

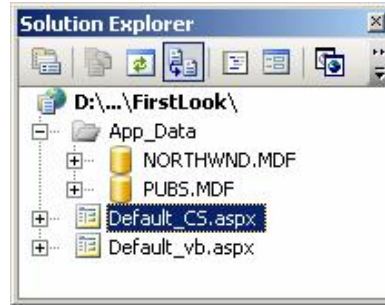
الفصل الرابع : قواعد البيانات و أدوات عرض البيانات.

Database, ADO.net and data show tools.

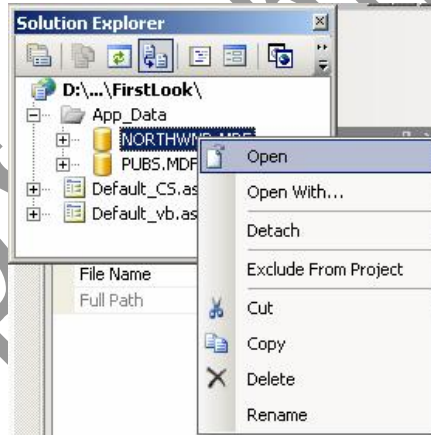
فى هذا الفصل و الذي أعتبره من أهم الفصول التى تهتم كل مطور يعمل فى السوق و يدري ما هى متطلبات السوق ؛ سيتم تناول ببعض الأدوات الجديدة لعرض البيانات و الأماكنيات الجديدة فى هذه الأدوات ؛ و بعد ذلك سوف نرى ما هو الجديد فى الـ (ADO.net)

نبدأ بـ

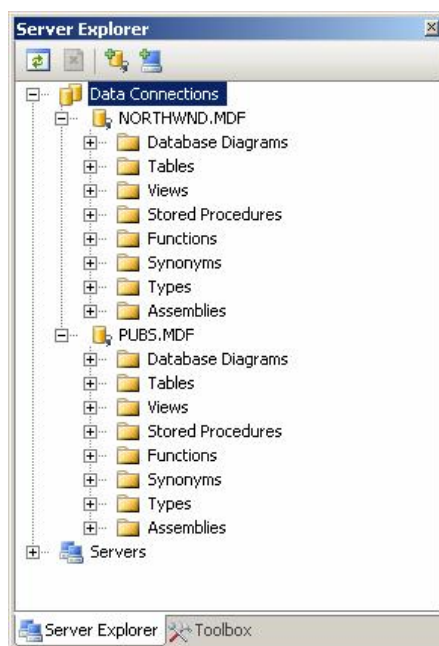
١ - أدوات التعامل مع البيانات الجديدة .



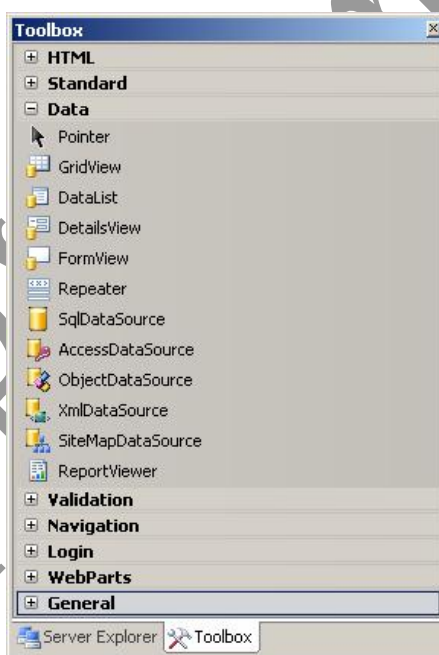
الشكل ١ [أولا عليك البدء بمشروع و تضع الملفات كما هو موضح أعلاه]



الشكل ٢ [نقوم بفتح قاعدة البيانات للتأكد من أن الإتصال سليم]

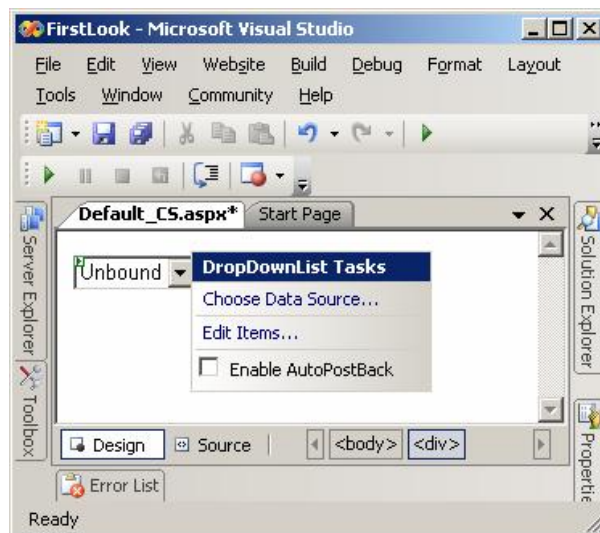


الشكل ٣ [عند بدأ الاتصال سوف نرى أن كل قاعدة بيانات تظهر في مجلد بنفس أسم قاعدة البيانات]

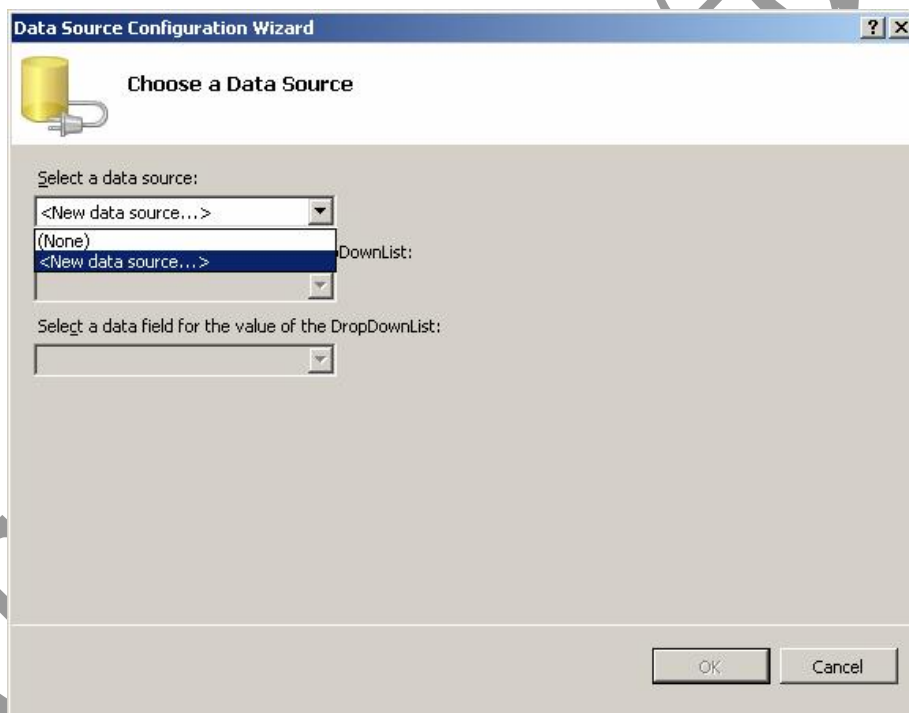


الشكل ٤ [هذه هي مجموعة الأدوات الخاصة بالتعامل مع قواعد البيانات - لن نتعامل مع كل الأدوات ولكن سوف نتعامل مع الأدوات التي تحتوي على شئ جديد]

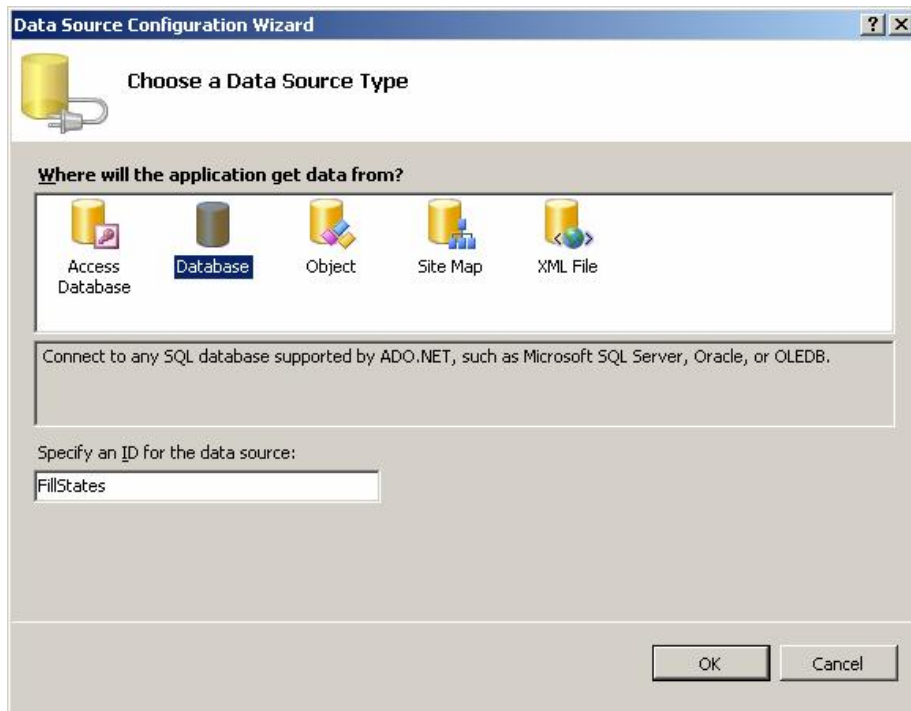
وسيتم توضيح هذا المثال التالي الذي سيتم تطبيقه على قاعدة البيانات (Pubs) و هي قاعدة بيانات خاصة بدور النشر و المؤلفين و الكتب و ما الى ذلك ، وأيضا سنوضح فيه كيفية التعامل مع الأداة الجديدة (Grid View) و هذه الأداة تعتبر بديل للـ (Data Grid) و هي أداة أكثر من رائعة بالإضافة الى الأداة الجديدة (Details View) و الآن لنبدأ المثال:-



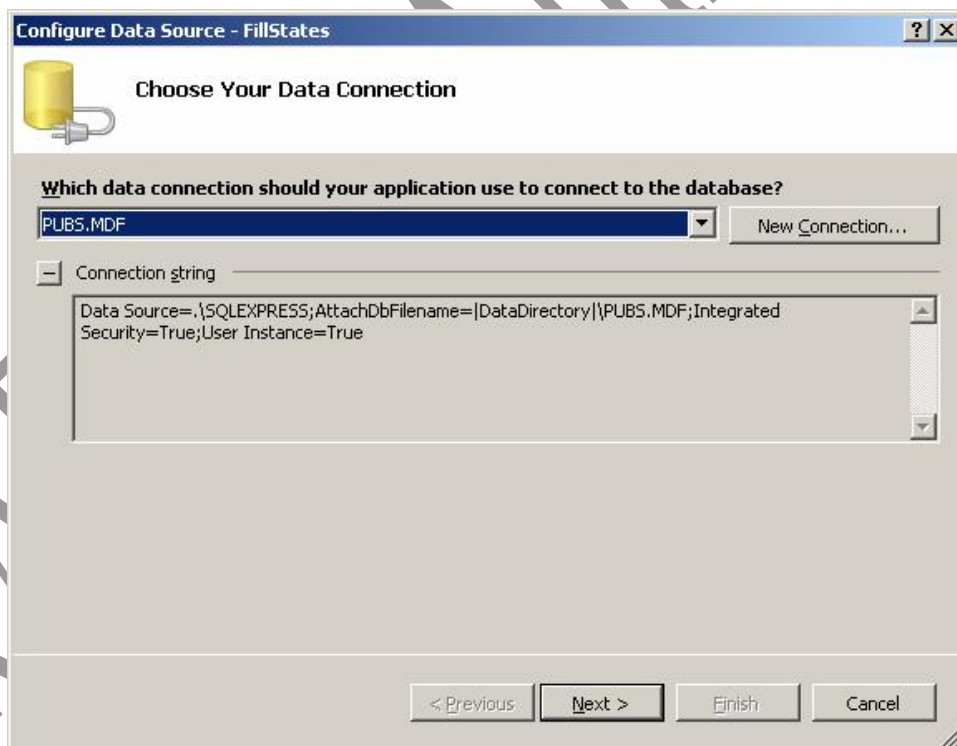
الشكل ٥ [نقوم بوضع (DropDownList) كما هو موضح و نضغط على الـ (Smart Tag) و منه نختار
[(Choose Data Source)



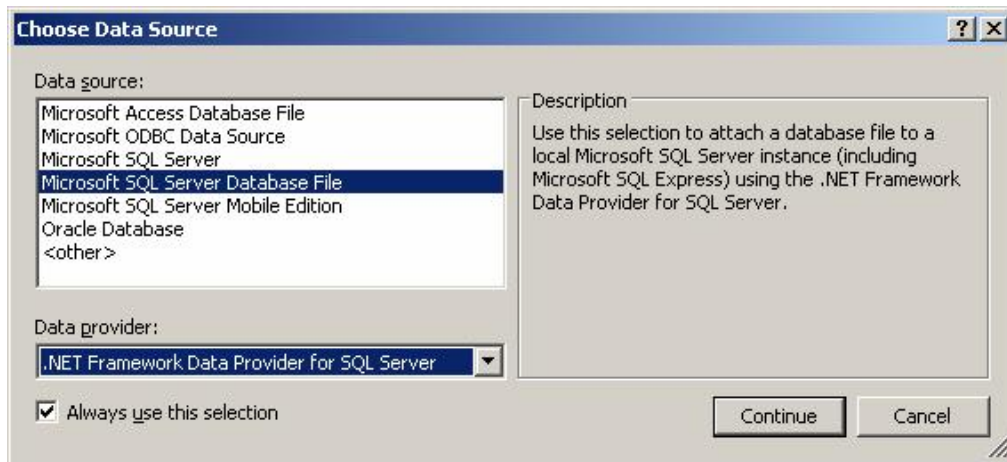
الشكل 6 [سوف تظهر لنا شاشة معالج ضبط و إعداد مصدر البيانات و سنختار منها مصدر بيانات جديد (New Data source)]



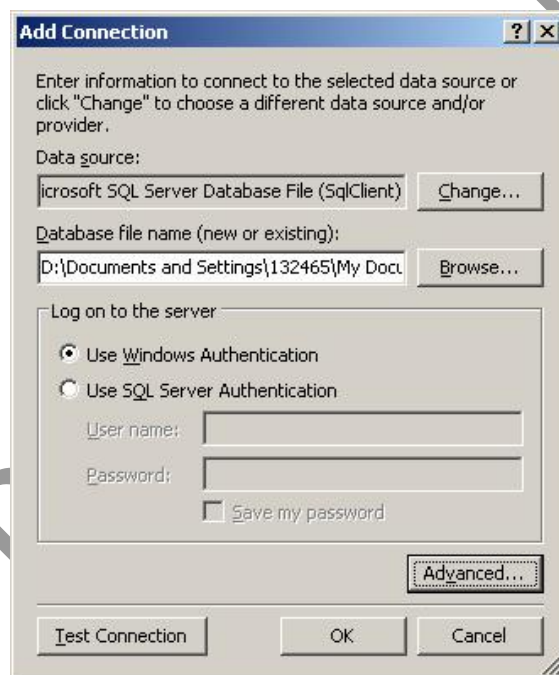
الشكل 7 [نختار نوع قاعدة البيانات و هذا طبقاً لمحركات قواعد البيانات الموجودة (Database Provider) ثم بعد ذلك نكتب اسم مصدر البيانات و نضغط على (OK)]



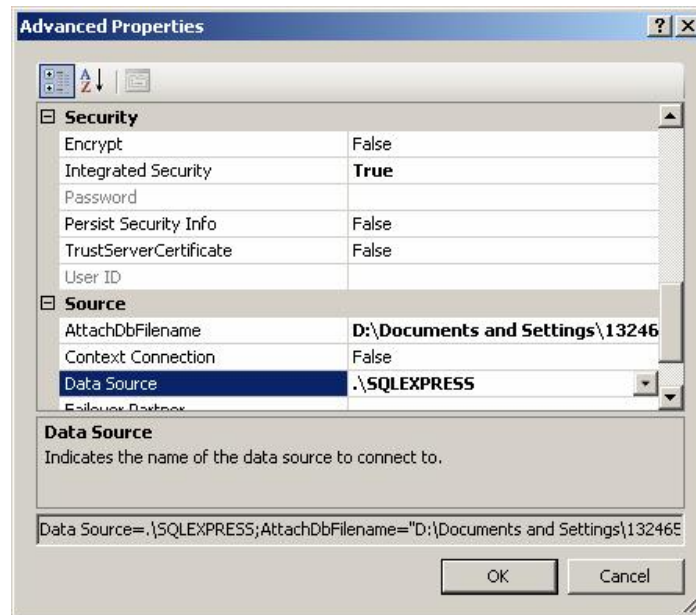
الشكل ٨ [سوف ننشئ جملة الإتصال عن طريق إختيار اسم قاعدة البيانات وبعدها سوف يقوم المعالج بإكمال الموضوع دون حاجة لتدخلك ؛ أما إذا كنت تود ضبط بعض الإعدادات الخاصة إضغط على (New Connection)]



الشكل ٩ [ستختار من هذه النافذة محرك قاعدة البيانات الذي تود التعامل معه ، و بعد ذلك اضغط على (Continue)]



الشكل ١٠ [تختار منه ملف قاعدة البيانات ، و بعد ذلك تقوم بالضغط على (Advanced) لضبط الإعدادات المتقدمة]



الشكل ١١] بعد الإنتهاء من شاشة خصائص الإعدادات المتقدمة اضغط على (OK) ؛ و سوف تعود إلى شاشة (Add Connection) و منها اضغط على (Test Connection) للتأكد من سلامة الاتصال و عدم وجود مشاكل]



الشكل 12]يظهر هنا ميزة جديدة توفر كثيراً من الوقت حيث يخبرك فيها المعالج ما بين إما حفظ جملة الاتصال في ملف (Web.config) أو أن يقوم بتركها في خصائص أداة مصدر البيانات، قم باختيار (Yes) و بعد ذلك أدخل الأسم الذي 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" />
```

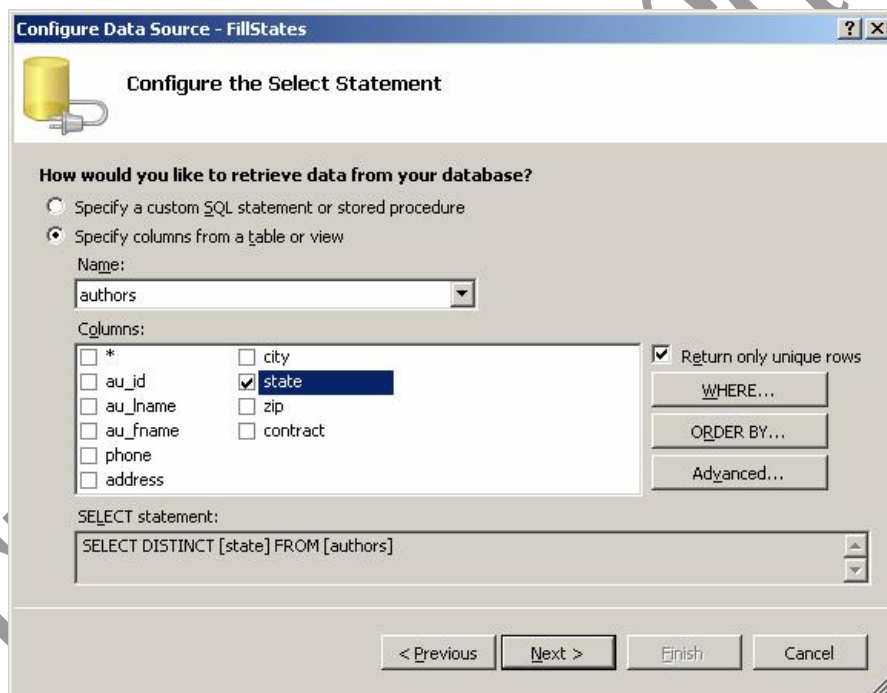


```
</connectionStrings>
```

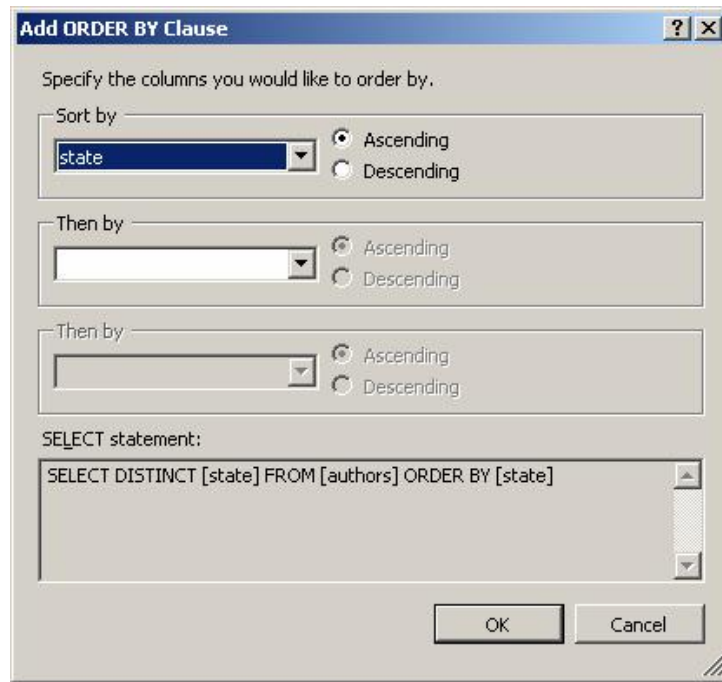
الشكل السابق يوضح شكل جملة الاتصال في الملف (Web.config)

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

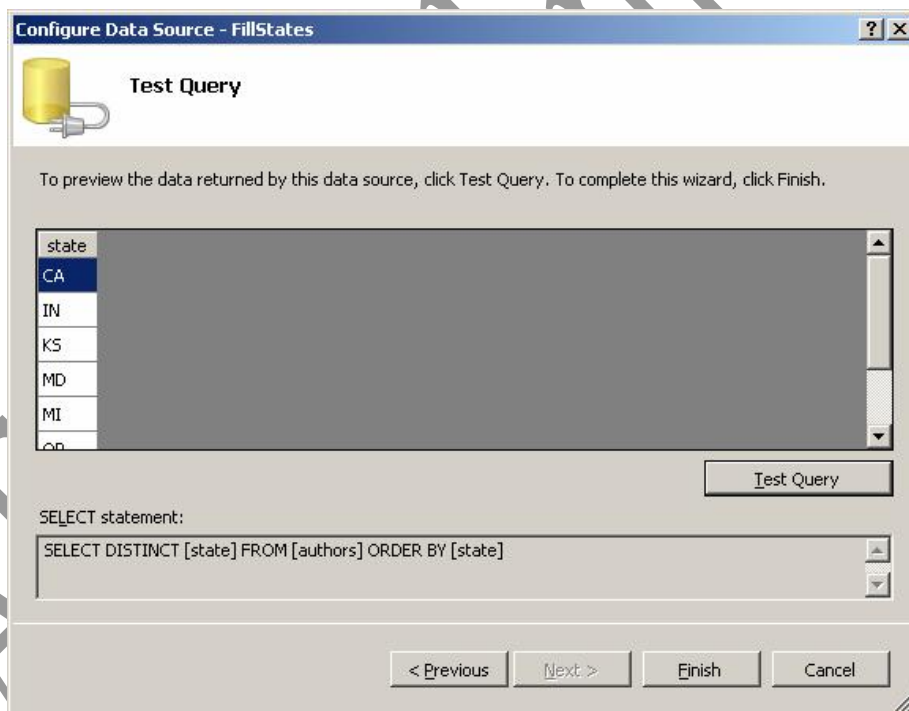
الشكل السابق عبارة عن عرض (Source view) للصفحة و هذا الجزء خاص بالأداة (SQL Data Source) ، و فيها خاصية جملة الاتصال و كما نرى أنها (ConnectionString:pubsConnectionString) و هذا يعني أنه ليقوم بعملية الاتصال سوف يحضرها من الملف (Web.config) . وبعد أن قمنا إعداد أداة الاتصال سنكمل بقية خطوات المثال .



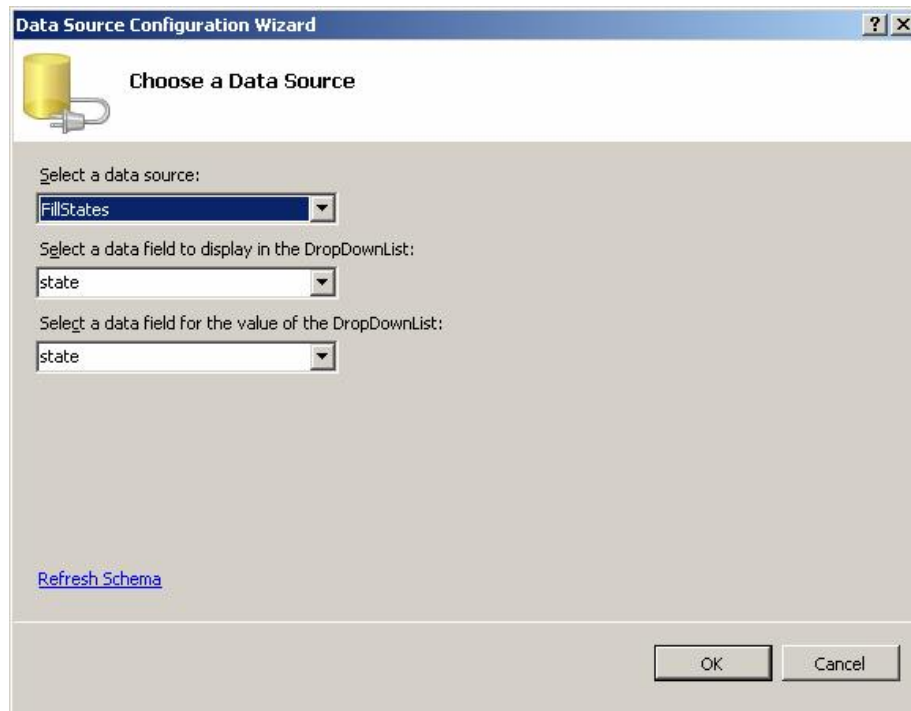
الشكل ١٣] نقوم بإعداد البيانات التي سوف تعرضها الأداة ، و كما نرى أننا اخترنا أن نقوم بعض الأعمدة من الجداول ، و الآن سوف نختار العمود المراد عرضه كما نرى هو (State) ثم سنقوم بتفعيل (Return only unique row) و هذا لمنع التكرار في النتائج بعد ذلك ثم يتم الضغط على (Order by)



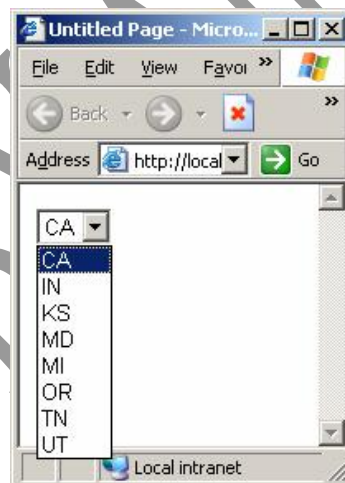
الشكل ١٤ [تختار ترتيب عرض البيانات أضغط على (OK) سوف تعود الى الشكل السابق ثم أضغط على (Next)]



الشكل 15 [يتم اختبار جملة الاستعلام للتأكد من سلامتها]



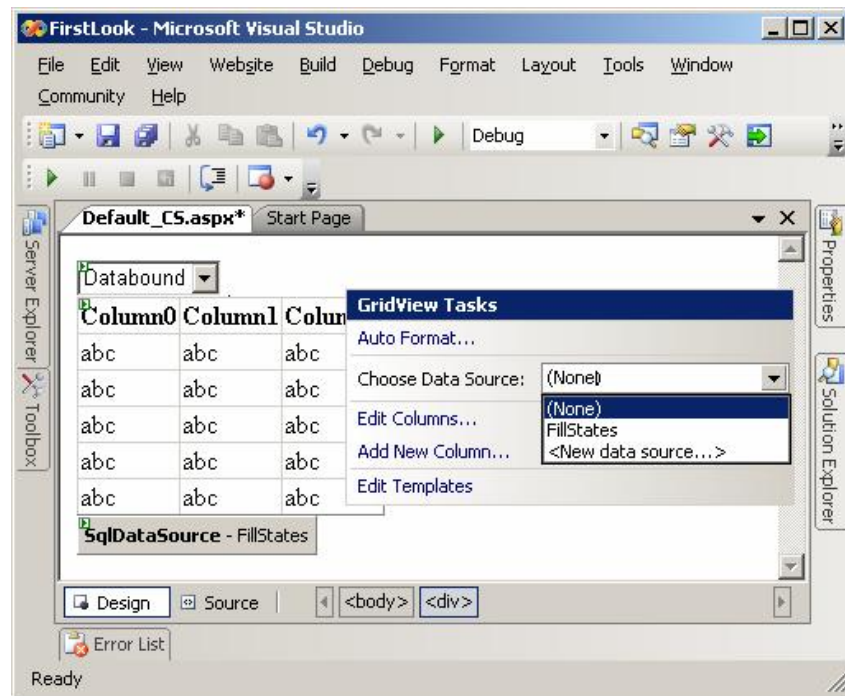
الشكل 16 [يتم تحديد كلاً من العمود الذي سوف يتم مع عرض البيانات (Displayed value) بالإضافة للعمود الذي سوف يتم فيه وضع القيمة (Value of selected item)]



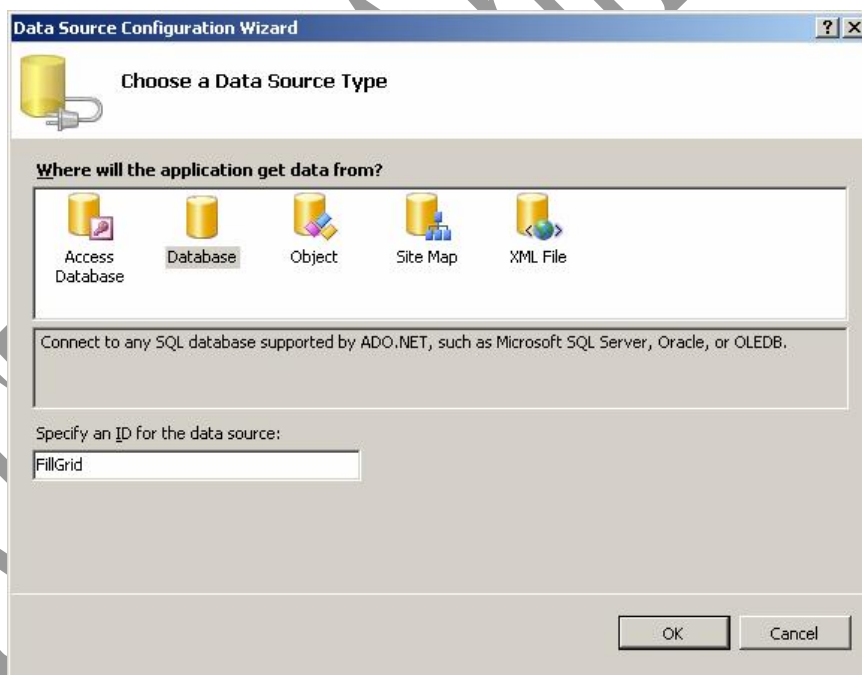
الشكل ١٧ [هذا هو شك المثال بعد التنفيذ]



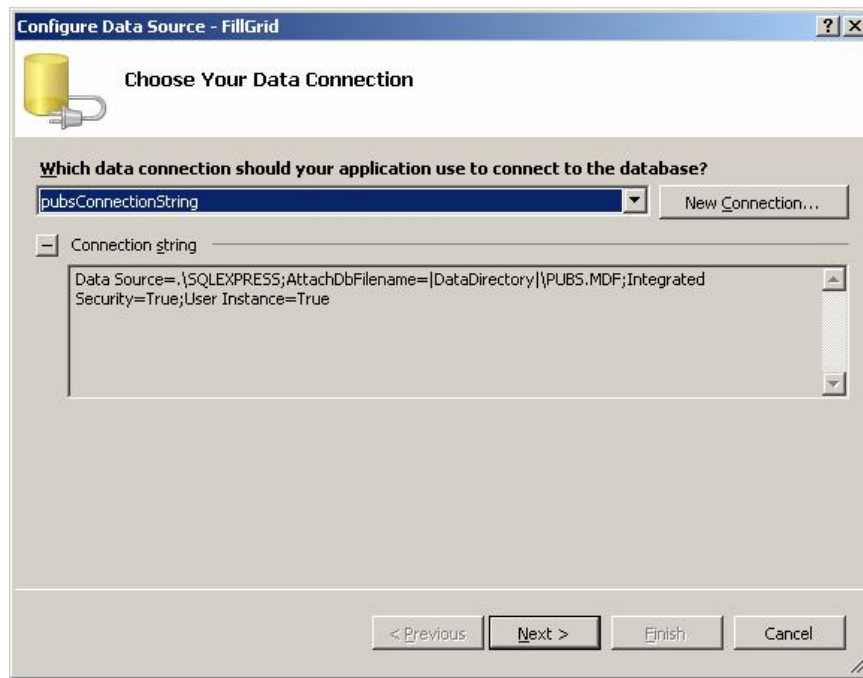
الشكل ١٨ [وهنا نقوم بوضع الأداة (GridView) على الصفحة لنبدأ التجربة الفعلية لأدوات البيانات الجديدة]



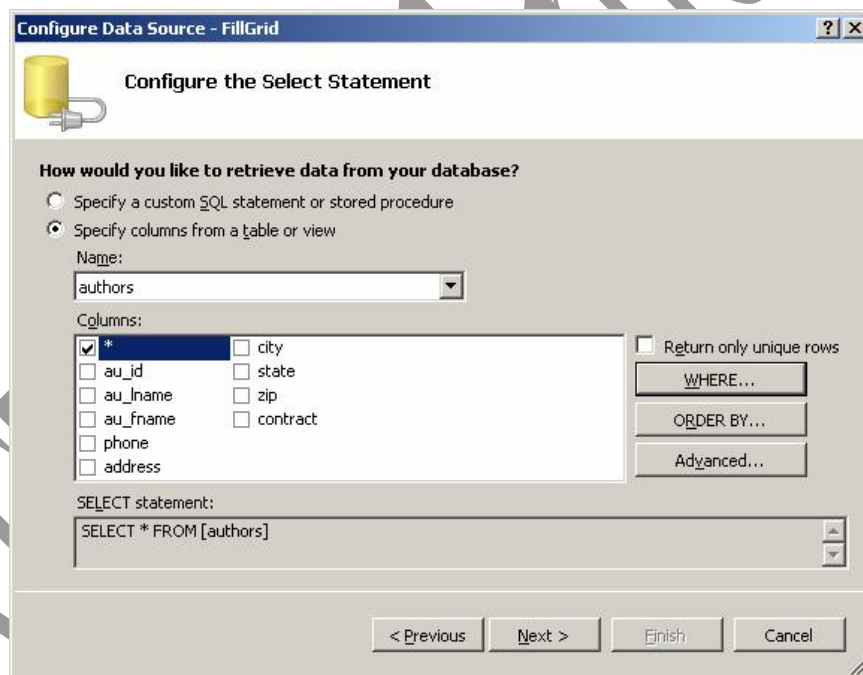
الشكل ١٩ [نقوم بفتح الـ (Smart Tag) و نختار من (Choose Data source) مصدر بيانات جديد (New Data source) كما هو موضح في الشكل]



الشكل ٢٠ [نختار نوع المحرك واسم أداة مصدر البيانات]



الشكل ٢١] سوف نستخدم جملة الاتصال المحفوظة من قبل في (Web.config) [



الشكل ٢٢] نختار الأعمدة التي سيتم عرضها ، و بعد ذلك نضغط على (Where) [

الشكل ٢٣ وفي هذه المرحلة سيتم توفير الكثير من الوقت وذلك عن طريق ربط (DropDownList) بالـ (GridView) ؛ ويتم ذلك عن طريق إختيار العمود الذي سوف يتم عليه الشرط ثم إختيار مصدر قيمة العمود (Source) ؛ و بعد ذلك قم بإختيار الأداة التي سوف يتم أخذ القيمة منها بعد ذلك اضغط على (ADD)

الشكل ٢٤ [اضغط على (OK) ؛ و بذلك تكون قد قمت ببناء الشرط و ربطه بالأداة]

الشكل ٢٥] هو الشكل النهائي بعد تحديد الشرط و الاداة مصدر قيمة الشرط اضغط (Next) [

الشكل ٢٦] وهنا نقوم بإدخال قيمة لإختبار جملة الاستعلام [

Configure Data Source - FillGrid

Test Query

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

au_id	au_name	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	<input checked="" type="checkbox"/>
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	<input checked="" type="checkbox"/>
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose	CA	95128	<input checked="" type="checkbox"/>
274-80-9391	Straight	Dean	415 834-2919	5420 College Av.	Oakland	CA	94609	<input checked="" type="checkbox"/>
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	<input checked="" type="checkbox"/>
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	<input checked="" type="checkbox"/>
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	<input checked="" type="checkbox"/>
672-71-3249	Yokomoto	Akiko	415 935-4228	3 Silver Ct.	Walnut Creek	CA	94595	<input checked="" type="checkbox"/>
724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av.	Oakland	CA	94609	<input type="checkbox"/>
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA	94612	<input checked="" type="checkbox"/>
756-30-7391	Karsen	Livia	415 534-9219	5720 McAuley St.	Oakland	CA	94609	<input checked="" type="checkbox"/>
846-92-7186	Hunter	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	<input checked="" type="checkbox"/>
893-72-1158	McBadden	Heather	707 448-4982	301 Putnam	Vacaville	CA	95688	<input type="checkbox"/>

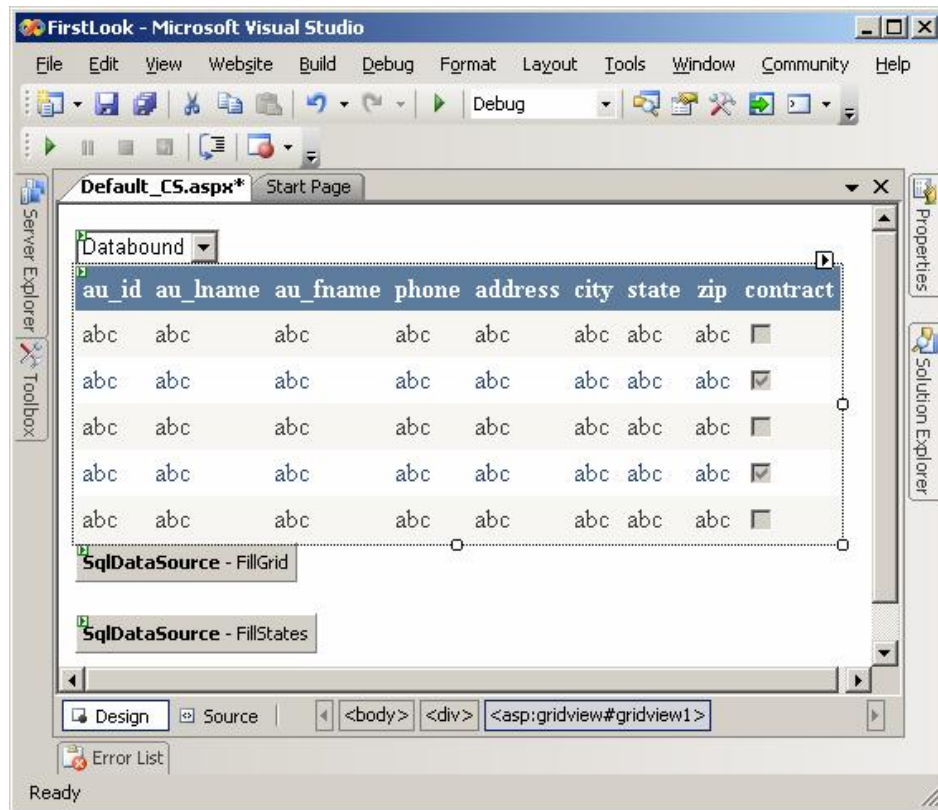
Test Query

SELECT statement:

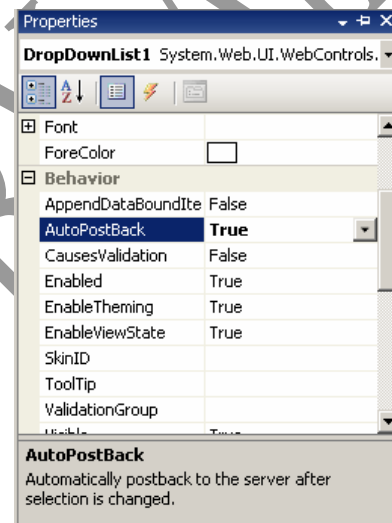
SELECT * FROM [authors] WHERE ([state] = @state)

< Previous Next > Finish Cancel

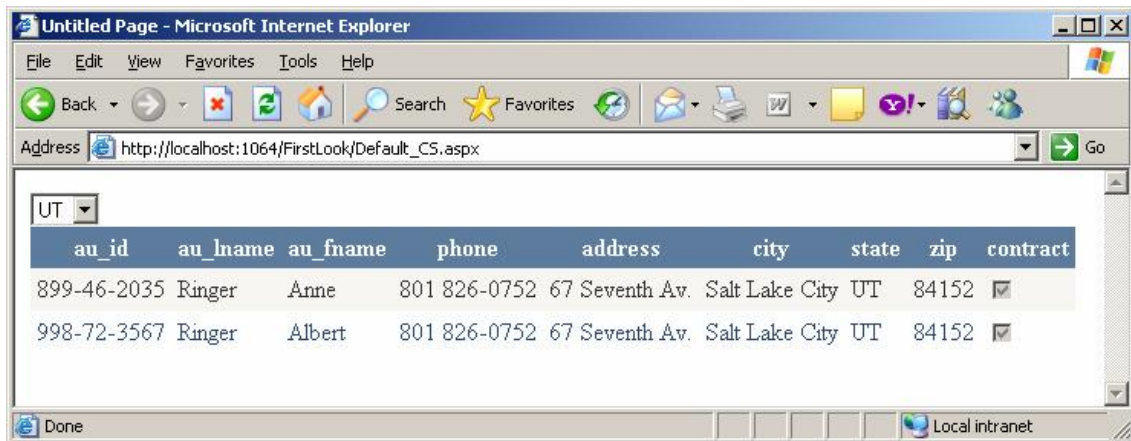
الشكل ٢٧] تتضح فيه نتيجة الاختبار - الحمد لله تمت بنجاح - [



الشكل ٢٨ [هذا يوضح شكل الصفحة بعد تعديل الشكل (Auto Format)]

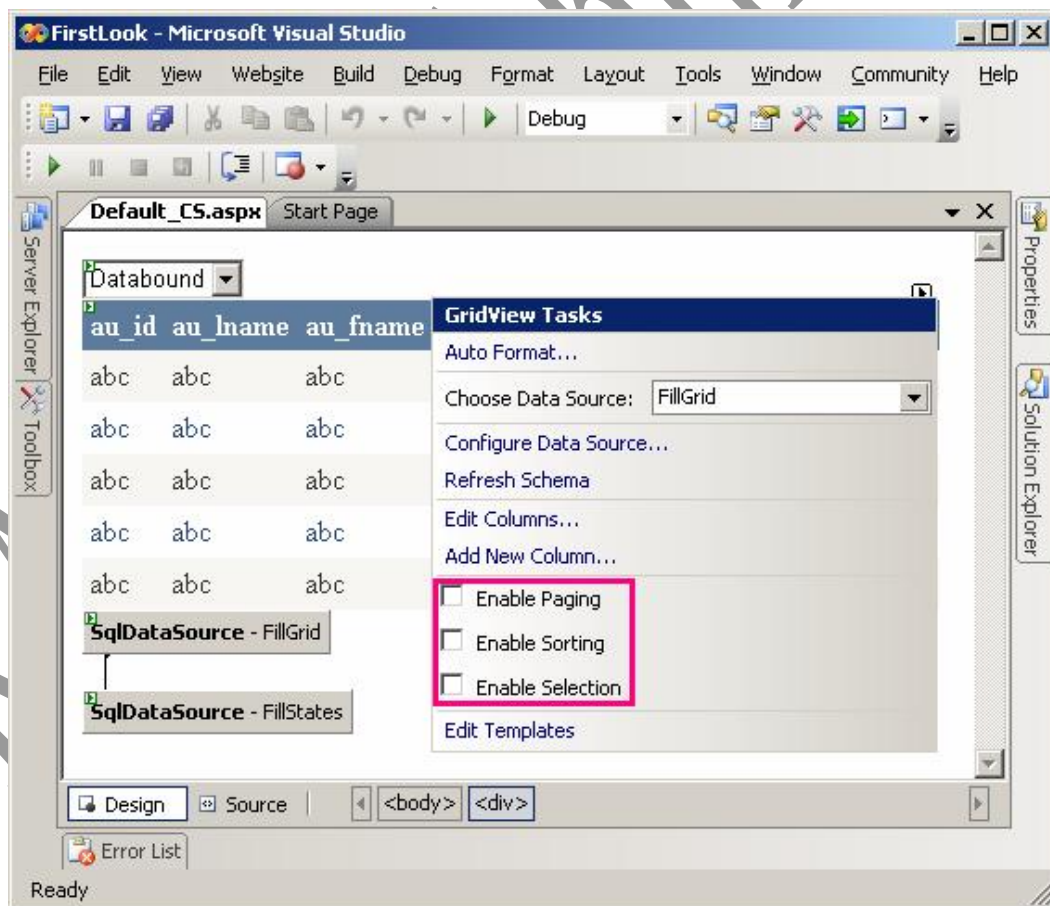


الشكل ٢٩ [هو لتفعيل الخاصية (AutoPostBack) حيث أن هذا الحدث هو الذي سوف يقوم بتفعيل المثال كله]

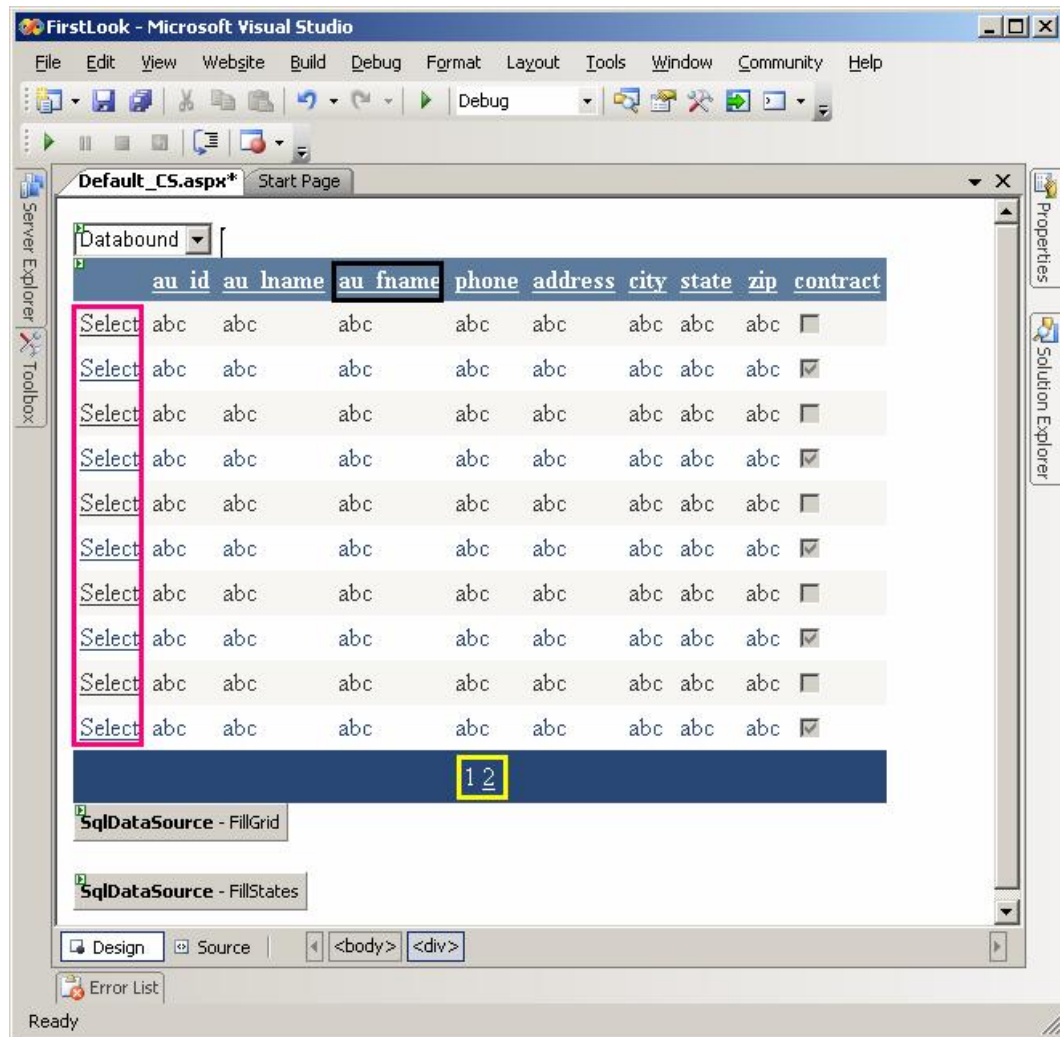


الشكل ٣٠ تظهر هذه النتيجة عند تغيير القيمة في الـ (DropDownList) حيث يقوم بعرض البيانات التابعة لها

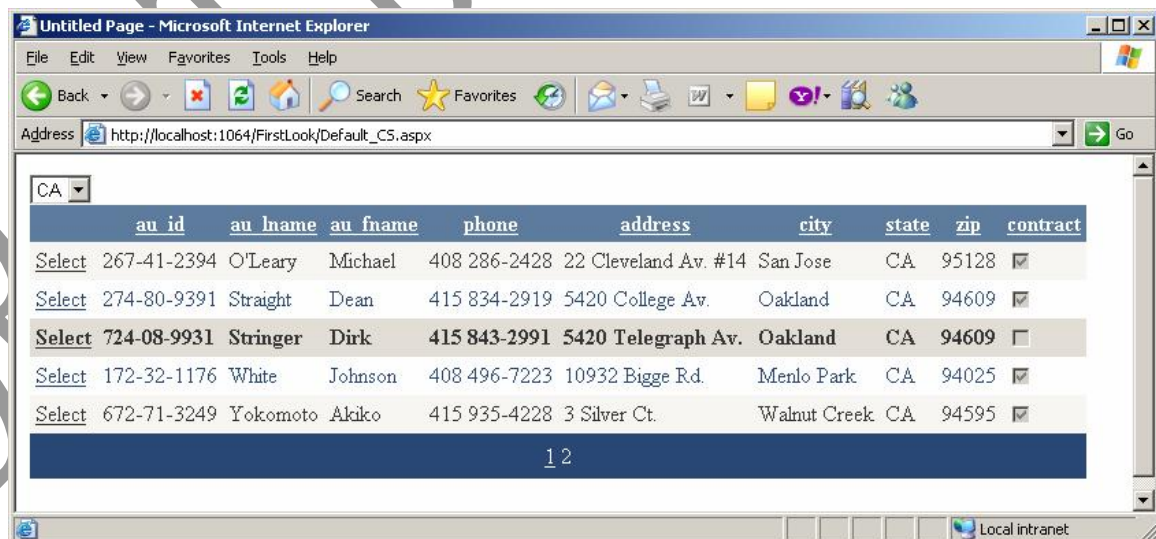
- ماذا لو أردنا أن نقسم البيانات لأن كميتها كبيرة جدا !!! و أيضا لو أردنا أن نعطي إمكانية للمستخدم في الترتيب و إختيار الصف الذي يريده !!! ؛ سنجد أن كل هذا موجود و بدون سطر كود واحد حيث ان كله يتم تشغيله ببساطة مع إصدار ٢٠٠٥ ، بينما في الإصدارات السابقة من فيجوال ستوديو كنا نضطر الى كتابة كود يقوم بهذه الخواص . و الآن لنكمل المثال.



الشكل 31 تلاحظ وجود ٣ إمكانيات يمكن تفعيلها وهما :- ١- التقسيم الى صفحات ، ٢- إعطاء المستخدم إمكانية ترتيب العناصر في الجدول ، ٣- إعطاء المستخدم إمكانية إختيار عنصر [



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



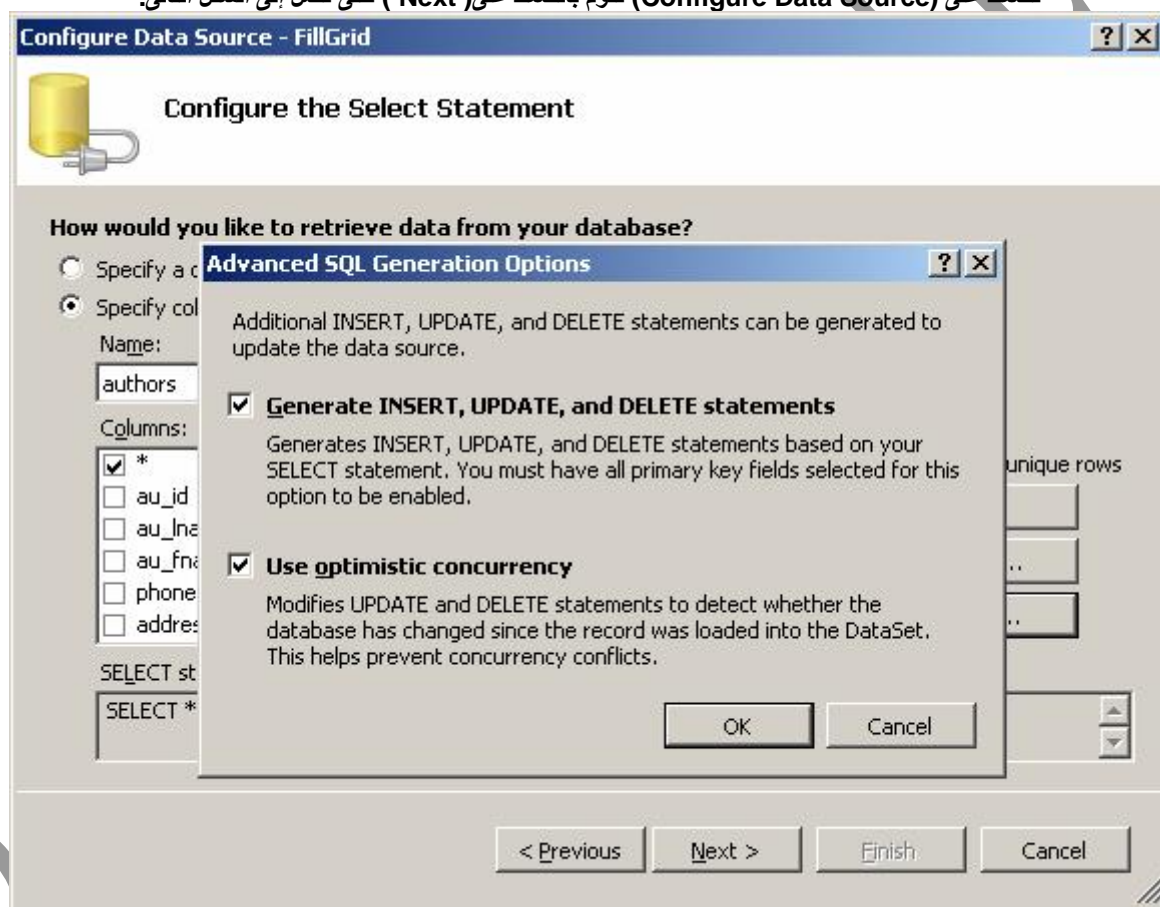
الشكل 33] الشكل النهائي للمثال بعد ضبط الخصائص و تفعيلها

- ولكن ماذا لو أردت أن أقوم بحذف أو تعديل البيانات كيف يتم ذلك!!! سيتضح كل هذا في تكملة المثال حيث سنقوم فيه بإضافة خاصيتي التعديل و الإضافة كما كنا نفعل في الإصدارات القديمة و لكن بدون سطر كود . بعكس ما كان يتم في الإصدارات السابقة كان هناك الكثير من الكود الذى يقوم بحل هذه المشكلة ويزيد هذا الكود كلما زادت عدد الأعمدة في حالة التعديل.

والآن نعود لنرى بقية المثال



الشكل ٣٤] نقوم بأظهار الـ (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 إلى الشكل التالي.



الشكل 35] الان نضغط على (Advanced) و سوف تظهر هذه النافذة لك و به إختياران علشك أن تقوم بتفعيلهم و هما كالتالى

الأختيار الاول (Generate Insert, Update and Delete statements)

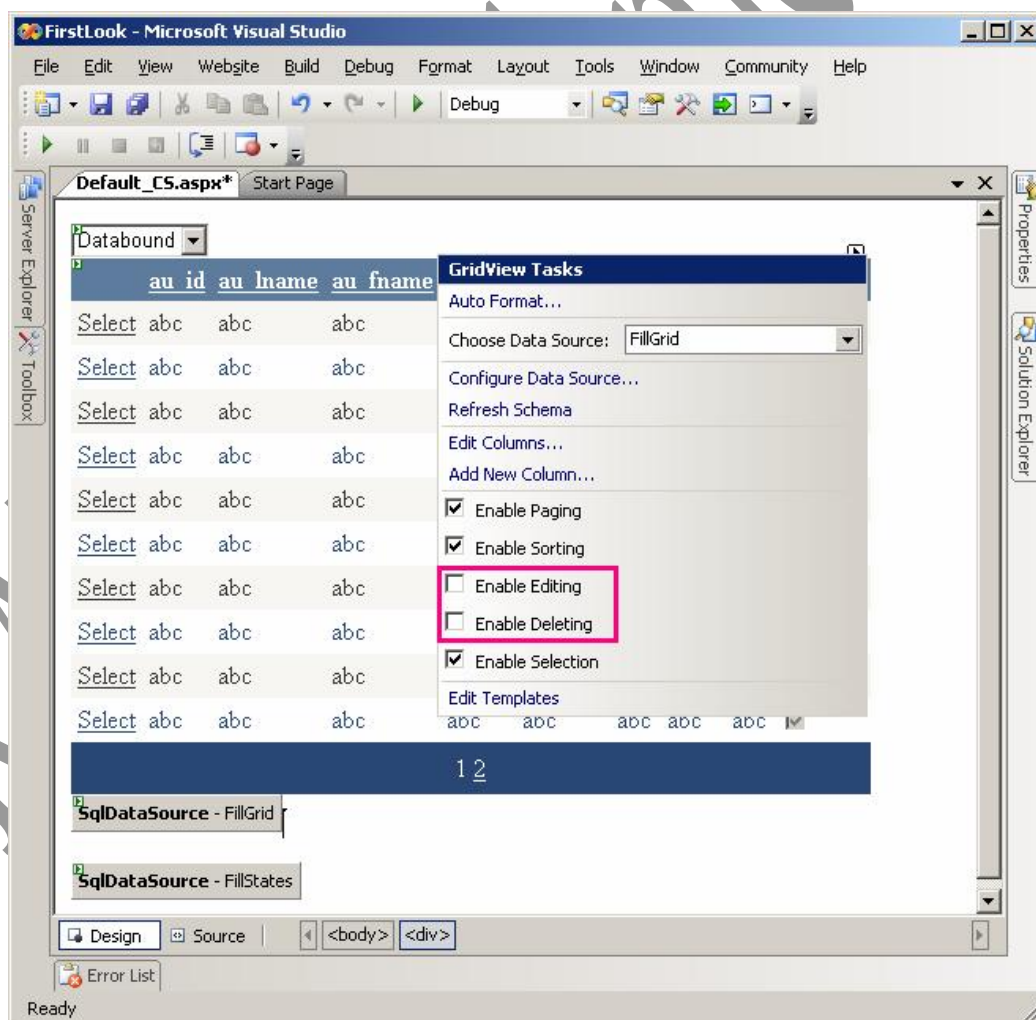
و هذا الإختيار يقوم عنك بإضافة خاصية الإضافة و التعديل و الحذف و هذا فى السابق كما ذكرت كان يتطلب عمل كثير و لكن الآن كل هذا سوف يتم عنك.

الأختيار الثانى (Use optimistic Concurrency)

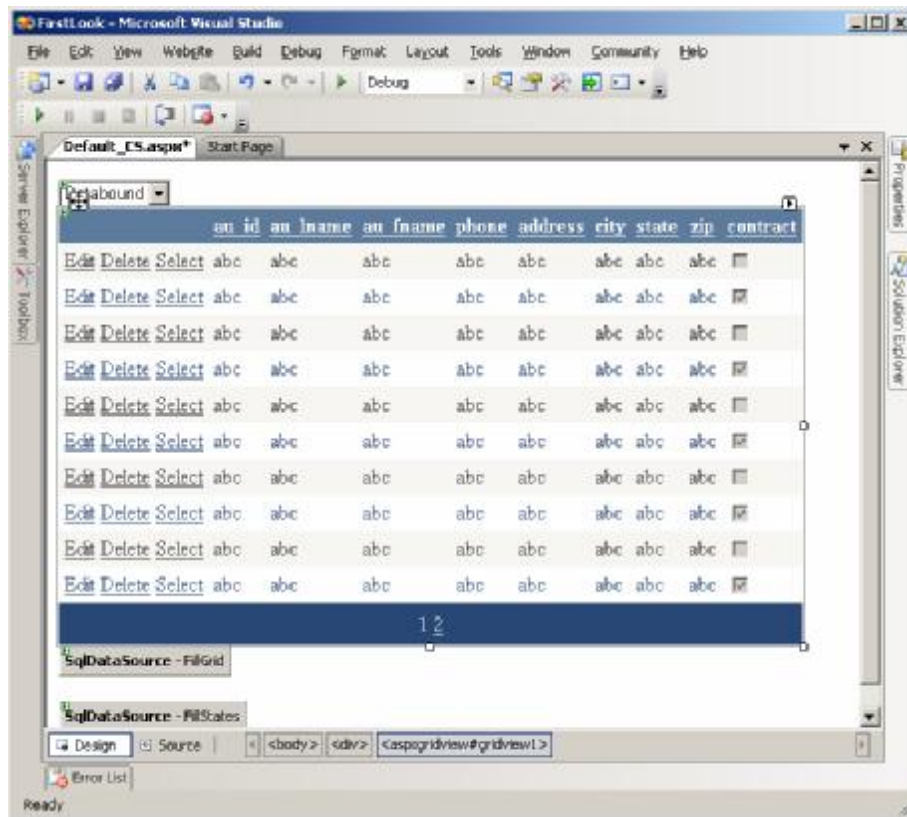
- ولابد من الاهتمام بعملية تأمين التعامل و تعديل البيانات ؛ لأن هناك العديد من المستخدمين لقاعدة البيانات .
- وسيتم الإستعانة بمثال واقعي و سهل الفهم – لشرح هذه الجزئية- ؛ فلنفرض أن هناك مستخدمين يقوموا بنفس الوقت بتعديل بيانات معينة و تصادف أن الأثنين يعدلان نفس البيانات ، و قام المستخدم الأول بحفظ البيانات ثم قام المستخدم الثاني بحفظ البيانات التي عدلها ، إذا فإن أخر تعديل هو التعديل الذي قام به المستخدم الثاني .
- وعندما حاول المستخدم الأول إستخدام البيانات وجد انها ليس كما حفظها ؛ هنا تكمن المشكلة وحلها يكون بوجود طريقة لضبط التعامل بين المستخدمين - بعضهم البعض - بمعنى أنه عندما يقوم مستخدم بتعديل البيانات ثم حاول حفظها ، يجب أن يكون هناك خطوة قبل الحفظ و هي التأكد من أن البيانات المخزنة لم تتغير و هذا ما يسمى بالـ (Optimistic Concurrency)

ففى الإصدارات السابقة كان المطور يقوم بعمل هذا الموضوع بمساعدة الـ (Dataset) حيث كان هناك ما يسمى بالـ (Data version) و هى تحتوى على البيانات بكل إصداراتها – ١- البيانات الأصلية " البيانات عند عرضها قبل التعديل " و ٢- البيانات الحالية " فى قاعد البيانات " – كما كان يقوم المطور بمقارنة البيانات للتأكد من أن البيانات لم تتغير فيقوم بالحفظ ؛ أما فى حالة إن تم تغيير البيانات فكان يقوم –المطور- بعرض رسالة تحذيرية للمستخدم .

الآن تقوم هذه الأداة الرائعة بهذا العمل عنك و بالتالى توفر الكثير من الوقت على المطور .



الشكل ٣٦ [بعد أن تم تفعيل الخواص فى الـ (DataSource) ؛ قم بإظهار الـ (Smart Tag) الخاص بالـ (GridView) و سوف تلاحظ ظهور خاصيتين قم بتفعيلهم و هما كما ترى (Enable Editing) & (Enable Deleting)]



الشكل 37 [تظهر كلاً من الوصلتين (Edit) & (Delete) ، و سوف يقوم كلاً منهم بمهمة التعديل و الحذف]

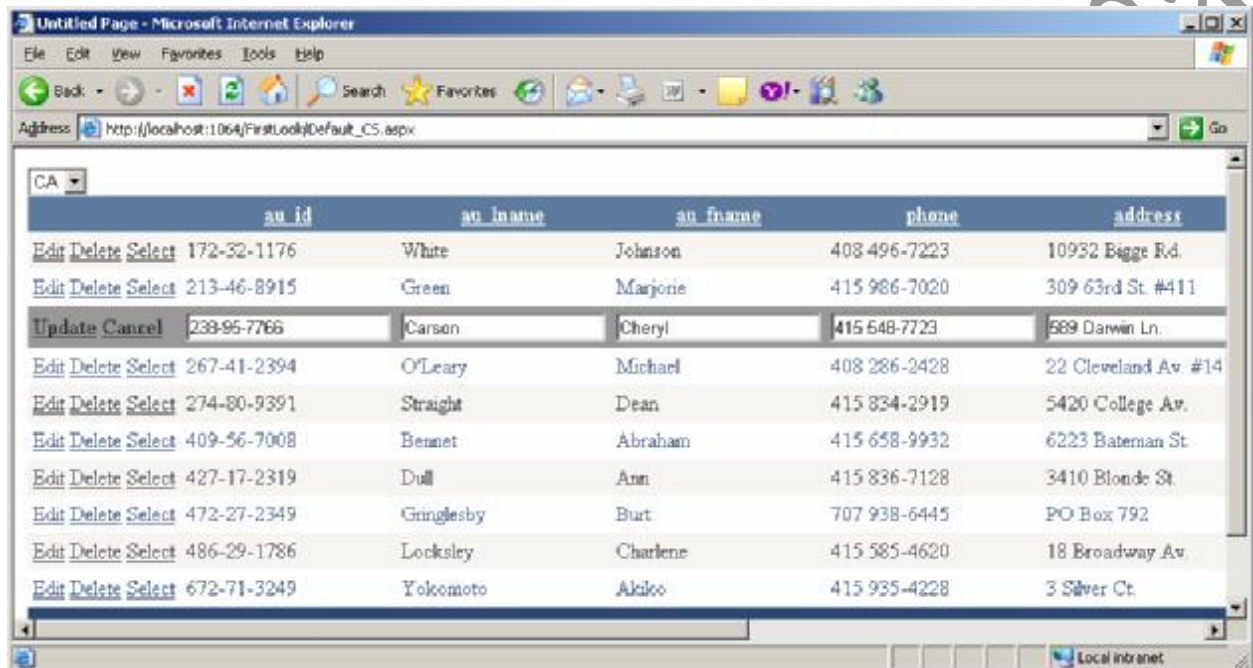
```
<asp:SqlDataSource ID="FillGrid" runat="server" ConnectionString="<%$
ConnectionString:pubsConnectionString %>"
    SelectCommand="SELECT * FROM [authors] WHERE ([state] = @state)"
    ConflictDetection="CompareAllValues"
    DeleteCommand="DELETE FROM [authors] WHERE [au_id] = @original_au_id AND
        [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 كما نرى به كل تفاصيل عملية التعديل و الحذف من بداية جملة الإستعلام (SQL) إلى المدخلات التي سيتم حفظها.



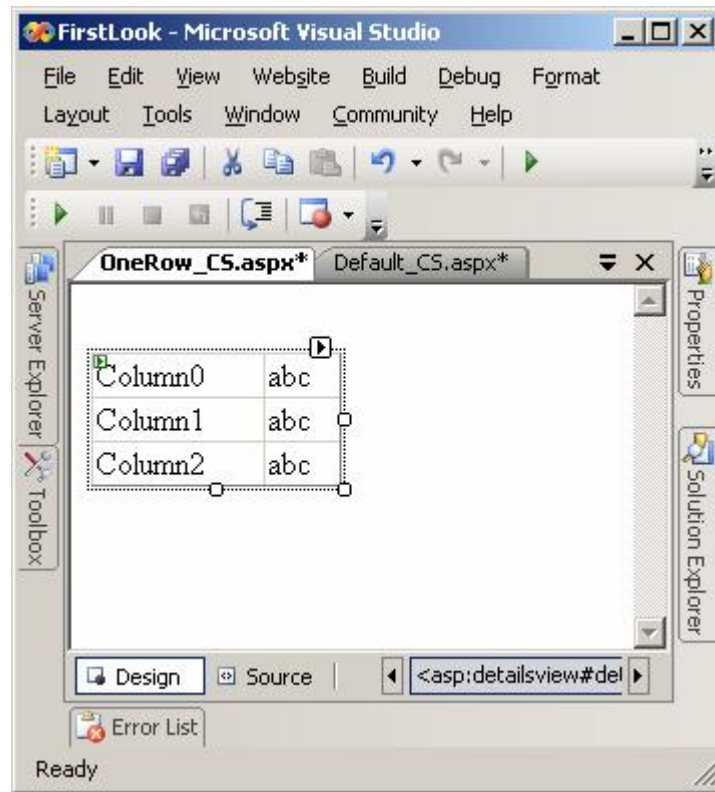
الشكل 38 [الشكل النهائي في إحدى محاولات للتعديل]

أداة Details View

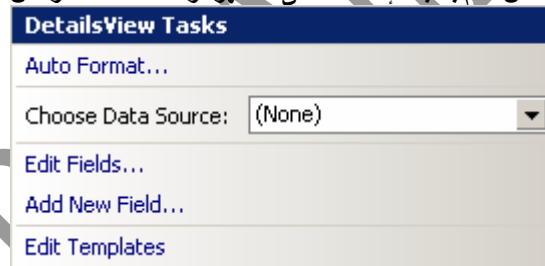
في بعض الأحيان -على ما أظن- كان بعضنا لا يتعامل مع الـ (Data Grid) ؛ و إنما كان يعرض البيانات بشكل مختلف عن ذلك ؛ حيث كان يتطلب الأمر- في بعض الأحيان- عرض صف واحد ، و كان هذا في بعض الأحيان يعتبر معاناة ؛ حيث أننا لو فرضنا أنك تود عرض ٧٠ بيان من قاعدة البيانات سيحتاج هذا للكثير من الكود لعمل هذا الموضوع -في الإصدارات السابقة- و لكن الآن مع الأداة (Details View) تم حل هذه المشكلة . و لنرى كيف نتعامل مع هذه الأداة و لنجرب مثال يساعدنا على فهم الفكرة بشكل أكبر.



الشكل 39 [نستعمل الأداة (Details View) من صندوق الأدوات و هي أداة -كما سنرى- مخصصة للعرض الفردي أو كما يسمونها للعرض التفصيلي]



الشكل 40 [يعرض شكل الأداة ، و الآن لنقم بضبط الإعدادات التي ستظهر قوة هذه الأداة ، و الآن لنضغط على الـ (Smart Tag)]



الشكل 41 [سوف نختار (Choose Data Source)]

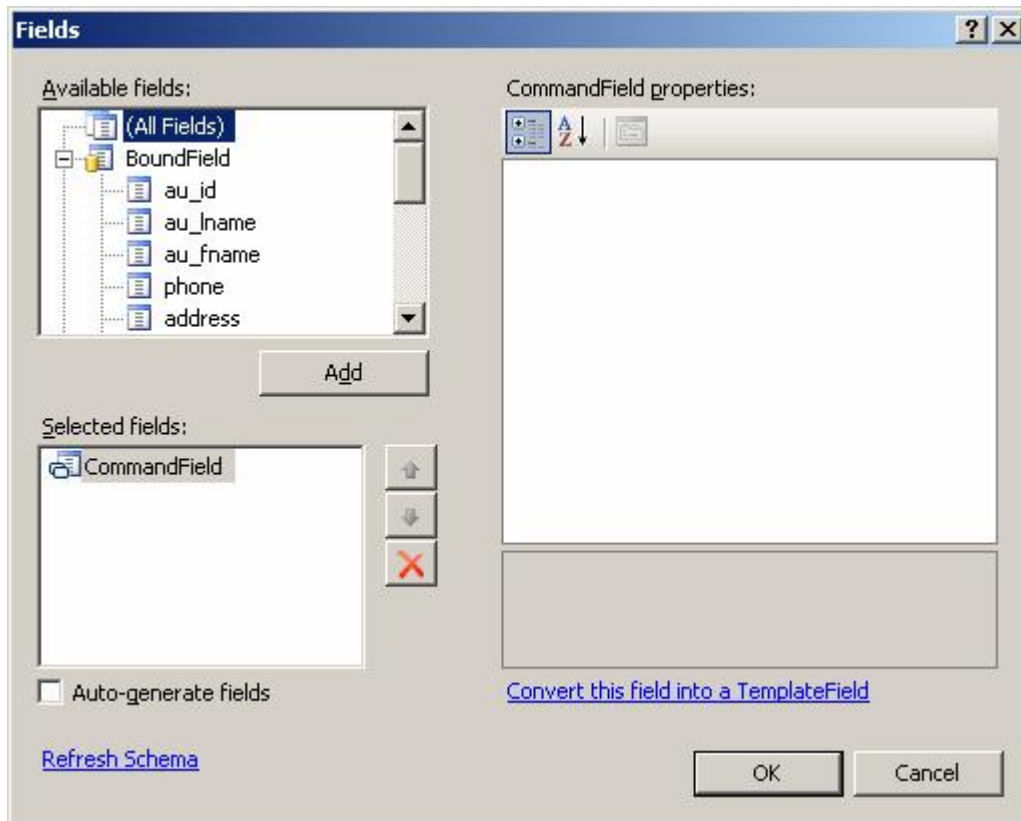


الشكل 42 [نختار (New Data source)]

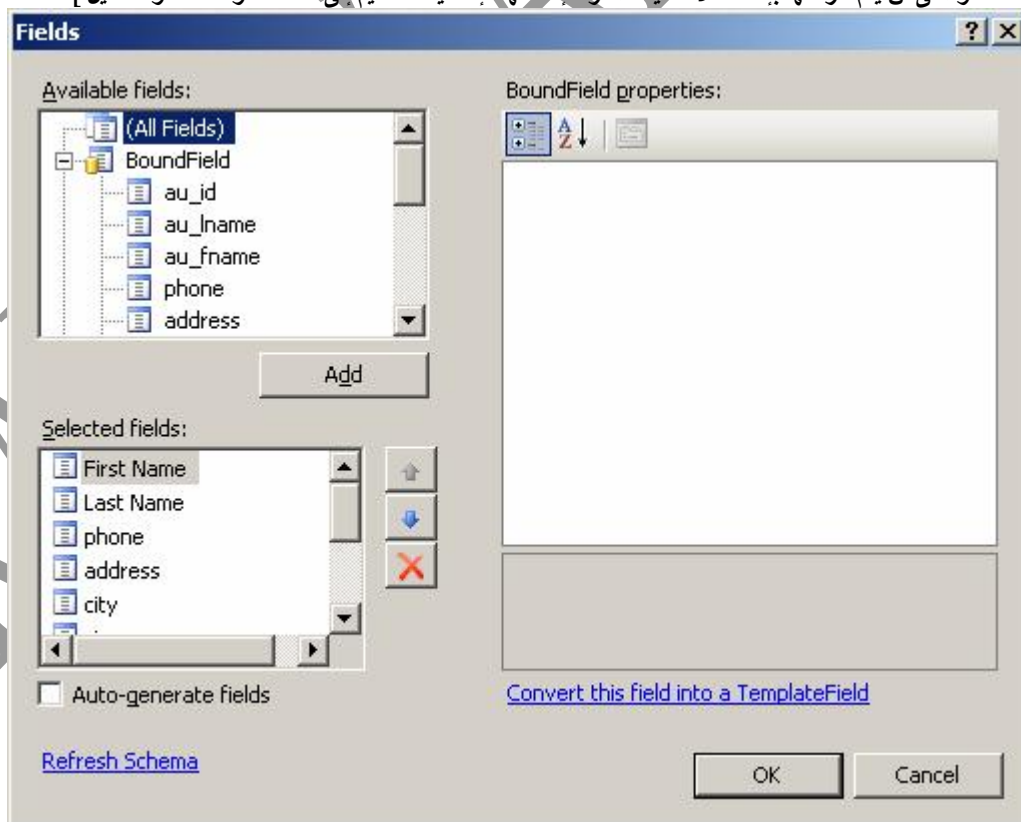
الشكل 43 [لن أذكر طريقة إعداد الـ (Datasource) حيث تم ذكرها فيما سبق، و الآن نختار جدول (Authors) ، ثم نضغط على زر (Advanced) ؛ سوف تظهر هذه النافذة فنختار منها مصدر فلترة البيانات - بمعنى آخر نختار مصدر الشرط (Where) في جملة الإستعلام - و كما نرى أنه من الممكن أن يكون مصدر الشرط أحد هذه الأنواع الموجودة في القائمة المنزلة (Source) ، بعد ذلك قم بتحديد أسم الـ (Query String) التي سوف تحمل قيمة الشرط]

الشكل 44] وفيه قمنا بتحديد حقل مصدر البيانات بعد ذلك نقم بالضغط على (Add) ، ثم بعد ذلك (OK) ، و نكمل بقية خطوات معالجة مصدر البيانات (Data source wizard) [

الشكل ٤٥] مرة أخرى نظهر الـ (Smart Tag) الخاص بالـ (Detail view) ، و نقوم بتفعيل كل الخيارات- كما نرى- و لكن لابد و أن نكون حذرين لان الخاصيتين الخاصتين بالتعديل و الحذف لابد و أن تكون مفعلة في الـ (Data Source) كما أوضحنا هذا من قبل [



الشكل 46 [من جزء الـ (Smart Tag) الخاص بالـ (Detail view) ، نغم باختيار الـ (Edit Fields) و هنا سوف نختار البيانات المراد عرضها و التي لن يتم عرضها بإضافة الأماكن المراد إضافتها كإمكانية التقسيم إلى صفحات و الحذف و التعديل]



الشكل 47 [بعد إختيار البيانات المراد عرضها ؛ و ذلك عن طريق إختيار الحقول مراد عرضه ثم (Add) بعد الإنتهاء من هذه العملية كل ما عليك هو أن تضغط على (OK)]

First Name	abc
Last Name	abc
phone	abc
address	abc
city	abc
zip	abc
contract	<input type="checkbox"/>
Edit	
Delete	
New	
1 2	

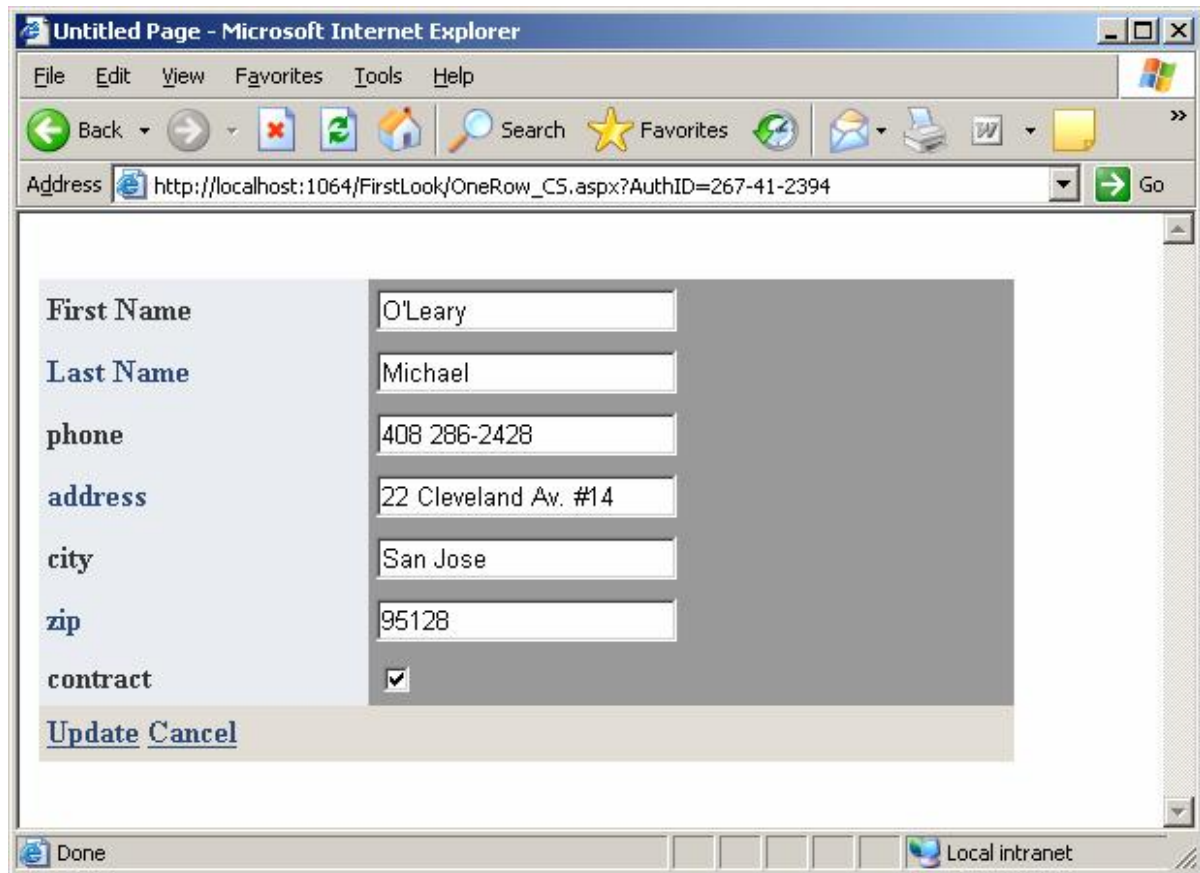
SqlDataSource - OneRow

الشكل 48 [يوضح شكل الأداة بعد الإنتهاء من إعدادها و طبعا بعد تعديل الـ (Auto Format) الخاص بالأداة]

Microsoft Internet Explorer window showing the web form. The address bar contains: 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	<input checked="" type="checkbox"/>
Edit	
Delete	
New	

الشكل 49 [و هو الشكل النهائي أثناء عمل الأداة ، و كما نلاحظ في شريط العنوان (Address bar) أن عنوان الصفحة يحتوي على (Query string) ، و هي كما قمنا بإعدادها 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

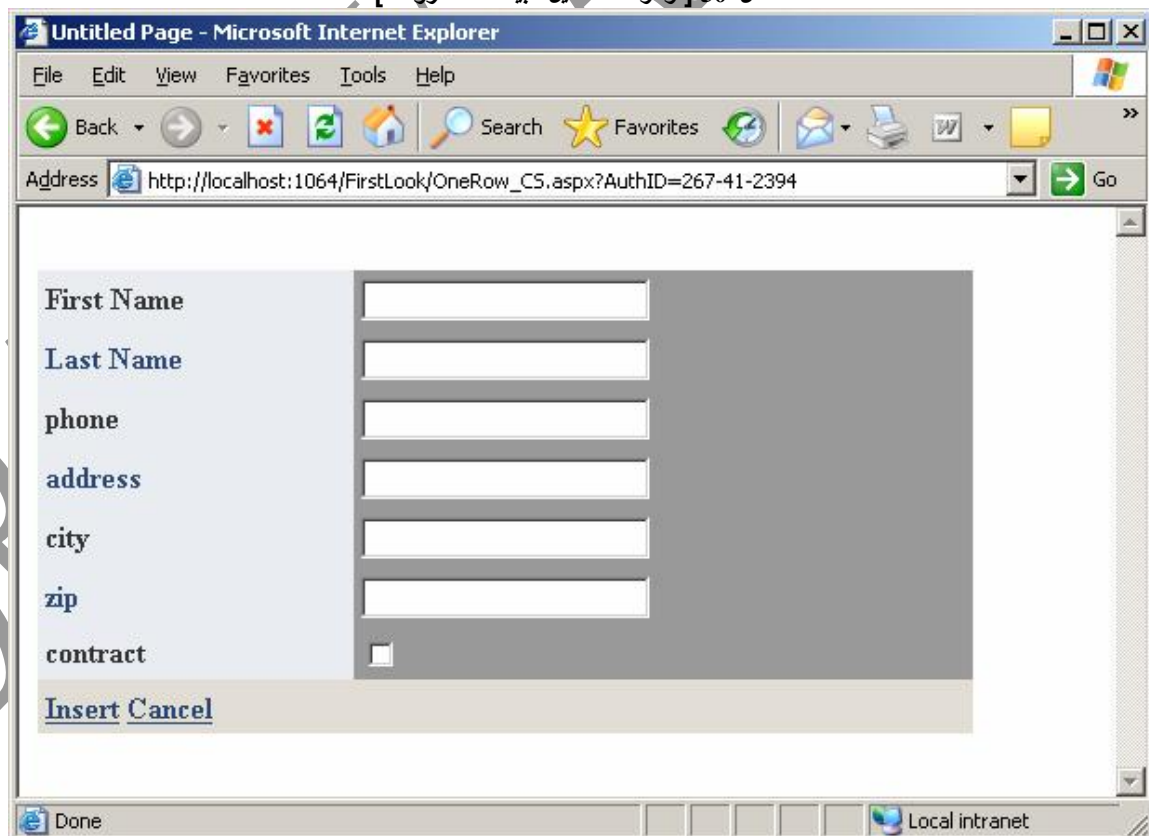
Address http://localhost:1064/FirstLook/OneRow_CS.aspx?AuthID=267-41-2394 Go

First Name	O'Leary
Last Name	Michael
phone	408 286-2428
address	22 Cleveland Av. #14
city	San Jose
zip	95128
contract	<input checked="" type="checkbox"/>

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

Done Local intranet

الشكل 50 [وهو 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 Go

First Name	
Last Name	
phone	
address	
city	
zip	
contract	<input type="checkbox"/>

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

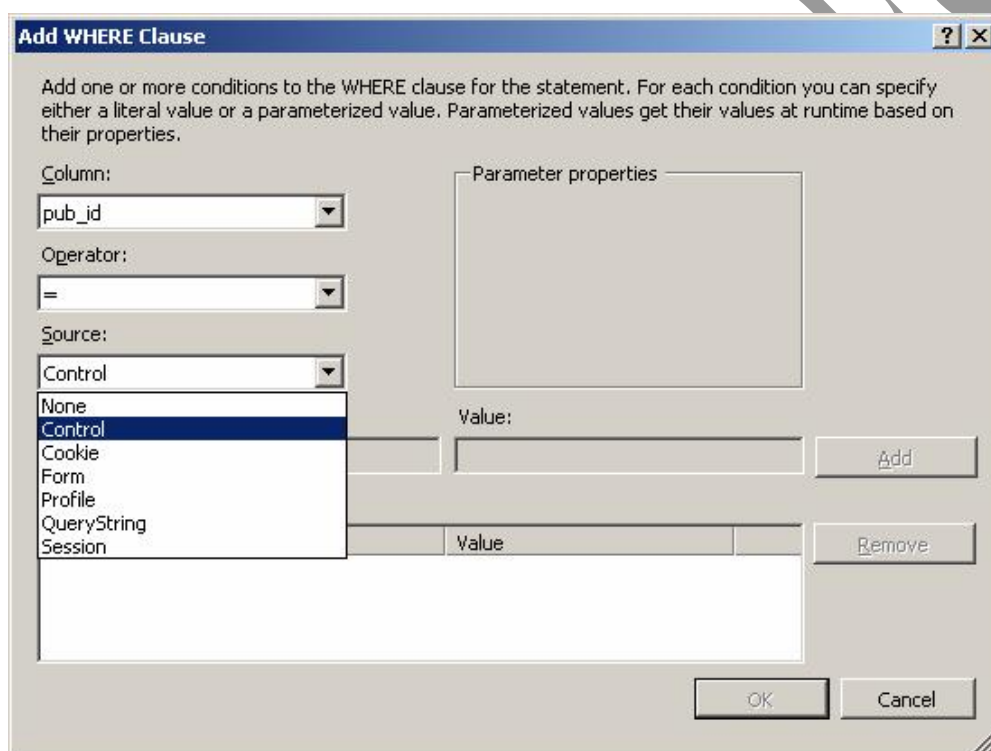
Done Local intranet

الشكل 51 [و هذا عند طلب إضافة بيان جديد]

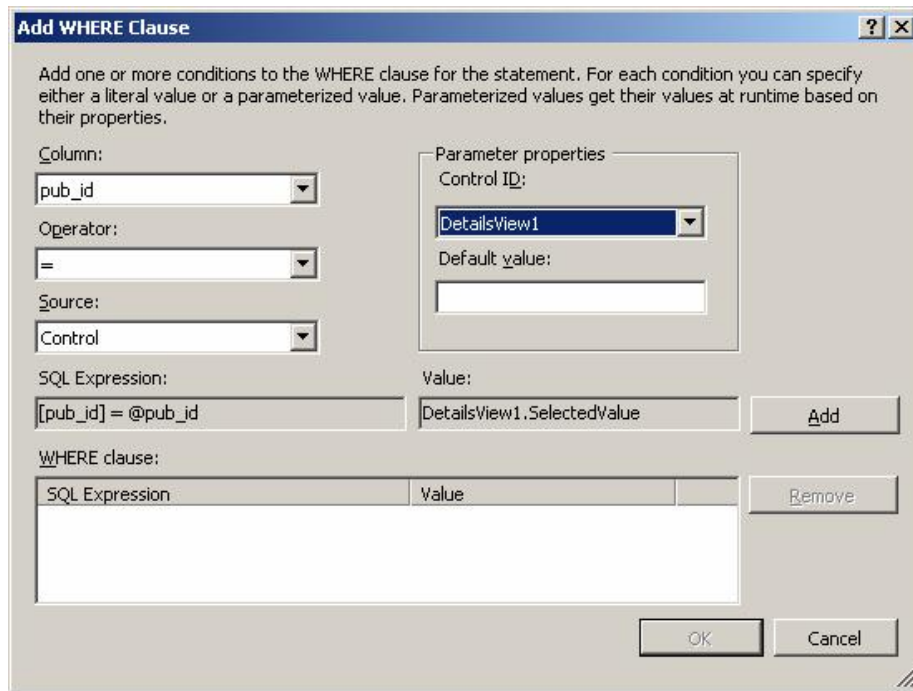
- ماذا لو أردنا أن نقوم بعمل نموذج (Master & Details) - هذا النموذج كلنا نعرفه- و أبسط مثال له هو أى فاتورة ؛ حيث يوجد بيانات المشتري فى بداية الفاتورة ثم يأتي بعد ذلك جدول بالأصناف، وسيتم إيضاح هذه المعلومة بشكل أكبر بالمثال التالى .

- والنموذج:- عبارة عن بيانات الناشرين و فى النصف الآخر من الصفحة بيانات الكتب التى نشرها الناشر، سوف نحتاج إلى وضع (Grid view) و (Detail View) ؛ وسوف تعرض محتويات الجدول (Publisher) فى الـ (Detail View) ، كما سيتم عرض الـ (Titles) فى الـ (Grid View) و لكن فى الجزء الخاص بالـ (Where) أو بمعنى آخر جملة الإستعلام.

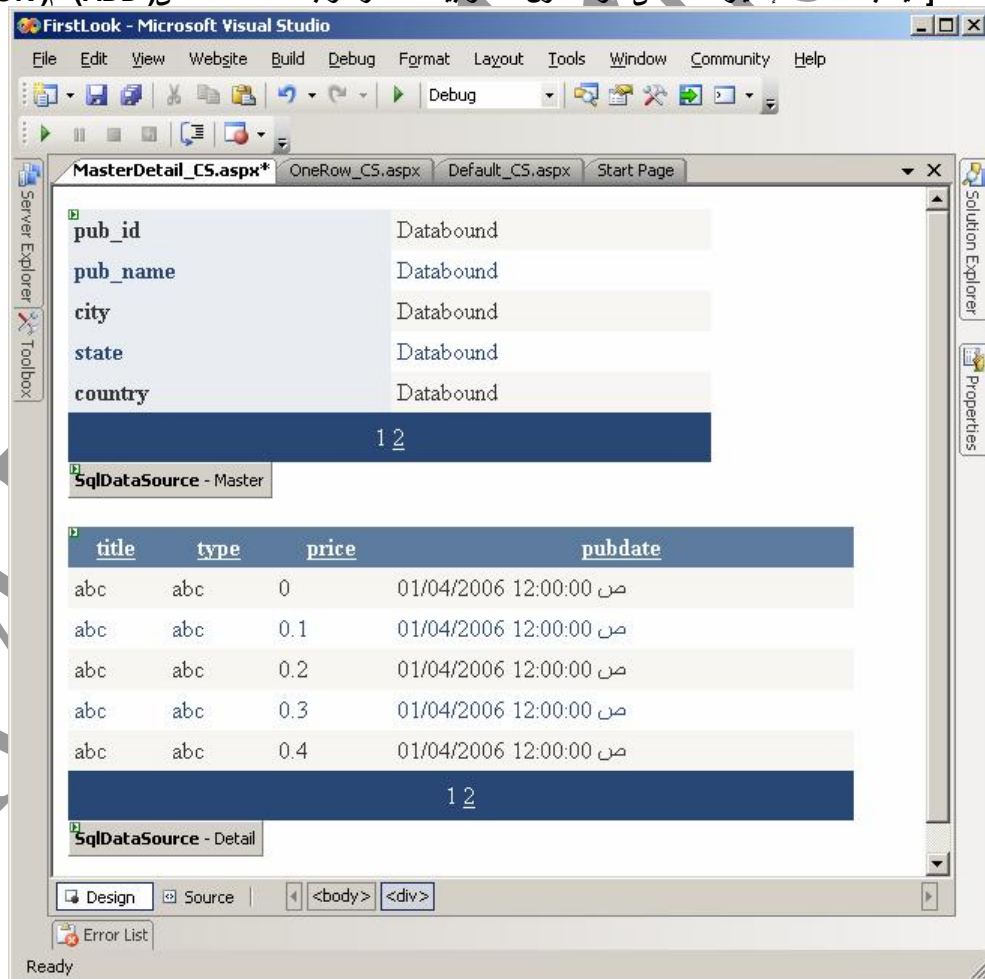
وسنقوم الخطوات التالية -الموضحة فى الرسومات- لتوضيح هذا الأمر .



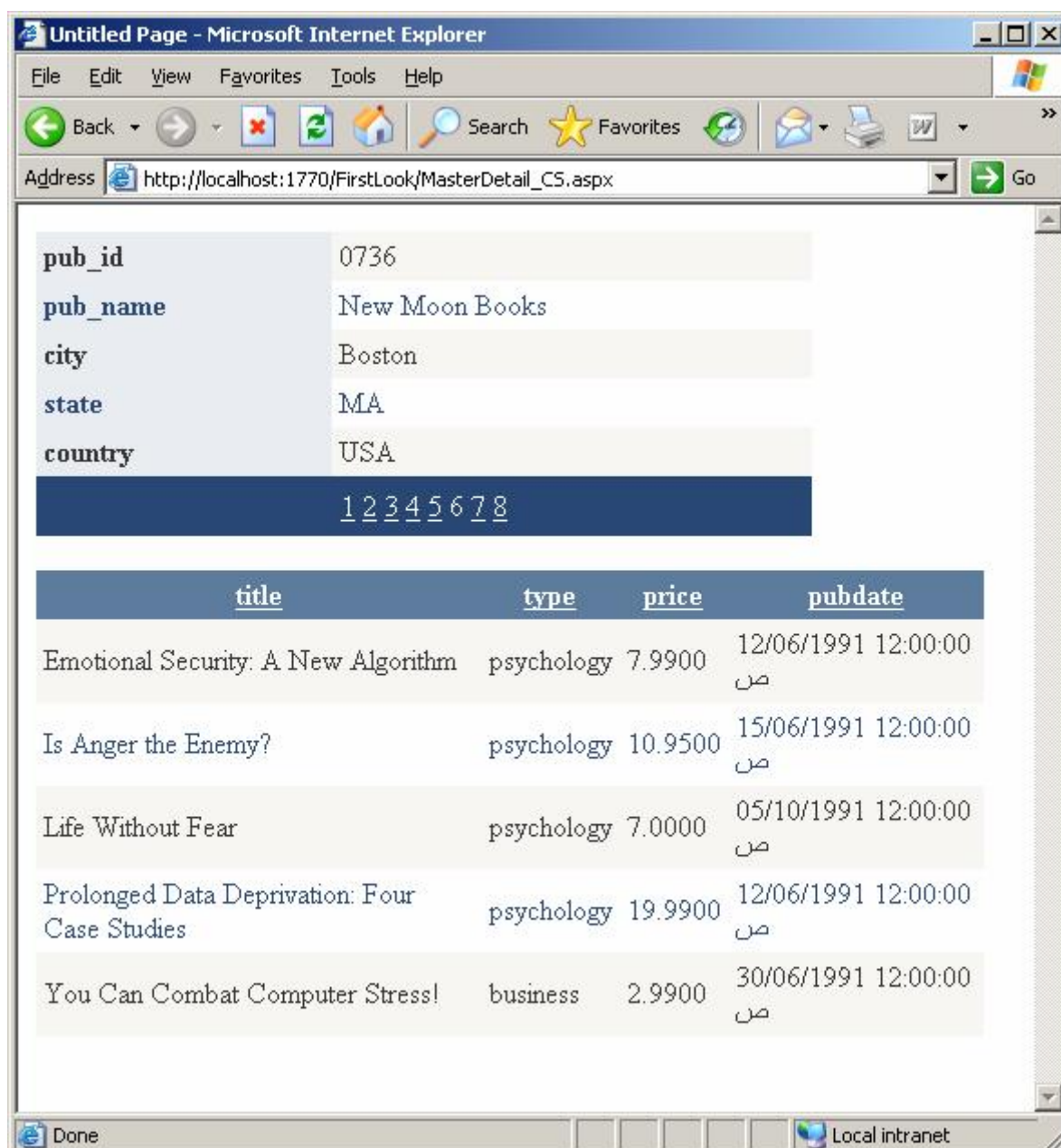
الشكل 52 [نختار الحقل الذي سوف يتم عليه تطبيق الشرط (Column) ، و بعد ذلك نختار نوع الشرط (Operator) ، و بعد ذلك مصدر الشرط و سوف نختار من مصدر الشرط النوع (Control)]



الشكل ٥٣ [يطلب منك هنا إختيار الأداة التي سوف تكون مصدر بيانات الشرط و بعد ذلك أضغط على (ADD) ثم (OK)]



الشكل ٥٤ [هو الشكل النهائي للمثال كما نرى في الجزء العلوى (Detail view) و في الجزء السفلى (Grid View)]



الشكل 55 [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 مواضيع أخرى لم نتطرق إليها على سبيل المثال عرض البيانات المخبأة لتخفيف الضغط على قاعدة البيانات و هذا ما يسمى بالـ (Caching) .

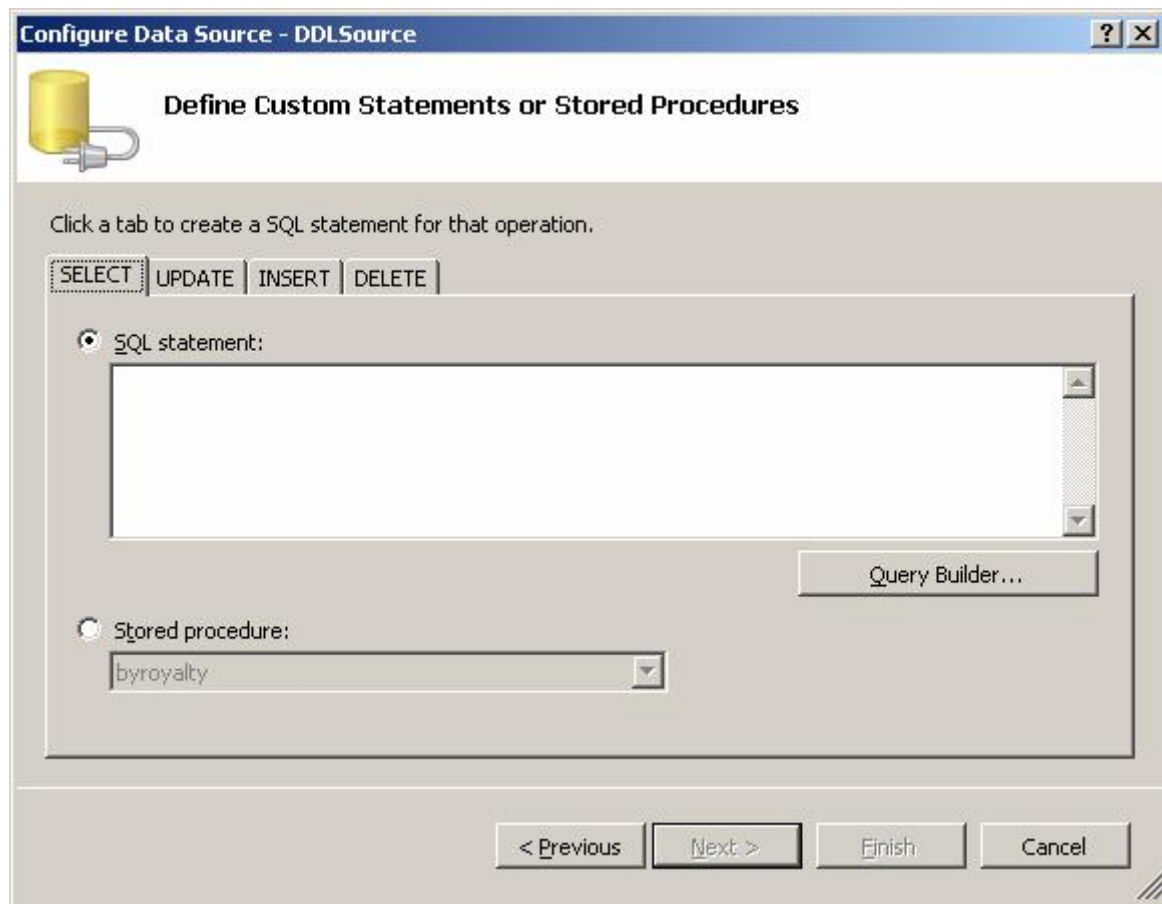
عرض البيانات المخبأة لتخفيف الضغط على قاعدة البيانات (Caching)

و الآن لنستعرض معاً كيفية تفعيل هذه الخاصية و التي قد توفر الكثير من الوقت و الأداء الذي سوف يحسب لصالح المطور .

فمثلاً عملية الإتصال بقاعدة البيانات و إحضار البيانات منها عملية تحتاج الى وقت ، فعلى سبيل المثال أنك لو أحضرت ١٠٠ صف من البيانات ؛ سوف تستهلك بعض الوقت في التالي :-

- ١- إجراء إتصال للتأكد من هوية المتصل بقاعدة البيانات وذلك لضمان حماية البيانات
- ٢- بعد ذلك تطبيق جملة الاستعلام ثم الرجوع بالنتيجة- هذه العملية فى حد ذاته تستهلك الكثير من الوقت- ومن هنا ظهرت الحاجة لما يسمى بتخبة البيانات (Data Caching) وهذه العملية نستعملها فى بعض الأحيان لانها توفر سرعة كبيرة جدا ؛ بالإضافة إلى أنها تقلل عدد مرات الإتصال بقاعدة البيانات ، كما أنها عادة لا تتم إلا على البيانات التى لا تتغير باستمرار .
- فعلى سبيل المثال قوائم المنتجات، نادرا ما تتغير ؛ ففى مثل هذه الحالة نقوم بتخبة البيانات الخاصة بالقائمة و عند الحاجة إليها نقوم بعرضها بدل من الإتصال بقاعدة البيانات و أستهلاك هذا الوقت، و لكن كل هذا الكلام نظرى و نحن كما تعودنا على أنه دائمة هناك تطبيق عملى على ما نقوم به.
- و الآن لنرى مثال يوضح ما نقصده بعملية تخبة البيانات . و الآن سوف نقوم بإعداد مثال بسيط وهو (Drop Down List) بها أسماء المؤلفين ، و عند الإختيار منها تظهر البيانات تفصيلية فى الـ (Grid View) و سوف تظهر مع بيانات حقل سوف نضيفه وهو حقل التاريخ ، و سوف نرى ما هو المقصود بشكل أوضح مع المثال.
- كما يوضح كلاً من (Drop Down List) و الـ (Detail View) ، و الآن لنقم بإعدادهم و هذه المرة سوف نجد طريقة جديدة للأعداد .

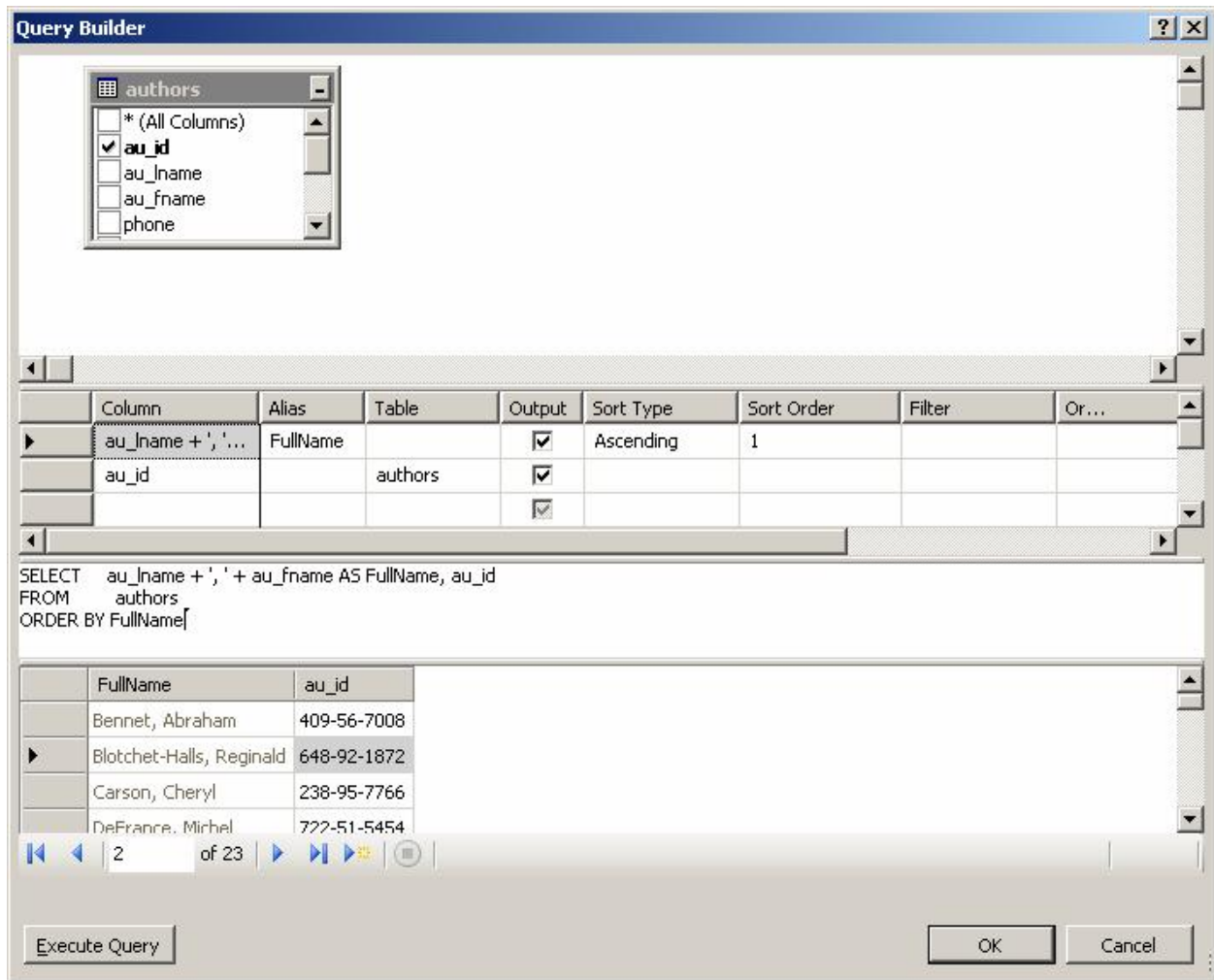
الشكل ٥٦ [بعد إعداد جملة الإتصال سوف يظهر لك هذا الشكل المعتاد لإختيار مصدر البيانات هذه المرة لن نأخذ من الجدول مباشرة و لكن سوف نقوم بعمل الإستعلام بأنفسنا، فم بإختيار (Specify a custom SQL statement or stored procedure)]



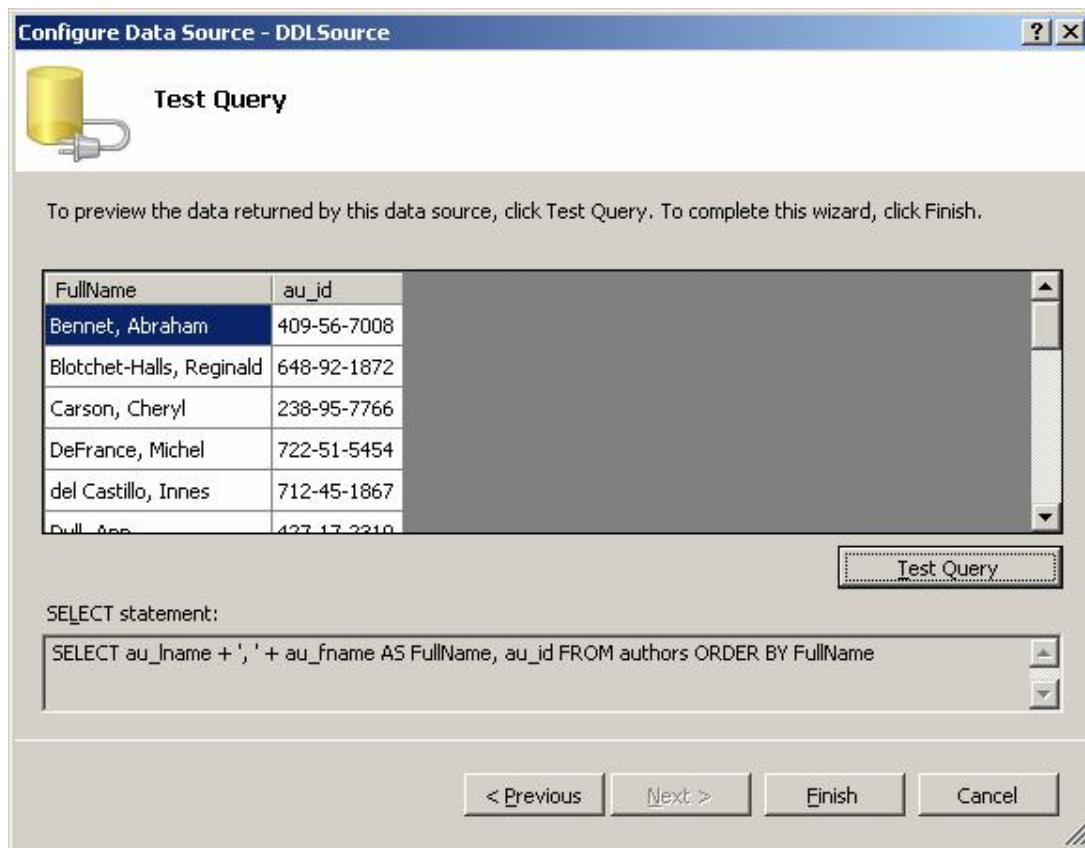
الشكل 57 [سيظهر لك هذا الشكل قم بالضغط على (Query Builder)]



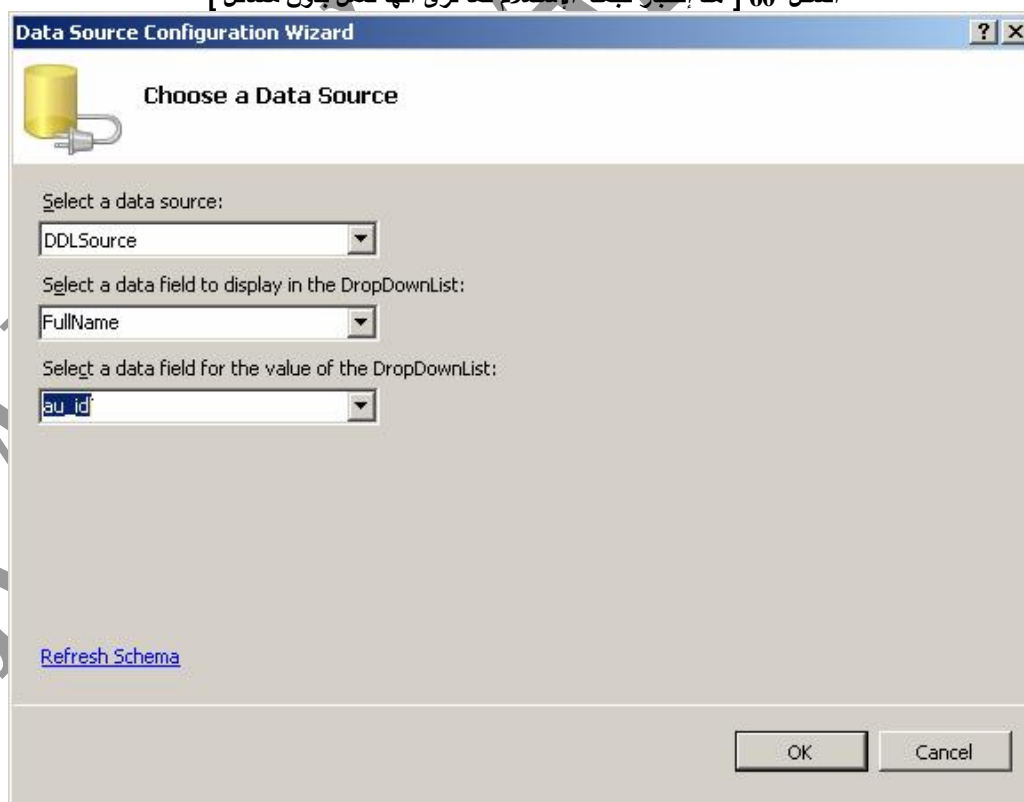
الشكل 58 [منها قم باختيار الجداول المشاركة في جملة الإستعلام و في حالتنا سوف نختار (authors) ، و بعد ذلك نضغط على (Add) ثم (close)]



الشكل 59] منها قم بإختيار البيانات التي سيتم عرضها ، كما ترى قم بإختيار 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 [هنا اختبار لجملة الاستعلام كما نرى أنها تعمل بدون مشاكل]



الشكل ٦١ [نقوم باختيار الحقل الذي سوف يظهر و الحقل الذي سوف يكون قيمة للعنصر في الـ (Drop Down List)]
بعد ذلك نقوم بإعداد 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 [يوضح جملة الاستعلام للـ (Detail View) ونلاحظ وجود حقل أسمه (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 [يتم إختيار مصدر بيانات الشرط، كما نرى و أعتدنا 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:

Control:

ControlID:

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 64 [إختار مصدر الشرط (Control) و سوف نختار العنصر الذى سوف يكون مصدر البيانات و هو فى هذه الحالة (Drop Down List) أو كما هو مسمى فى المثال (DropDownList1) ، نلاحظ أن هناك وصلة بلون أزرق (Show Advanced Properties) هذه الوصلة سوف تظهر بعض الخصائص المهمة فى عملية الربط بين الـ (Drop Down list) و الـ (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 ...

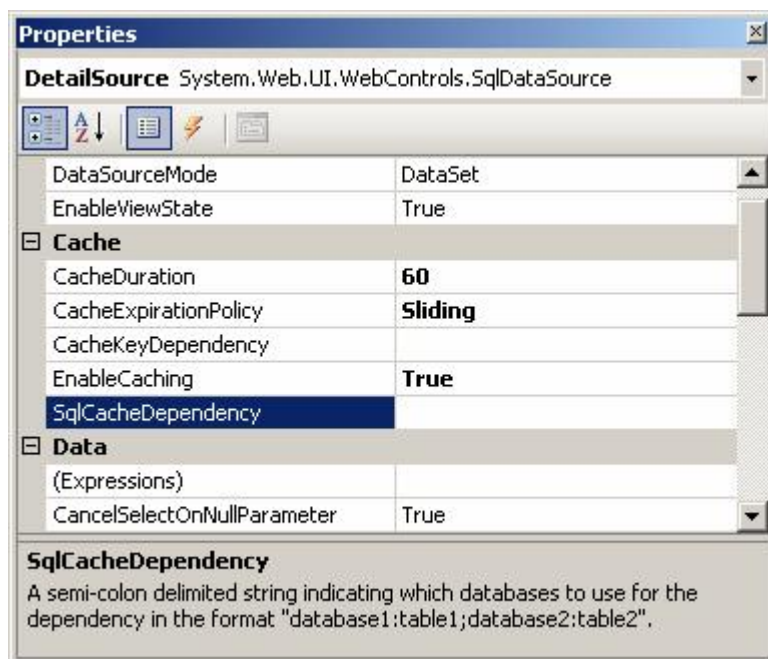
[Hide advanced properties](#)

SELECT statement:

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

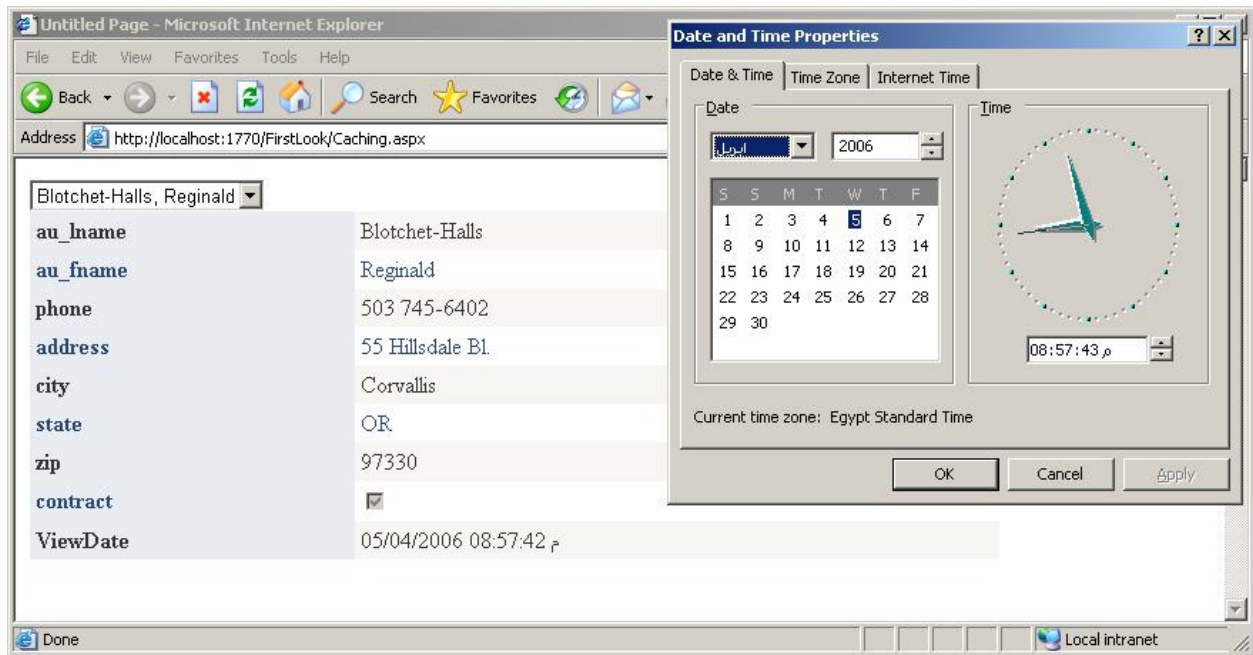
< Previous Next > Finish Cancel

الشكل 65] نلاحظ - كما ذكرنا من قبل- أنه ظهر مكان الوصلة لوحة خصائص مصفوفة للأداة (Drop Down List) و التي منها نقم بضبط الإعدادات المهم على سبيل المثال في حالة أن العنصر كانت قيمة فارغة يقوم بتحويلها إلى (Null) ، كما نلاحظ أن هناك وصلة (Hide Advanced properties) و هي تقوم بإخفاء لوحة الخصائص، و الآن لنقم بالضغط على (Next) حتى نصل إلى نهاية المعالج]



الشكل 66 [و أهم ما في الموضوع خصائص الـ (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

- ١ - الخاصية (CacheDuration) : - و المقصود بهذه الخاصية هو مدة تخبئة البيانات و التي بعدها سوف يقوم مصدر البيانات بإحضار البيانات من جديد، و يتم هنا ضبط الوقت بالثانية.
 - ٢ - الخاصية (EnableCaching) : - و هذه الخاصية المسؤولة عن العملية التفعيلية الخاصة بتخبئة البيانات في مصدر البيانات.
 - ٣ - الخاصية (CacheExpirationPolicy) : - وهي خاصية مهمة ، و من الممكن أن تشكل فارقاً كبيراً و هذه الخاصية لها إختياران هما (Sliding) و الآخر (Absolute) و الآن لنقوم بشرح كل واحدة منهم :-
- (Sliding) : في هذه الحالة سوف يقوم مصدر البيانات بتخبئة البيانات للفترة المحددة لتخبئة البيانات ، لكن في حالة استدعاء بيانات من مصدر البيانات سوف يتم تمديد وقت التخبئة إلى المدة التي تم تحديدها ، بمعنى أنه إذا قام المستخدم بعمل (Refresh) للصفحة سوف يقوم مصدر البيانات بحساب وقت التخبئة من جديد ، و إذا قام المستخدم بعمل (Refresh) في الثانية ٤٥ و كانت مدة التخبئة ٦٠ ثانية سوف يقوم مصدر البيانات بإعادة الوقت من جديدة أي أن الـ ٤٥ ثانية سوف تكون صفراً ليبدأ العد من جديد .
 - (Absolute) : في هذه الحالة سوف تكون تخبئة البيانات لمدة محددة ليس أكثر ، بمعنى تحت أي ظرف من الظروف سوف يقوم مصدر البيانات بإحضار البيانات من جديد بعد الوقت المحدد لتخبئة البيانات .



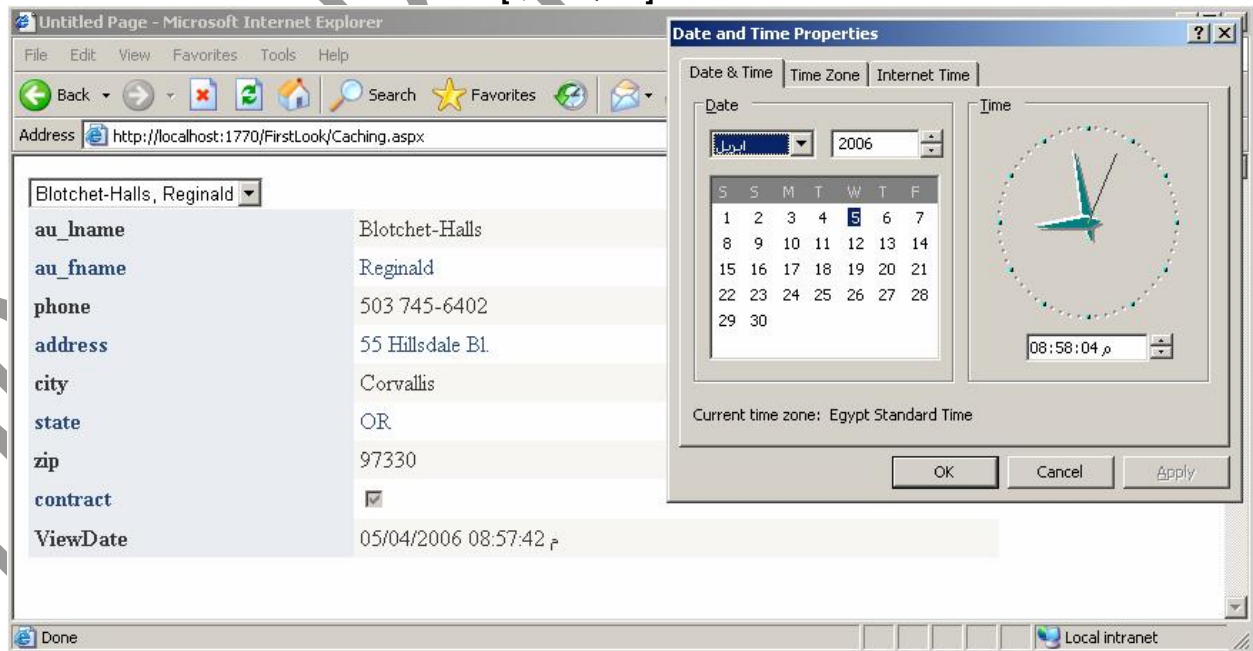
الشكل 67 [يوضح الوقت الذي تم إحضار البيانات فيه بعد تشغيل هذا المثال]

Greene	Morningstar
Blotch-Halls	Reginald
Yokomoto	Δkiko

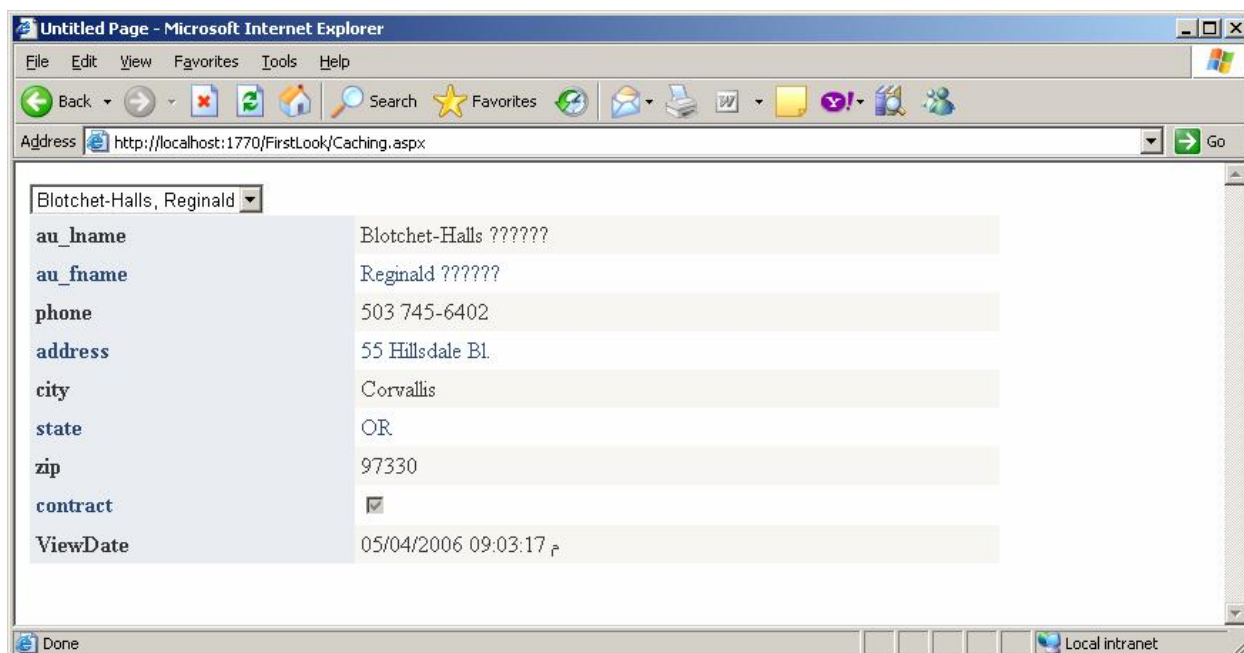
الشكل ٦٨ - [وفيه نقوم بالدخول على جدول البيانات و نقوم بتعديل أسم المؤلف المعروف]

Greene	Morningstar	61
Blotch-Halls ?????	Reginald ?????	50
Yokomoto	Δkiko	41

الشكل ٦٩ [شكل بعد التعديل]



الشكل 70 [نلاحظ عدم تغير البيانات مع أننا قمنا بتغيير البيانات و هذا ما حدث نتيجة لتخبة البيانات (Caching)]



الشكل 71 [يلاحظ ظهور تحديث للبيانات و لكن بعد الوقت المحدد لتخبة البيانات]

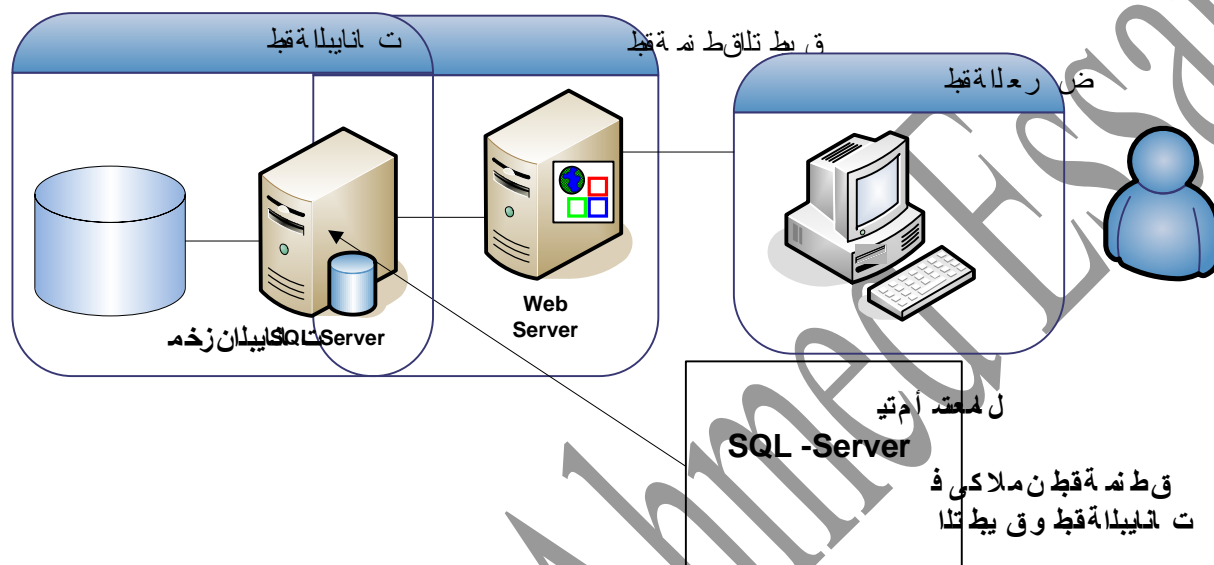
هل لاحظت أن أسم المؤلف في الـ (Drop Down list) لم يتغير؟؟؟
السبب بكل بساطة أن الـ (Visual Studio.net 2005) يقوم بمعالجة إرتداد الصفحات (Post Back) كان هذا مطلوب عمله لتوفير الوقت المستهلك في إحضار البيانات ، بالإضافة إلى أنها تحافظ على حالة العنصر عند إعادة عرض الصفحة.

§ هناك نوع آخر من تخبة البيانات يسمى (Caching with Dependency) :- و معنى هذا المصطلح أنه سوف يتم تخبة البيانات و لكنه سوف يتم التأكد "هل تغيرت البيانات؟" ؛ ففي حالة أن البيانات تغيرت سوف يقوم مصدر البيانات بإحضار البيانات التي تغيرت ، هذا الموضوع يتطلب إعدادات خاصة في قاعدة البيانات و هي أنه سوف يتم إنشاء جدول بالعناصر و حالتها و يتم إنشاء حدث (Trigger) يقوم بمراقبة حدوث أى تغيير لجدول معين ؛ ثم يقوم الحدث (Trigger) بتخزين وقت آخر للتعديل ؛ ليقوم مصدر البيانات بعد ذلك بالتأكد من أن البيانات لم تتغير من هذا الجدول.
§ على ما أظن أن هناك بعض المحترفين الذين سوف يقولون أننا لا نتعامل مع البيانات بهذا الشكل و أننا نتعامل بمنطق تعدد الطبقات أو كما يسمى (N-Tier) ، و أن كل هذه الإمكانيات لا تنفعهم ، بكل بساطة مايكروسوفت قامت بتخصيص أداة جديدة لهذا الغرض ، و لا تقلق من ناحية كيفية الإستعمال لأن هناك دائما مثال لتوضيح هذه الأمور.
§ و لكن أولاً لابد من توضيح معنى كلمة "تعدد الطبقات" أو (N-Tier) ، الفكرة بكل بساطة أنه يتم تقسيم البرنامج من ناحية التصميم و البناء الداخلى الى الأقسام الأكثر شيوعاً و هو البناء المعتمد على "3 طبقات" (3 Tiers Architecture) :-

الطبقة الأول 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 في هذه الطبقة .
الطبقة الثالثة و الأخيرة : هي طبقة العامل مع المستخدم و هذه الطبقة هي الوصلة بين منطق التطبيق (Business logic) و المستخدم.

§ الشكل التالي يوضح العلاقة بين الطبقات و بعضها.



الشكل ٧٢] نرى هنا أن التطبيق مقسّم 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 الإمكانيات تمكنك من أن تضع جزء كبير من طبقة منطق التطبيق في الـ (SQL Server) ، و الآن لنبدأ مثال - إلى حد ما كبير - سنقوم فيه 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
    End Function

```

```
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
```

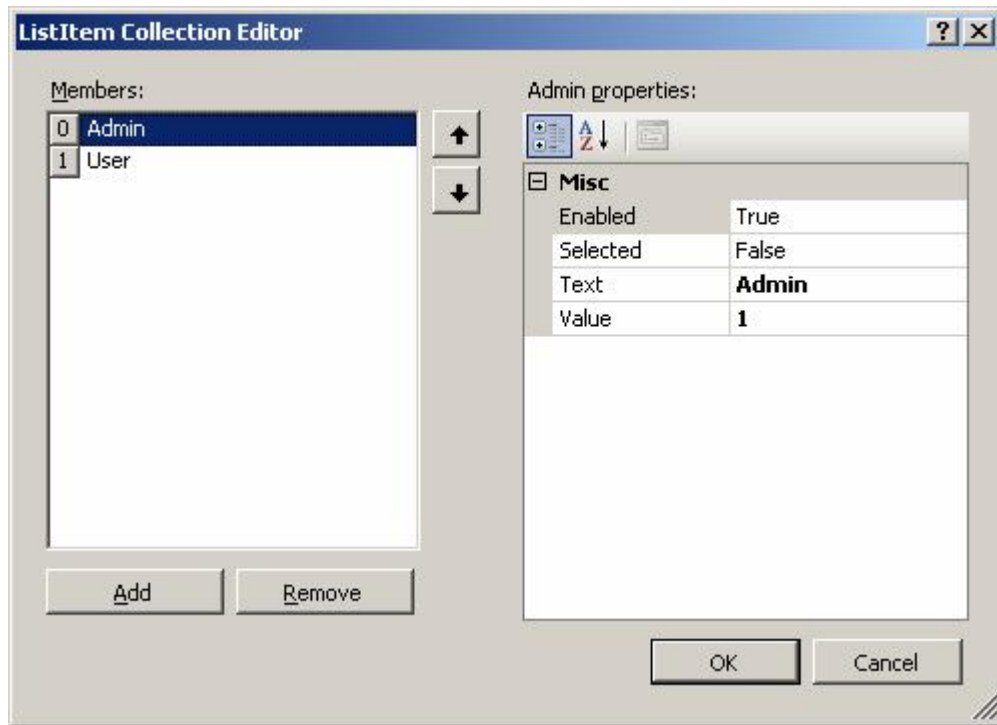
كود التعامل مع قاعدة البيانات بصيغة الـ (VB.net)

```
CREATE PROCEDURE dbo.spGetAuthors AS
select * from authors
RETURN
```

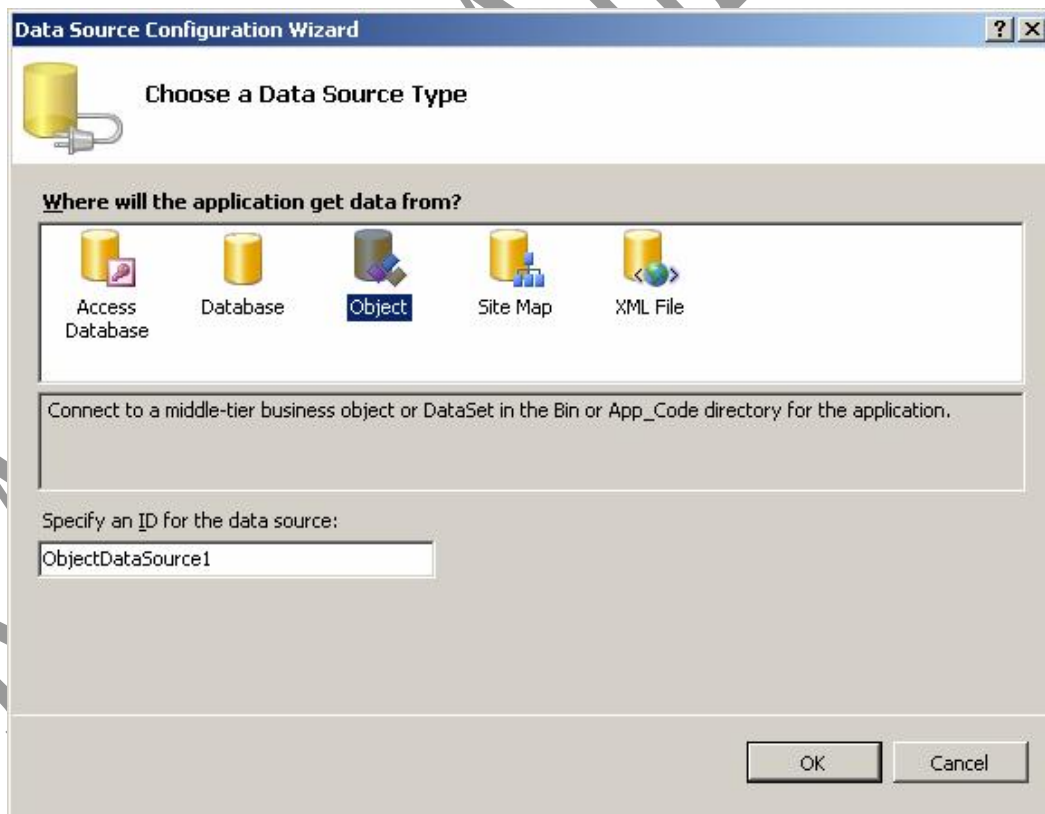
الأجرائية المخزنة (GetAuthors)

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

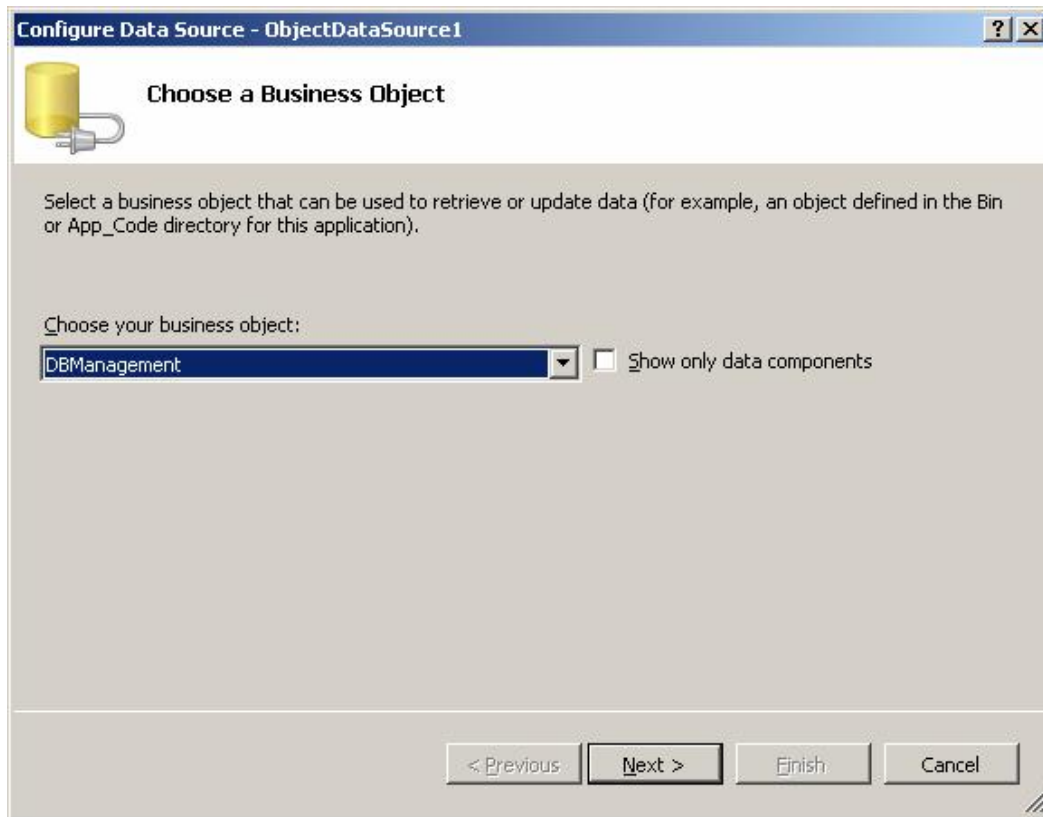
الأجرائية المخزنة (GetAuthorByIDWithPermission)



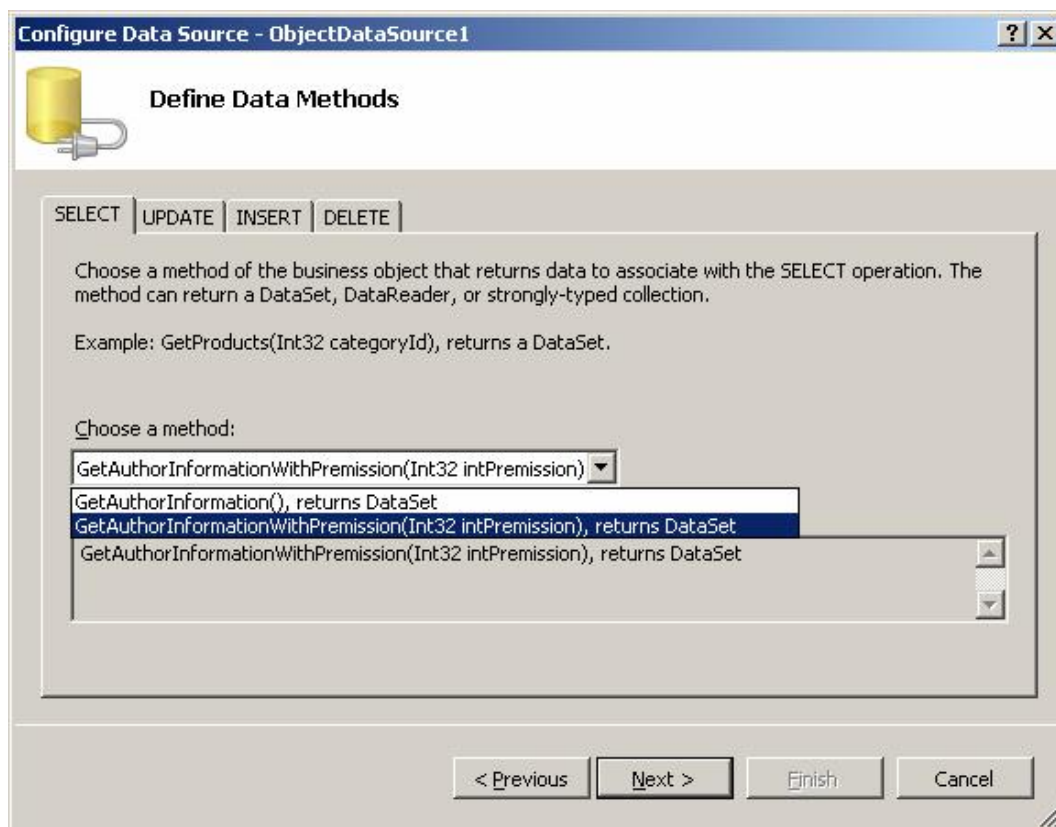
الشكل 73 [نقوم بإضافة (DropDownList) ، و نقوم بإعداد محتوياتها- كما نرى- و هي (Admin) و القيمة الخاصة به هي (1) و (User) و القسم الخاص به صفر]



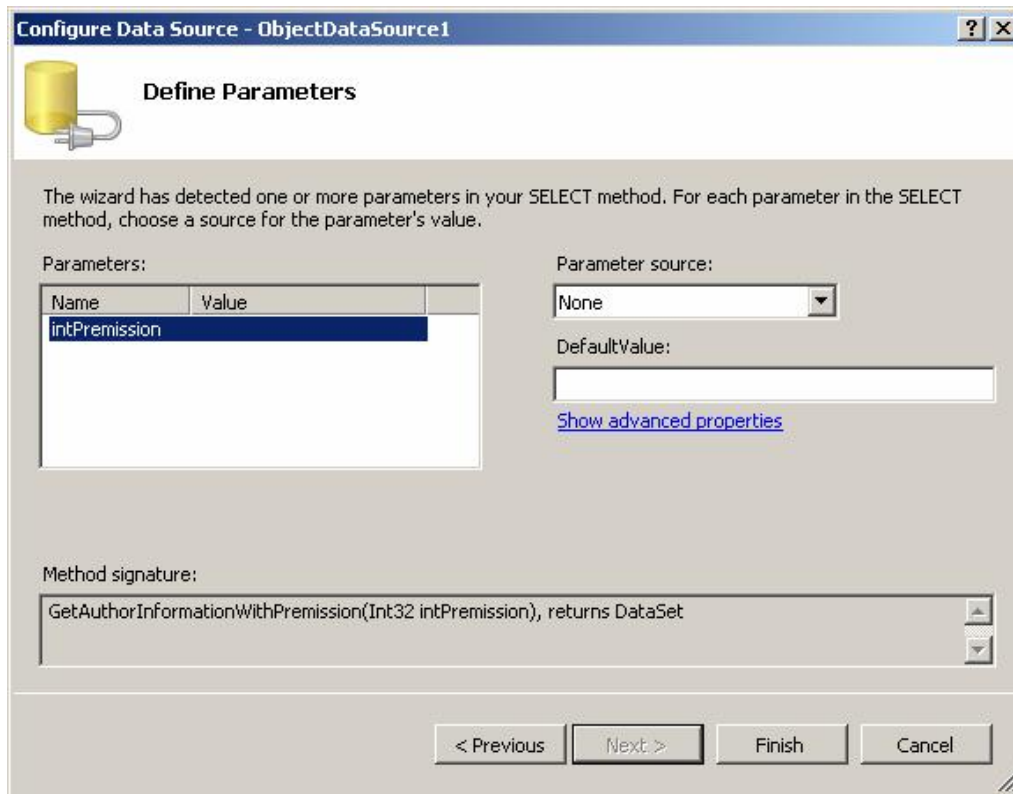
الشكل 74 [نقوم بوضع (Grid View) على الصفحة ، و بعد ذلك نقوم بإعداد مصدر البيانات ، - و كما نرى و على غير العادة- سوف نغير مصدر البيانات من (Database) إلى (Object) و نقوم بتسميته ثم نضغط على (OK)]



الشكل 75 [سوف نختار فيه المكتبة التى تتعامل مع البيانات ، أو بمعنى أصح الـ (Business Tier) التى سوف تمد الـ (Interface) بالبيانات]



الشكل ٧٦] يتم فيه إختيار الدالة التي سوف ترجع لنا البيانات أو الدالة التي تحتوى على منطق البرنامج ، و- في حالتنا- هنا وضعنا الكود الخاص بمنطق البرنامج في الدالة (GetAuthorInformationWithPermission) و هذه الدالة تستقبل متغير من نوع رقمى [(Integer)
السؤال هنا كيف سوف تستقبل هذا المتغير أو من أين سوف يتم أمداد هذه الدالة بهذا المتغير؟
الإجابة في الشكل التالى [77]



الشكل ٧٧ [-الأجابة عن السؤال السابق في هذه النافذة و كما نرى- هناك قائمة في يسار الشاشة تحتوى على المتغيرات التى تحتاجها الدالة و على اليمين قائمة بالمصادر التى 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 [نقوم باختيار القائمة المنزلة (Drop Dow List) و التي تحتوى 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...

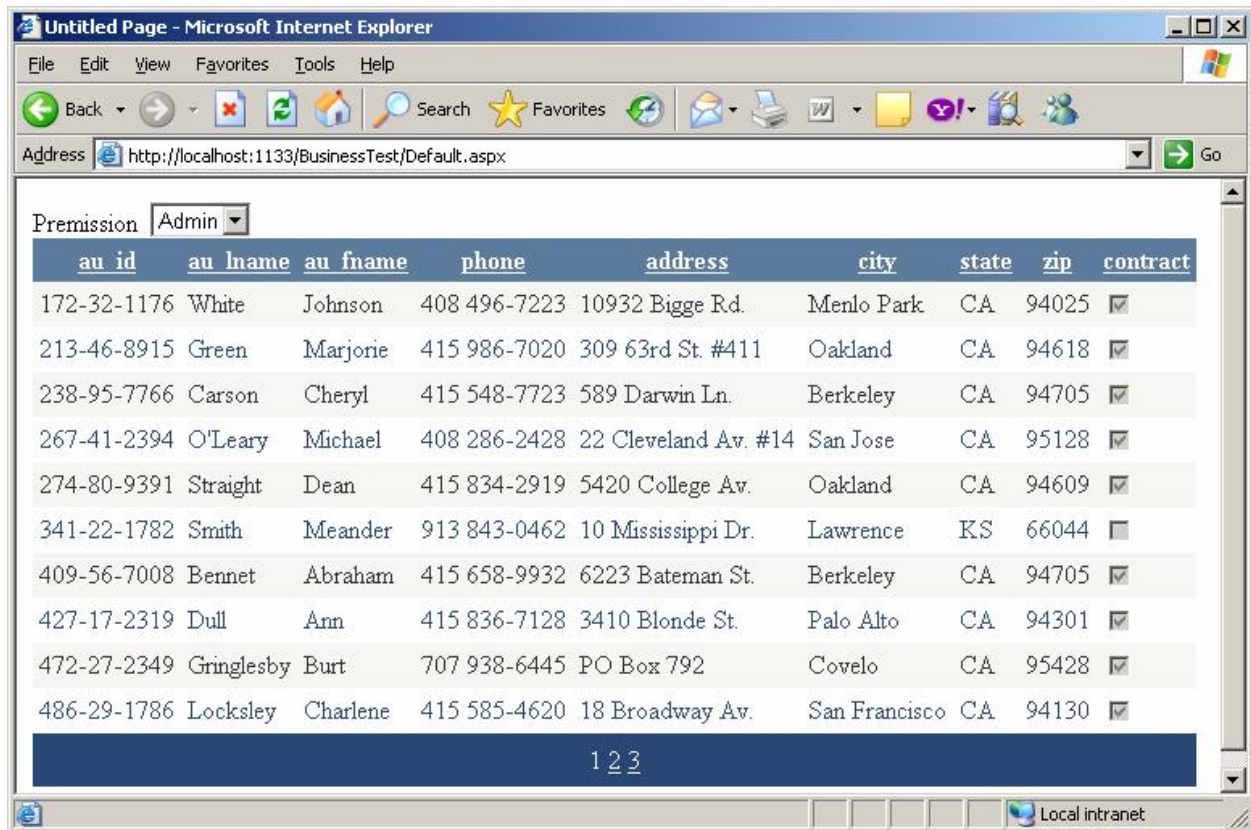
☒ Enable Paging

☒ Enable Sorting

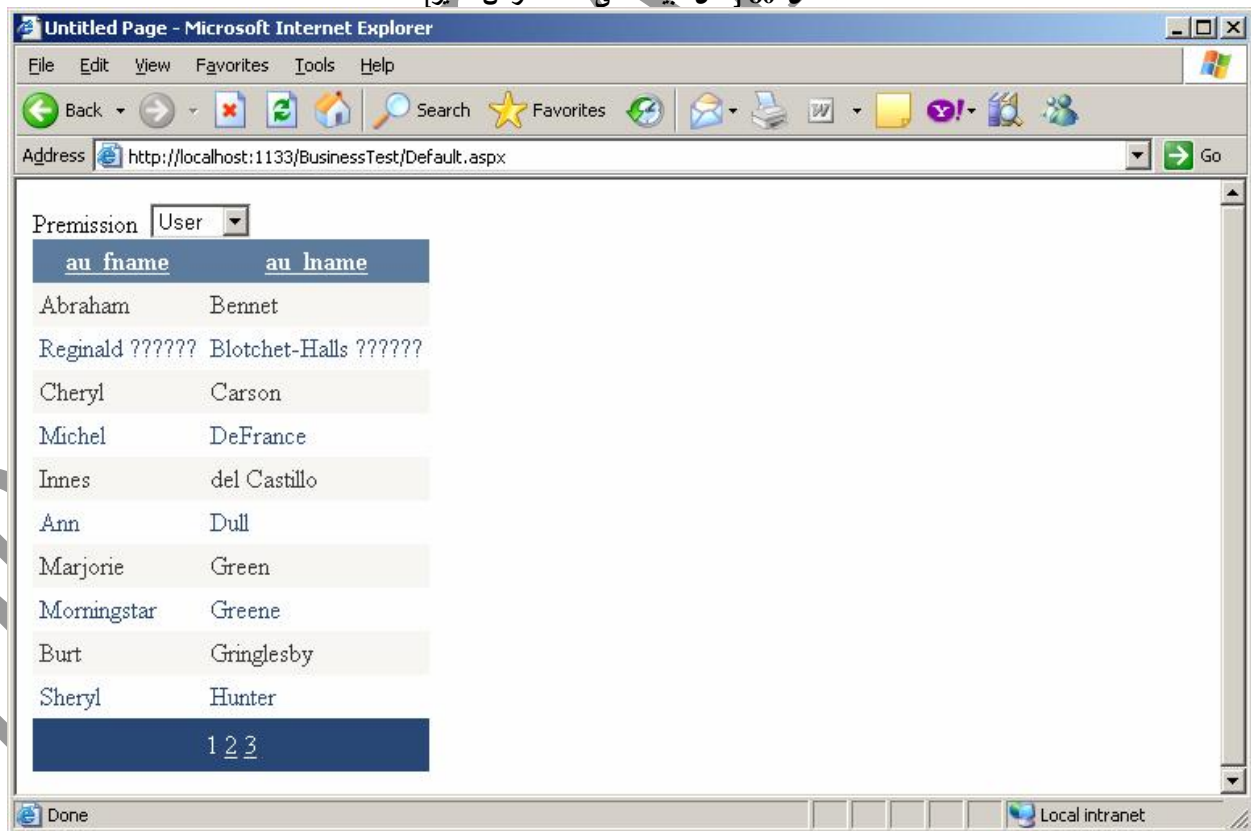
☐ Enable Selection

Edit Templates

الشكل ٧٩ [- شئ رائع آخر- وهي أنه مازال بإمكاننا إستعمال خاصية السرد و تقسيمه إلى صفحات فى الـ (Grid View)]



الشكل 80 [شكل البيانات في حالة العرض لمدير]



الشكل 81 [شكل في حالة العرض لمستخدم عادي]

§ والآن بعد أنتهاء المثال لنوضح الأمور قليل ، ففي المثال السابق قمنا بالتالي : -

- ١- أردنا وضع قانون صغير لعمل التطبيق ألا وهو أن هناك نوعان من المستخدمين (مدير / مستخدم)
- ٢- قمنا بعمل مكتبة تحتوى على هذا القانون و هذا ما يسمى بطبقة منطق التطبيق هذه المكتبة تقوم بالإتصال و بإحضار البيانات طبقا للقانون الذى وضعناه للتطبيق
- ٣- بعد ذلك يقوم الفيچوال ستوديو بإنهاء العمل مع البيانات من حيث ضبط مصادر البيانات للدوال الخاصة بمكتبات منطق التطبيق أو مصدر البيانات لأدوات العرض مثل (Grid View) و الـ (Detail view) ... و هكذا.

§ تأمين قاعدة البيانات

و الآن ننتقل الى موضوع آخر مفيد جدا لمن يريد تأمين قاعدة البيانات وتتم عن طريق تشفير جملة الإتصال فى الملف (Web.Config) ، يجب أولا توضيح ما سيتم علمه لحماية جملة الإتصال بكل بساطة فى بداية تحميل الصفحة سوف نقوم بفك تشفير جملة الإتصال (Decrypt the connection string) ، و عند الإنتهاء نقوم بعمل تشفير لجملة الإتصال (Encrypt the Connection string) ، و لكن فى أول مرة سوف نقوم بعمل صفحة مؤقتة تقوم بتشفير جملة الإتصال ؛ حتى يتسنى لنا إمكانية ما سبق ذكره . و لكن قبل هذا سوف نقوم بعمل دالتين واحدة لتشفير جملة الإتصال و الأخرى لفكر التشفير.

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

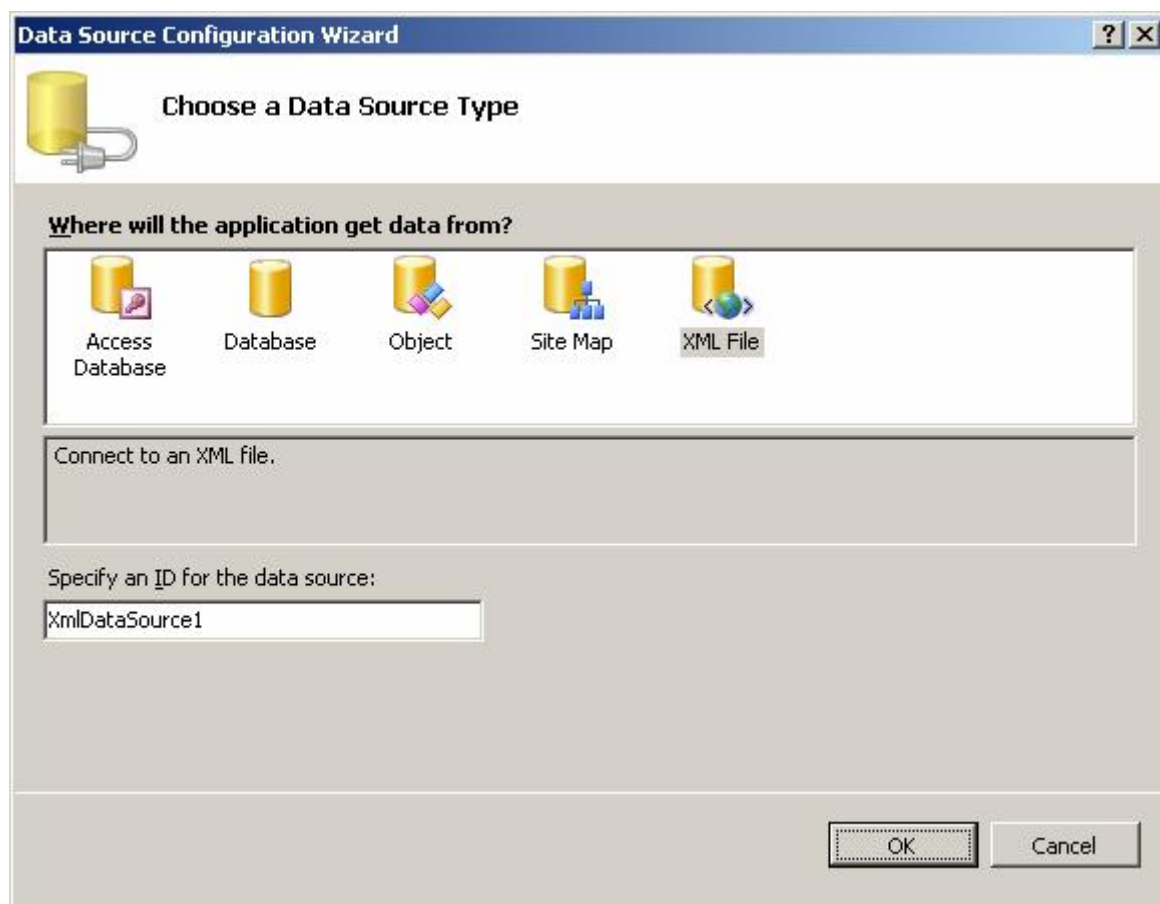
<CipherValue>AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAA5Ktsb5/qvEyHjwCtfMWR1wQAAAAACAAAA
AAADZgAAqAAAAABAAAADep0UAEYD4/sCOUtrPTYcAAAAAASAAACgAAAAEAAAAAPoSeBLOmk6nK8KbO
r2Q3q34AQAAAdYY5dqi5avWAT7hoODwLvcpBbm7mkLp/+FPedccvJmX0JdR6zqNEEOXrIT8/e2WkaT
WfxruM2MDwo/E2YGTW7/Cu4tV4V26tKwxwPiy2oUhoKEjYwKIVEeuQU2QzT1qQ27jSRjUEZnFnw+P
Z2B9g3+xROUAzeWtzFWfKpn3tJwEUcHgEGecPL0kIwU0FZYCbSzQwdLgk88q+bvJEqzVCR18Ozldu
G9rNqJsUvYVgmiSty017q4n2/3CeJldysCo8kV1Pz85c1JTZplgstwyZVi0myAEcjGLpU2A3e7fJE
IUnQphxmIsW/9Yyi0H3ntUSqVbFo7gNOoz8TOvY45yO0tfo099ApYsZMI4Oh8i2H5RzoB290bE4p0
2KNzIfoF/N0Fqp4zhJu7sNJJ2ejXoujKGh802p2sj3+tdi4+S3x08UoaQVJ3KfMPlx0JnMsLBK799
qS0K6+Pb1Wzp0tIOc9GqIEECwVIFnhjtp+Vyl12kCDW++FQ5da28GQjZULt8LgwauObVXweGgrkO0
O4kNMmrQuIqR5OLRh8BA3TusPVAso9QPzIyDreSbS6grzB3dQrqIhp+fgA2wRBh/m/2pU9ZNcsHUCF
TGZ2WtKqjFZLGP4fqUVio3J31iU0zMW1k5vmPmLojefvzxyg9PK9nXhhr0pj/cj1Ig5FAAAAPPuOY
MVAJeVFT9UqRV28WDFNXbf</CipherValue>
  </CipherData>
</EncryptedData>
</connectionStrings>
```

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

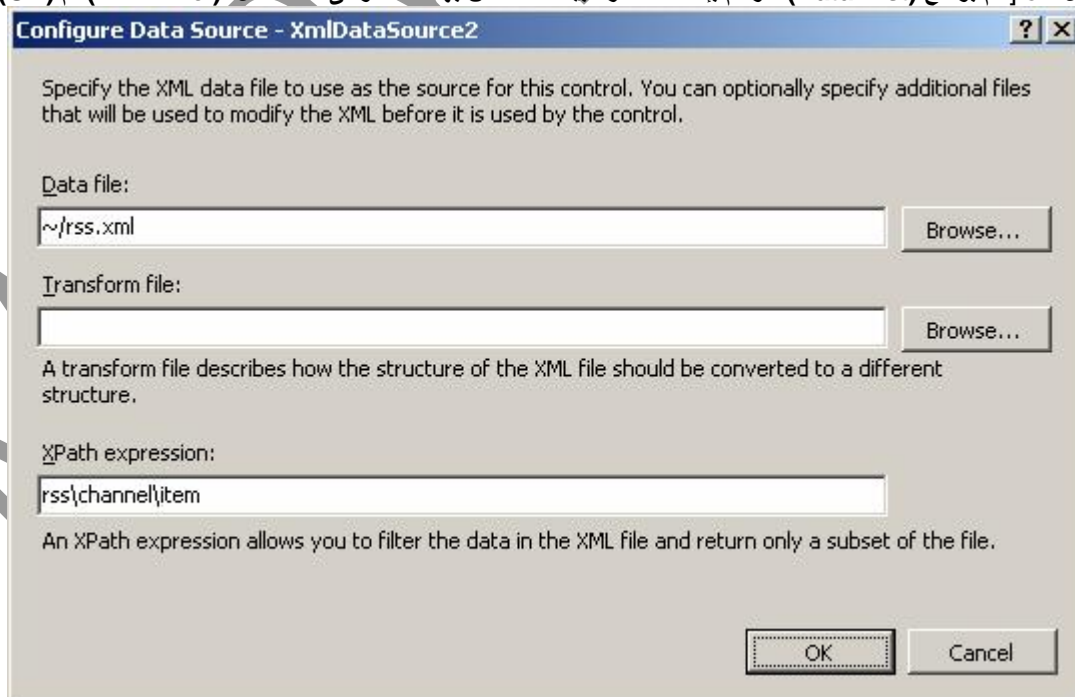
```
Private Sub DecryptMyConnectionString()
    Dim config As Configuration = _
System.Web.Configuration.WebConfigurationManager.OpenWebConfiguration(_
    Request.ApplicationPath)
    فتح ملف الـ (Web.config)
    Dim section As ConfigurationSection = config.Sections("connectionStrings")
    تحديد النطاق المراد فك تشفيره
    section.SectionInformation.UnprotectSection()
    فك تشفير النطاق الذي تم إختياره
    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 بإعداد 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 (Template item) - و كما نرى - نأخذ البيانات من ما يسمى (XPath) ، و هذا يبدو مختلفاً عما كان عليه في مصادر البيانات الأخرى حيث أن مصادر البيانات الأخرى كانت تستخدم ما يسمى بالـ (DataBind) .

و فى النهاية هذا الفصل أود أن نتذكر سوياً ملخصاً لما تعلمناه فى 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. تخبئة البيانات (Caching) و كيفية الاستفادة منها .
٩. بناء جمل الإستعلام على أدوات البيانات .
١٠. كيفية الربط بين أدوات البيانات الموجودة على الصفحة .