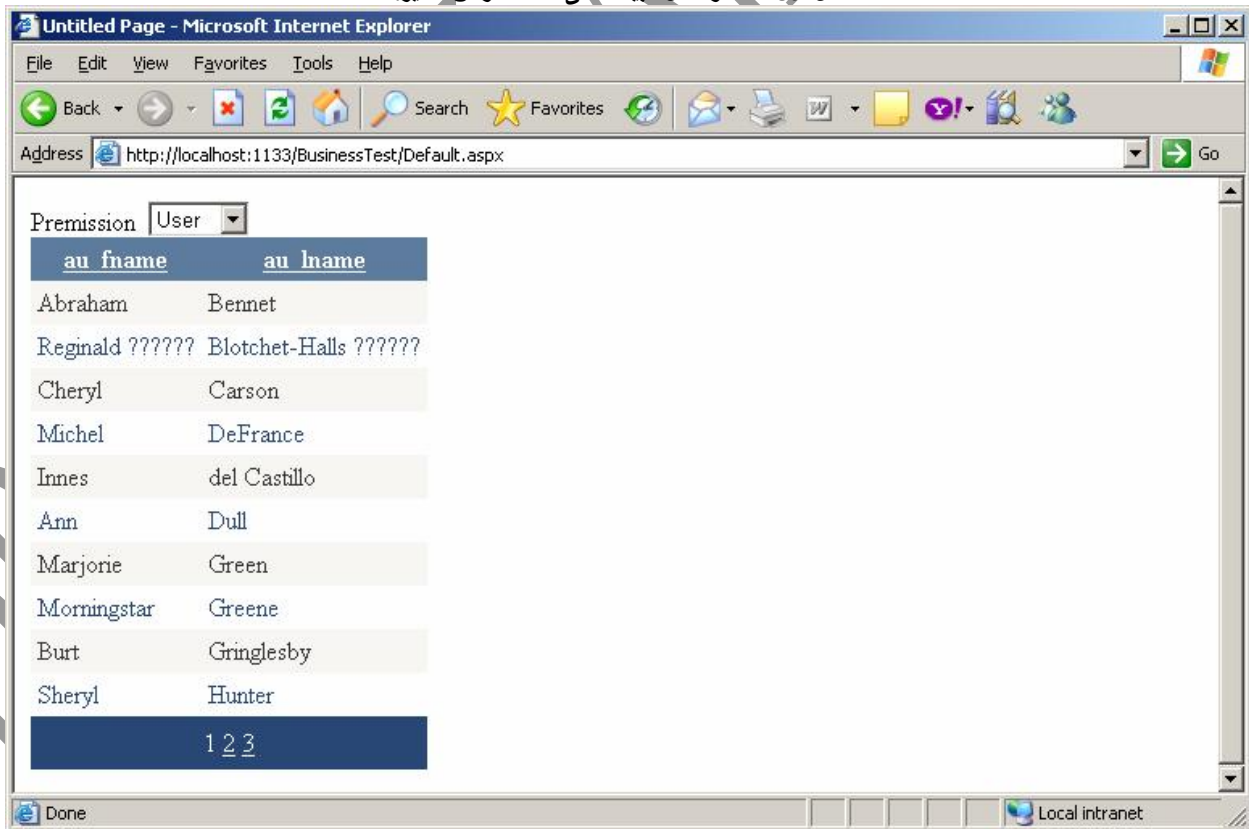
☐ Enable Selection

Edit Templates

الشكل ٧٩ شئ رائع آخر وهي أنه مازال بإمكاننا استعمال خاصية السرد و تقسيم الى صفحات في الـ Grid View



الشكل 80 هذا هو شكل البيانات في حالة العرض لمدير.



الشكل 81 شكل في حالة العرض لمستخدم عادي

و الان و بعد أنتهاء المثال لنوضح الأمور قليل، فى المثال السابق أردنا وضع قانون صغير لعمل التطبيق الا وهو أن هناك نوعان من المستخدمين (مدير / مستخدم) ، قمنا بعمل مكتبة تحتوى على هذا القانون و هذا ما يسمى بالطبقة منطق التطبيق هذه المكتبة تقوم بالاتصال و أحضار البيانات طبقا لقانون الذى وضعناه للتطبيق، و بعد ذلك يقوم الفيجوال ستوديو بإنهاء العمل مع البيانات من حيث ضبط مصادر البيانات للدوال الخاصة بمكتبات منطق التطبيق أو مصدر البيانات لأدوات العرض مثل Grid View و Detail view و هكذا.

و الآن ننتقل الى موضوع آخر مفيد جدا لمن يريد تأمين قاعدة البيانات و هى تشفير جملة الاتصال فى الملف Web.Config ، يجب أولا توضيح ما سيتم عمله لحماية جملة الاتصال، بطل بساطة فى بداية تحميل الصفحة سوف نقوم بفكر تشفير جملة الاتصال Decrypt the connection string ، و عند الانتهاء نقوم بعمل تشفير لجملة الاتصال Encrypt the Connection string و لكن فى أول مرة سوف نقوم بعمل صفحة مؤقتة تقوم بتشفير جملة الاتصال، حتى يتسنى لنا إمكانية ما سبق ذكره و لكن قبل هذا سوف نقوم بعمل دالتين واحدة لتشفير جملة الاتصال و الأخرى لفكر التشفير.

```
public static void EncryptConnectionString()
{
    Configuration config =
    System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(
    Request.ApplicationPath);
    // فتح ملف web.config

    ConfigurationSection section = config.Sections["connectionStrings"];
    // تحديد النطاق المراد تشفيره
    section.SectionInformation.ProtectSection(
    "DataProtectionConfigurationProvider");
    // تحديد مخصص التشفير و تشفير النطاق الذي تم اختياره
    config.Save();
    // حفظ الملف web.config
}
```

```
Private Sub EncryptMyConnectionString()
    Dim config As Configuration = _
    System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(_
    Request.ApplicationPath)
    // فتح ملف web.config
    Dim section As ConfigurationSection = config.Sections("connectionStrings")
    // تحديد النطاق المراد تشفيره
    section.SectionInformation.ProtectSection(_
    "DataProtectionConfigurationProvider")
    // تحديد مخصص التشفير و تشفير النطاق الذي تم اختياره
    config.Save()
    // حفظ الملف web.config
End Sub
```

```
<connectionStrings
configProtectionProvider="DataProtectionConfigurationProvider">
```

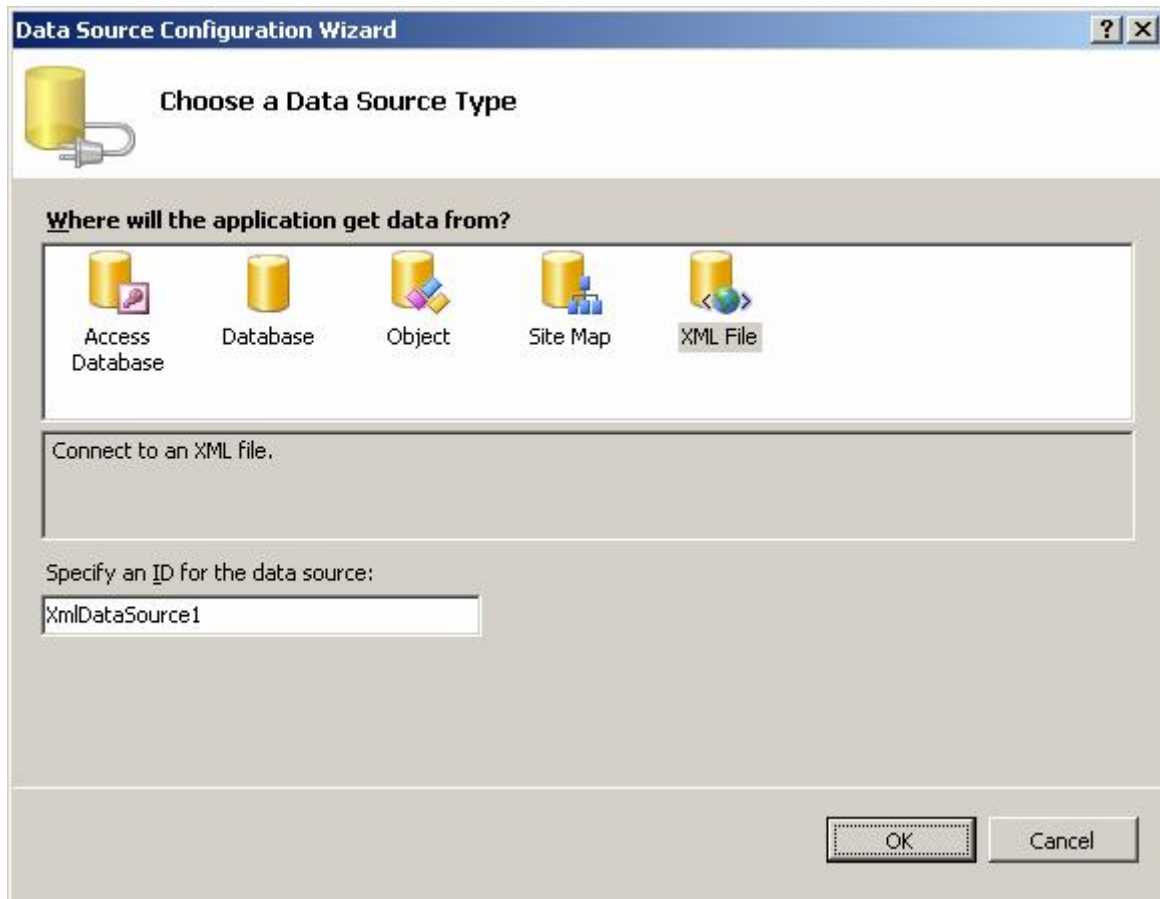
```
<EncryptedData>
<CipherData>
```

```
<CipherValue>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA5KTsb5/qvEyHjwCtfMWR1wQAAAAACAAAA
AAADZgAAqAAAAABAAAADep0UAEYD4/sCOUtrPTYcAAAAAASAAACgAAAAEAAAAPoSeBL0Mk6nK8Kb0
r2Q3q34AQAAAdYY5dqi5avWAT7ho0DwLvcPBbm7mkLp/+FPedccvJmX0JdR6zqNEEOXrIT8/e2WkaT
Wfxrum2MDwo/E2YGTW7/Cu4tV4V26tKwxwPiy2oUh0KEjYwkIVEeuQU2QzT1qQ27jSRjUEZnFnw+P
Z2B9g3+xROUAzewtzFwfKpn3tJwEUcHgEGecPL0kIwU0FZYCbSzQwdLgk88q+bvjEqzVCR180z1du
G9rNqJsUvYVgmiSTy017q4n2/3CeJldysCo8kV1Pz85c1JTzplgstwyZVi0myAECjGLpU2A3e7fJE
IUnQphxmIsW/9Yyi0H3ntUSqVbFo7gN0oz8T0vY45y00tfo099ApYsZMI40h8i2H5RzoB290bE4p0
2KNzIf0F/N0Fqp4zHjU7sNJJ2ejXoujKgh802p2sj3+tdi4+S3x08UoaQVJ3KFmP1x0JnMsLBK799
qS0K6+Pb1Wzp0tIOc9GqIEECwVIFnhjtp+Vyl12kCDW++FQ5da28GQjZULT8Lgwau0bVXweGgrk00
04kNmRQuIqR50LRh8BA3TusPVASo9QPzIyDreSbS6grzB3dQrqiHp+fgA2wRBh/m/2pU9ZNcsHUCF
TGZ2WtKqjFZLGP4fqUVio3J31iU0zMW1k5vmPmLojefvzxyg9PK9nXhhR0pj/cj1Ig5FAAAAPPu0Y
MVAJeVFT9UqRV28WDFNXbf</CipherValue>
</CipherData>
</EncryptedData>
</connectionStrings>
```

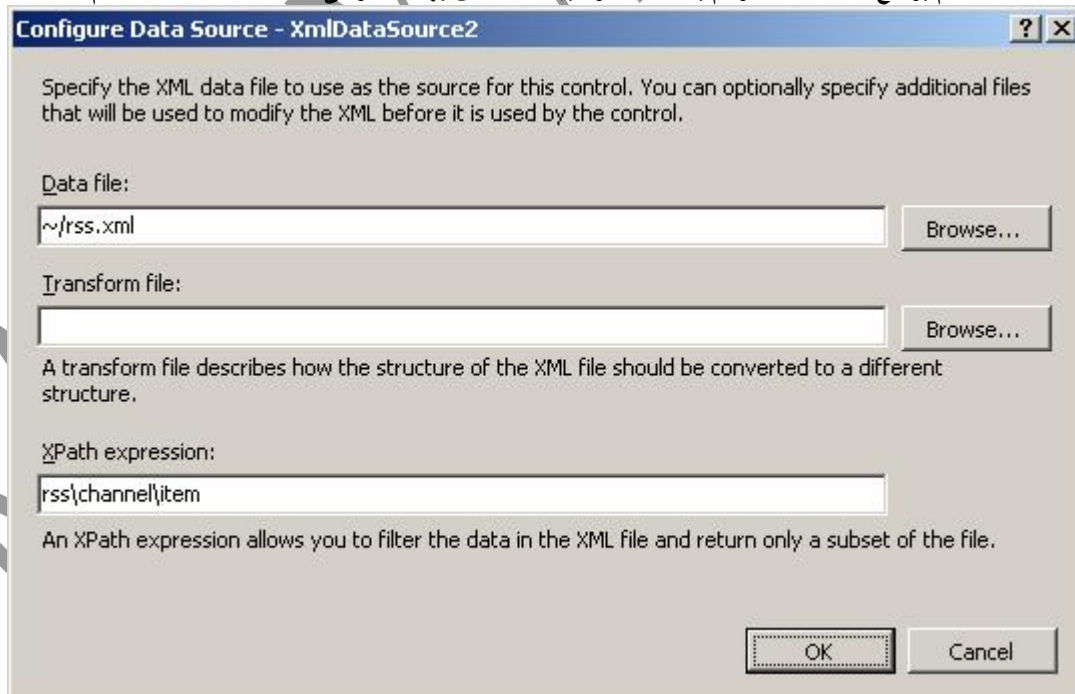
```
private void DecryptMyConnectionString()
{
    Configuration config =
System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(
Request.ApplicationPath);
    فتح ملف web.config
    ConfigurationSection section = config.Sections["connectionStrings"];
    تحديد النطاق المراد فك تشفيره
    section.SectionInformation.UnprotectSection();
    فك تشفير النطاق الذي تم اختياره
    config.Save();
    حفظ الملف web.config
}
}
```

```
Private Sub DecryptMyConnectionString()
Dim config As Configuration = _
System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(_
Request.ApplicationPath)
    فتح ملف web.config
Dim section As ConfigurationSection = config.Sections("connectionStrings")
    تحديد النطاق المراد فك تشفيره
section.SectionInformation.UnprotectSection()
    فك تشفير النطاق الذي تم اختياره
config.Save()
    حفظ الملف web.config
End Sub
```

و الآن لنرى شئ آخر جديد و مفيد جدا وهو التعامل مع الـ XML ، الآن يوجد أداة للتعامل مع الـ XML و لنوضح كيفية التعامل مع الـ XML بمثال صغير جدا وهو عرض ملف XML خاص بنظام متابعة RSS يمكن الحصول على صفحات مشابهه من أى موقع لديه هذا الرمز **RSS** و الآن لنتابع المثال



الشكل 82 قم بوضع Data List و قم بأعداد مصدر البيانات الخاص بها كما هو في هذا الشكل XML File ثم OK



الشكل 83 بعد ذلك نقوم بتحديد الملف المراد التعامل معه كمصدر بيانات و نقوم بتخصيص مسار البيانات (كما نعلم أن الـ XML بداخله يحتوي على ترتيب شجري لتقسيم البيانات)

```
<asp:DataList ID="DataList1" runat="server" DataSourceID="XmlDataSource1">
```

```
<itemtemplate>
  <b><#XPath("title")%> </b><br />
  <i><#XPath("description") %></i>&nbsp;<#XPath("pubDate")%><br />
  <a href='<#XPath("link") %>'>Link</a><br />
</itemtemplate>
</asp:DataList><asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/RSS.xml"
XPath="rss/channel/item"></asp:XmlDataSource>
```

بعد ذلك نقوم بإضافة كود العرض كما هو موضح في **Template item** و كما نرى أننا نأخذ البيانات من ما يسمى **XPath** و هذا يبدو مختلفاً حيث كان في مصادر البيانات الأخرى حيث أن مصادر البيانات الأخرى كانت تستخدم ما يسمى **بالـDataBind**.

و في النهاية هذا الفصل أود أن نذكر سويًا ملخصاً لما تعلمناه في هذا الفصل:

١. التعامل مع الأداة **Detail view**
٢. التعامل مع الأداة **Grid View**
٣. التعامل مع الأداة **SQL data source**
٤. التعامل مع الأداة **Object data source**
٥. التعامل مع الأداة **XML data source**
٦. تشفير جملة الاتصال
٧. فكرة تعدد الطبقات **N-Tier**
٨. تخزين البيانات **Caching** و كيفية الاستفادة منها
٩. بناء جمل الاستعلام على أدوات البيانات
١٠. كيفية الربط بين أدوات البيانات الموجودة على الصفحة