

الفصل الثامن

سلسلة

ASP.NET

خطوة بخطوة حتى الاحتراف

C# & VB

اعداد المهندس

محمد عمر الحاج خلف

الفصل الثامن

استخدام الأداة GridView

في هذا الفصل

- ✓ عرض وتعديل وحذف البيانات
- ✓ اختيار البيانات وترتيبها وتقسيمها إلى عدة صفحات
- ✓ البحث ضمن البيانات
- ✓ تنسيق مظهر الأداة GridView
- ✓ أنماط الحقول ضمن الأداة GridView
- ✓ عرض الصور ضمن الأداة GridView
- ✓ الأحداث التي تطلقها الأداة GridView
- ✓ تمييز الأسطر وفق شرط معين
- ✓ استخدام الأداة DropDownList داخل الأداة GridView
- ✓ استخدام GridView داخل GridView آخر

استخدام أداة التحكم GridView

تعتبر الأداة GridView من أكثر أدوات ASP.NET تنوعاً بالخصائص والمهام التي يمكن تنفيذها من خلالها , في هذا الفصل سنحيط بجميع المهام الأساسية لهذه الأداة حيث بالإمكان استخدامها لعرض البيانات , تحديد البيانات التي سيتم عرضها , ترتيب البيانات , عرض البيانات على عدة صفحات , تحرير البيانات وتعديلها , كما سنتعلم كيفية إضافة أدوات التحكم الأخرى ضمن الأداة GridView , وكيفية استخدام أدوات التحقق من الدخول مع الأداة GridView , بالإضافة إلى ماسبق ستكون لنا وقفة مع كيفية تخزين الصور في قاعدة البيانات وعرضها على صفحات الويب , بالإضافة إلى مواضيع عديدة سنتم مناقشتها خلال هذا الفصل .

ملاحظة

كل فقرة ضمن هذا الفصل هي امتداد للفقرة التي تسبقها , لذلك أرجو قراءة هذا الفصل من بدايته وعدم القفز بين الفقرات .
كما أنصح بقراءة فصل " أساسيات التعامل مع قواعد البيانات " قبل الخوض في هذا الفصل .

تحضير قاعدة البيانات

قبل البدء بفقرات هذا الفصل يجب علينا بناء قاعدة البيانات والجداول التي سنقوم بالعمل عليها , تعلمنا خلال الفصول السابقة كيفية عمل هذا , سنقوم ببناء جدولين , الجدول الأول لتخزين معلومات عن أقسام الشركة والجدول الآخر لتخزين بيانات الموظفين , جدول الأقسام يحمل الاسم departments وله البنية التالية :

اسم الحقل	نمط البيانات	وصف الحقل
Id	Int	مفتاح رئيسي يأخذ قيمته بشكل تلقائي دون تدخل المستخدم
Name	Nvarchar(50)	لتخزين اسم القسم
Description	Nvarchar(100)	وصف القسم

	Column Name	Data Type	Allow Nulls
🔑	id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
	description	nvarchar(100)	<input checked="" type="checkbox"/>

أما جدول الموظفين فنعطيه الاسم employees وله البنية التالية :

وصف الحقل	نمط البيانات	اسم الحقل
مفتاح رئيسي يأخذ قيمته بشكل تلقائي دون تدخل المستخدم	Int	Id
لتخزين اسم الموظف	Nvarchar(50)	Fname
كنية الموظف	Nvarchar(50)	Lname
متزوج أم لا	Bit	Marriage
الراتب	Int	Salary
رقم القسم الذي يتبع له	Int	deptID
صورة شخصية	Nvarchar(50)	Photo

حيث إن الحقل deptID من الجدول employees سيكون بمثابة مفتاح أجنبي Foreign Key مرتبط بالحقول id من الجدول departments (لم أقم بوضع قيد مفتاح أجنبي ضمن قاعدة البيانات على الرغم من جواز ذلك , إنما لم أشأ إدخال القارئ بأمر يفترض أنها من أساسيات قواعد البيانات ولست هنا بصدد دراسة قواعد البيانات) .

نقوم بملئ جدول الأقسام بالبيانات التالية :

discreption	Name	Id
يهتم بالشؤون الإدارية	قسم الإدارة	1
يهتم بالقضايا المالية	قسم المحاسبة	2
يهتم بالعلاقات الخارجية	قسم العلاقات	3

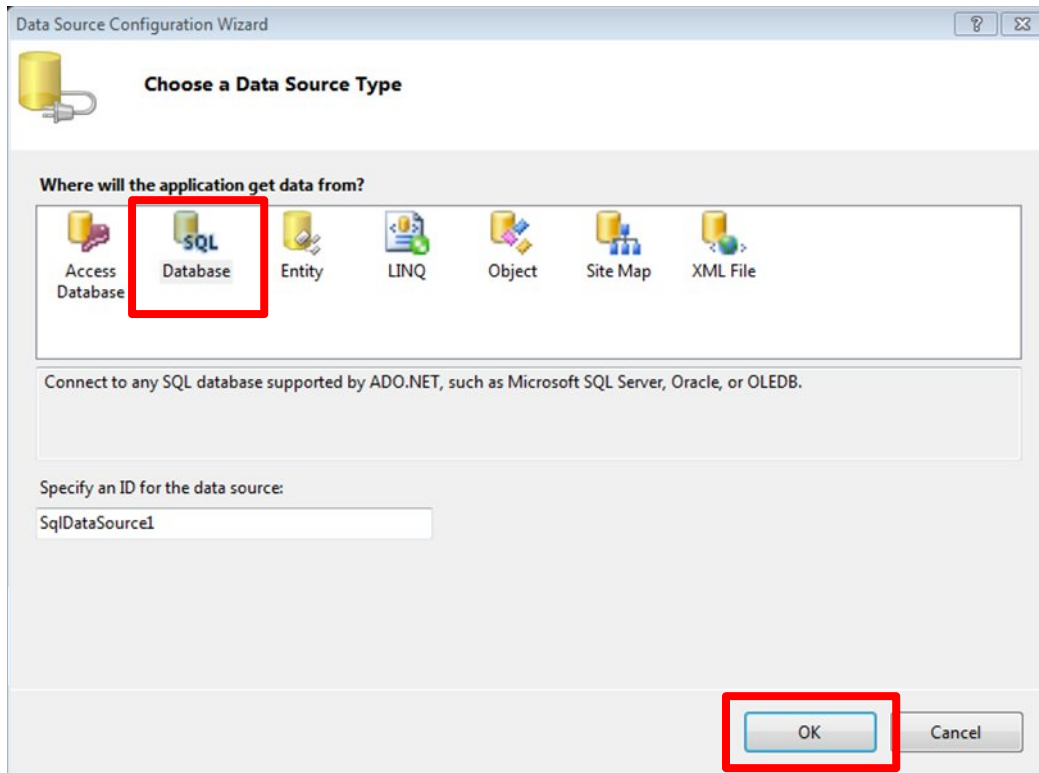
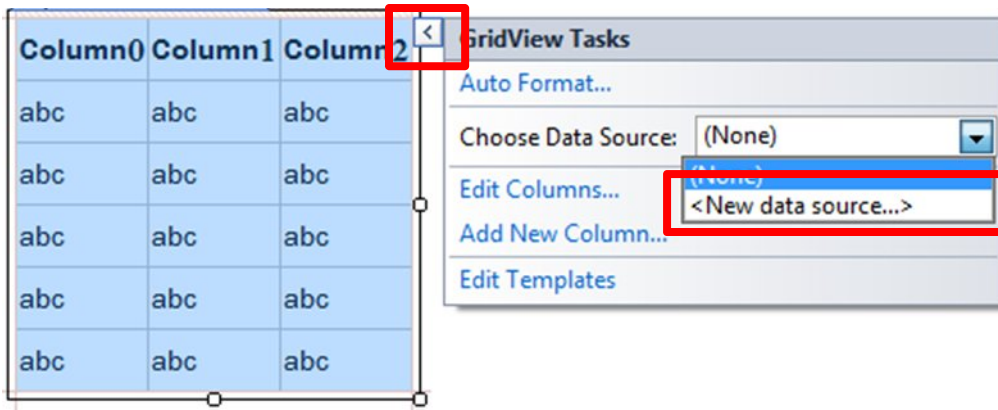
كما نقوم بملئ جدول الموظفين بالبيانات التالية :

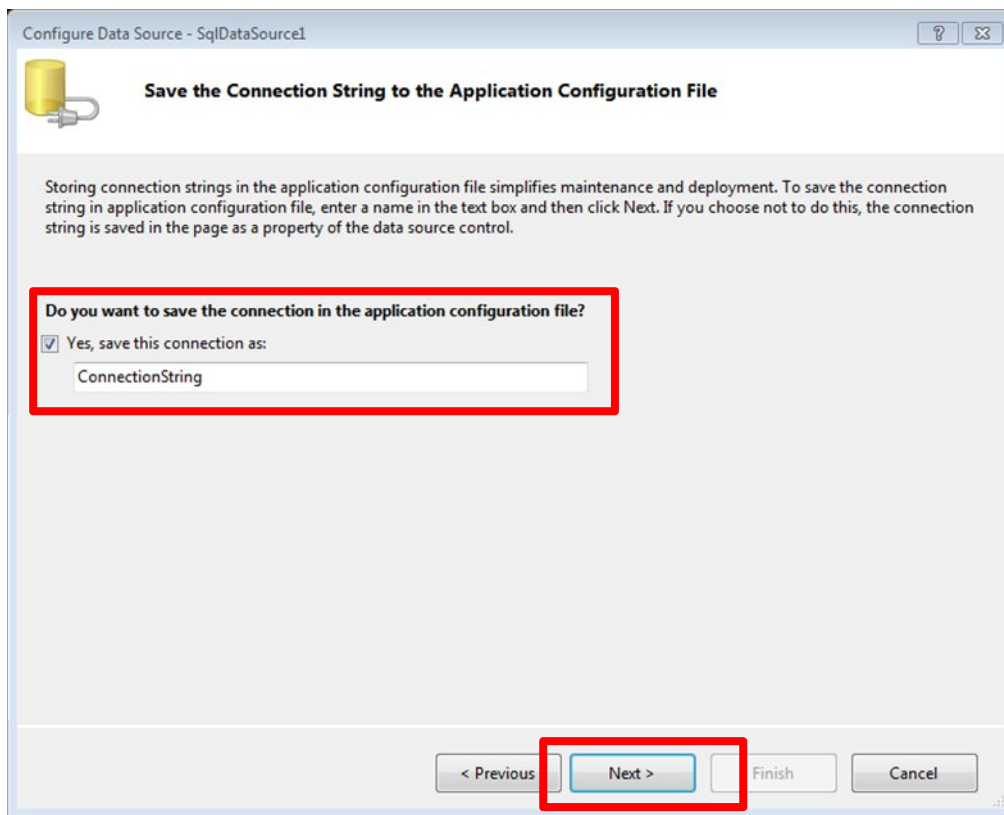
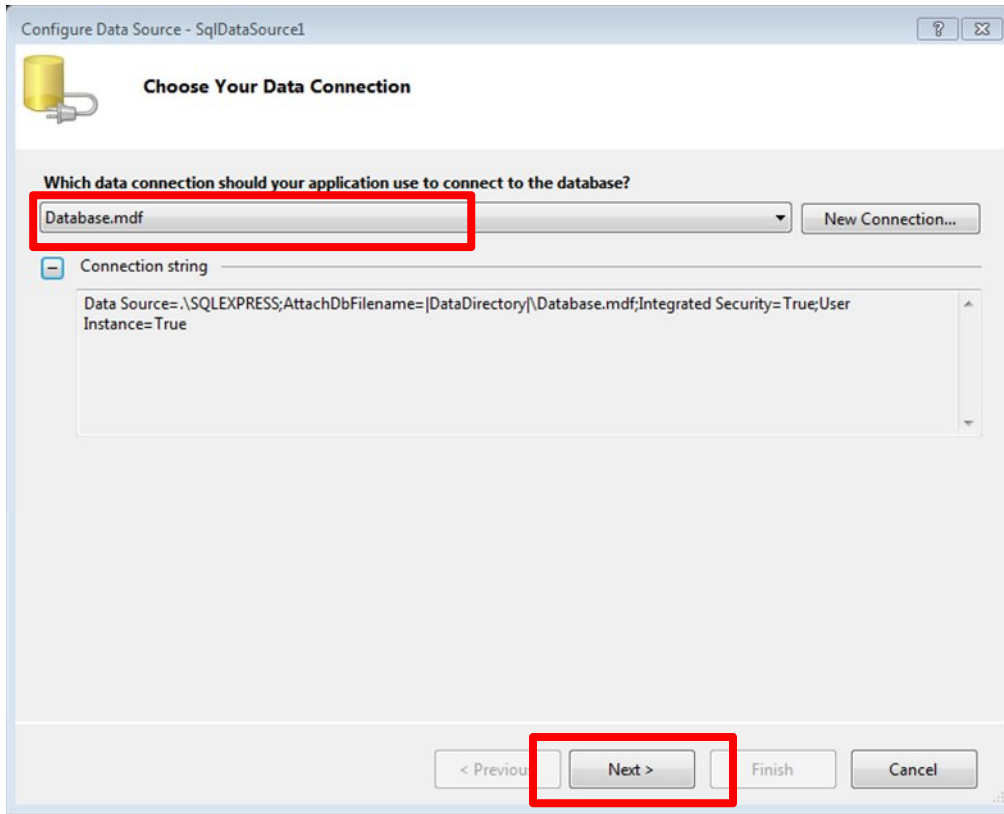
photo	deptID	salary	marriage	lname	Fname	id
~/images/1.jpg	1	1000	0	الحاج خلف	محمد	1
~/images/2.jpg	1	1500	0	أحمد	إيمان	2
~/images/3.jpg	2	1200	1	المصطفى	أحمد	3
~/images/4.jpg	2	1300	0	الخلف	بتول	4
~/images/5.jpg	3	900	1	ابراهيم	قصي	5
~/images/6.jpg	3	800	1	مراد	لين	6

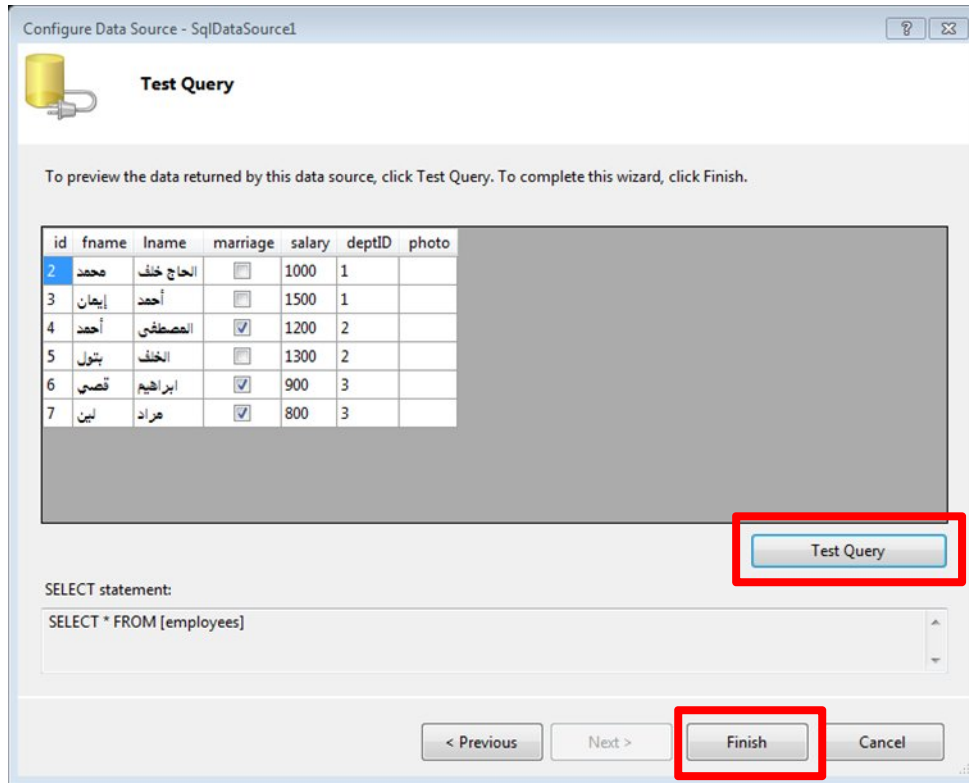
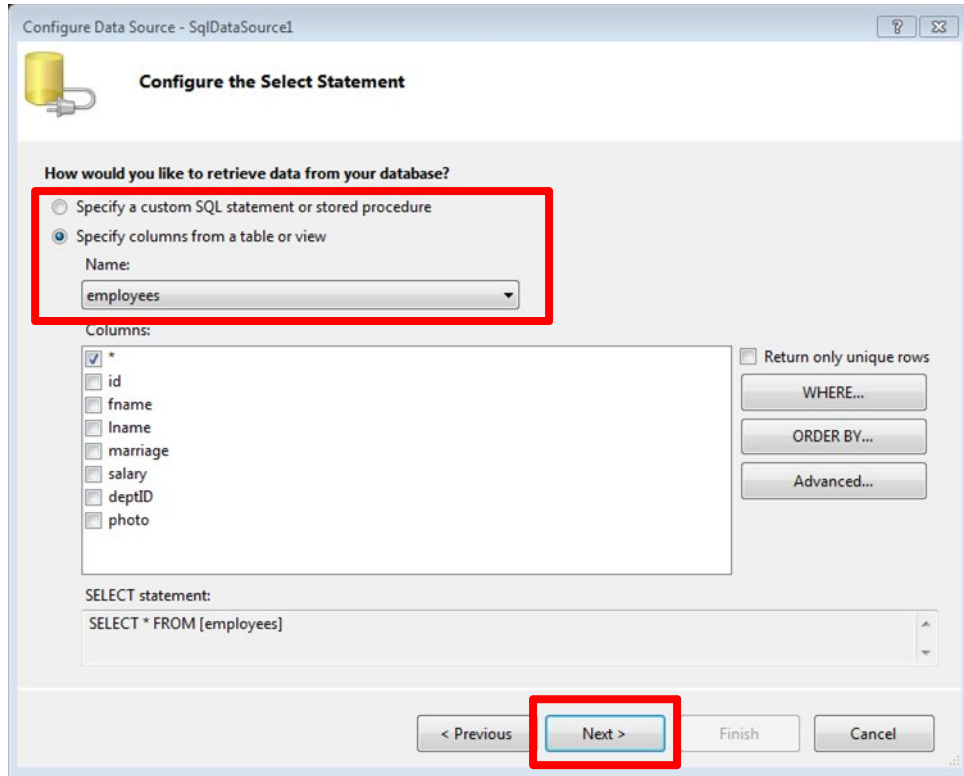
قم بإنشاء مجلد جديد ضمن مسار المشروع الحالي وسمه images وضع بداخله خمس صور ولتكن أسماؤها مطابقة لأسماء الصور التي أدخلناها في الجدول السابق. بهذا نكون قد انتهينا من إعداد قاعدة البيانات وأصبحنا جاهزين للبدء بفقرات هذا الفصل .

عرض البيانات

تمكننا أداة التحكم GridView من عرض البيانات على صفحة الويب بسهولة كبيرة , حيث يتم الاتصال مع قاعدة البيانات بواسطة الأداة SqlDataSource , أضف صفحة جديدة و قم بإضافة الأداة GridView إليها والتي تجدها ضمن القسم Data ضمن نافذة الأدوات Toolbox ثم تابع الخطوات التالية :







بعد الانتهاء من الخطوات السابقة , بإمكاننا تنفيذ الصفحة لتظهر البيانات ضمن الأداة GridView كما توضح الصورة التالية :

id	fname	lname	marriage	salary	deptID	photo
2	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
3	إيمان	أحمد	<input type="checkbox"/>	1500	1	
4	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	
5	يتول	الخلف	<input type="checkbox"/>	1300	2	
6	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3	
7	لين	مراد	<input checked="" type="checkbox"/>	800	3	

أما كود الصفحة فيكون :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="fname"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="lname"
        SortExpression="lname" />
      <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
        SortExpression="marriage" />
      <asp:BoundField DataField="salary" HeaderText="salary"
        SortExpression="salary" />
      <asp:BoundField DataField="deptID" HeaderText="deptID"
        SortExpression="deptID" />
      <asp:BoundField DataField="photo" HeaderText="photo"
        SortExpression="photo" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]" />
</div>
```

ما قمنا به فعليا هو ربط الأداة SqlDataSource مع الجدول employees حيث ستقوم بجلب البيانات المحددة ومن ثم قمنا بربط الأداة GridView مع مصدر البيانات SqlDataSource وذلك عبر الخاصية DataSourceID كما هو موضح في الكود السابق والذي سنقوم بشرحه بالتفصيل ضمن الفقرات اللاحقة.

اختيار البيانات

سنقوم ضمن هذه الفقرة بتوضيح كيفية تحديد البيانات التي ستعرض ضمن الأداة GridView بناءً على ما اختاره المستخدم ضمن GridView آخر .

id	name
Select 1	قسم الإدارة
Select 2	قسم المحاسبة
Select 3	قسم العلاقات

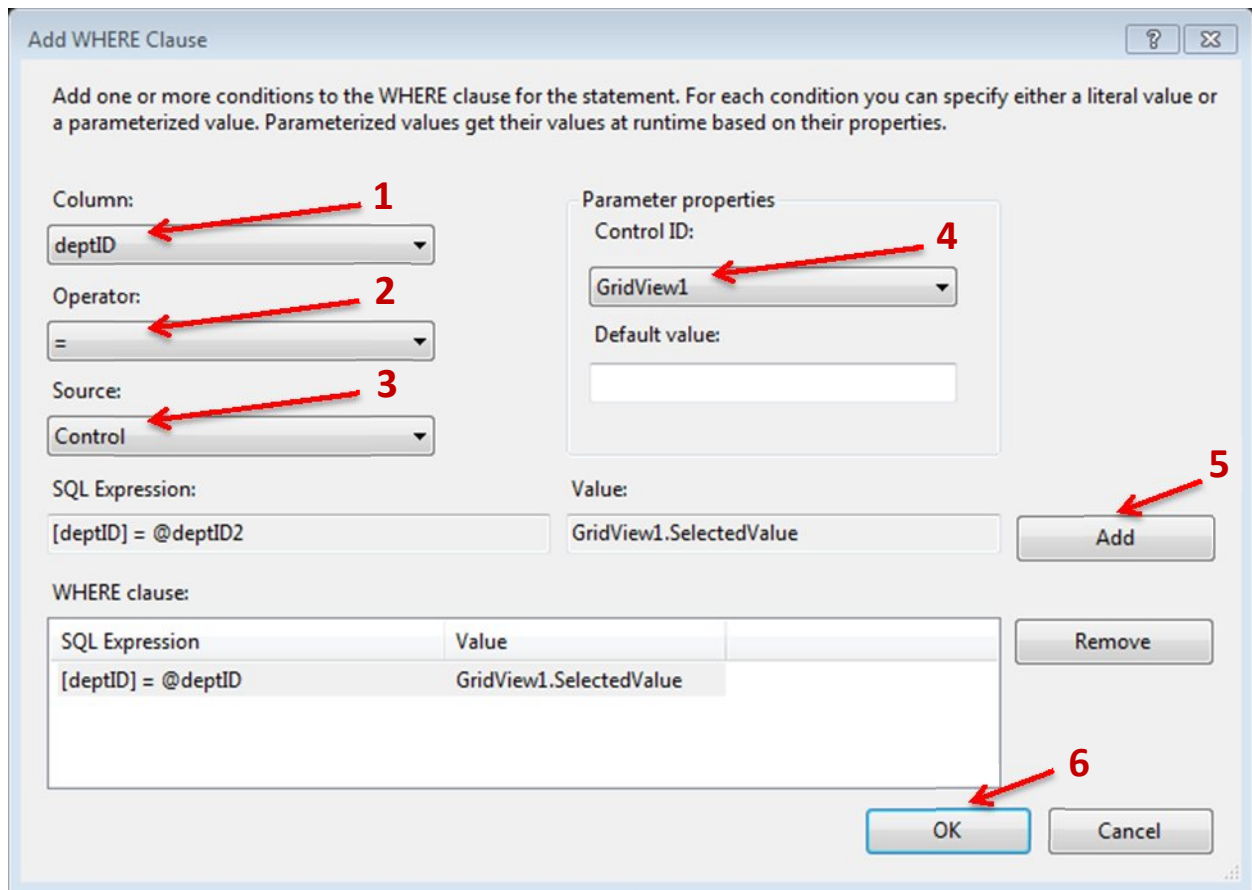
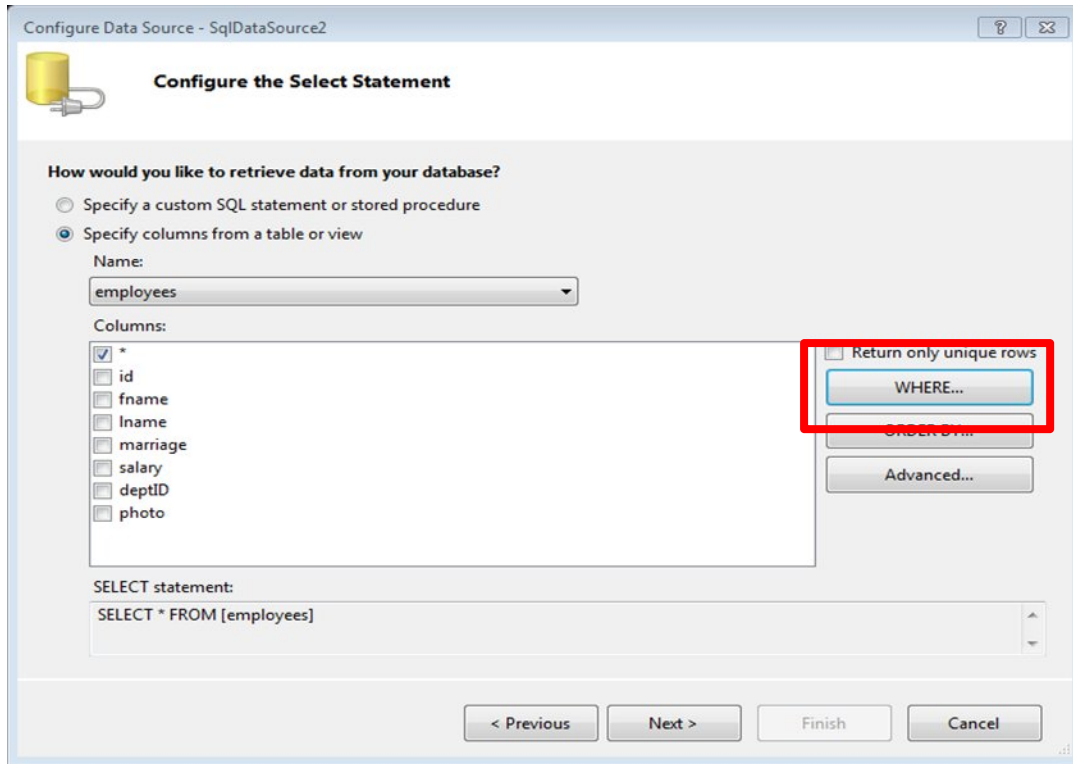
id	fname	lname	marriage	salary	deptID	photo
4	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	
5	يتول	الخلف	<input type="checkbox"/>	1300	2	

ننشئ صفحة جديدة ونضيف إليها أداة GridView1 ونقوم بربطها مع الجدول departments , نقوم بإعطاء الخاصية DataKeyNames القيمة id , والخاصية AutoGenerateSelectButton القيمة True كما يوضح الكود التالي :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    AutoGenerateSelectButton="True" DataKeyNames="id"
    DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="name" HeaderText="name"
        SortExpression="name" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [id], [name] FROM [departments]"></asp:SqlDataSource>
</div>
```

نقوم بعد ذلك بإضافة أداة GridView2 ونقوم بربطها مع الجدول employees ولكن في عبارة الـ select نضيف شرط أن تكون سجلات الموظفين التي سيتم إحضارها موافقة لشرط أن يكون رقم القسم موافق لرقم القسم الذي تم اختياره ضمن الأداة GridView1 . (لمزيد من المعلومات راجع فصل مقدمة إلة قواعد البيانات) .



أما كود الصفحة كاملا فيكون كالتالي :

ASP.net كود

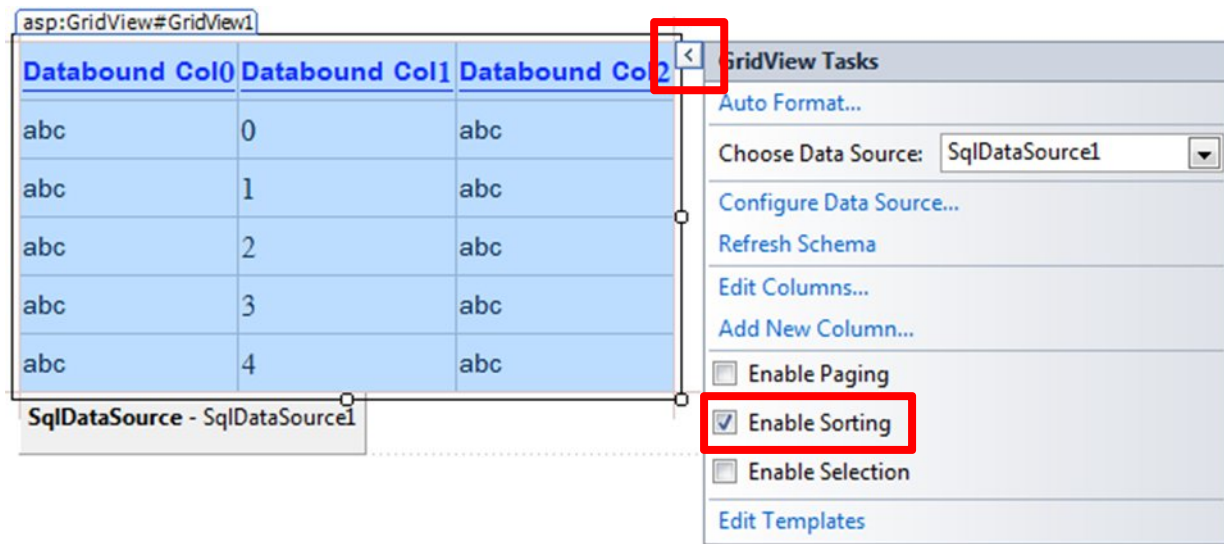
```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    AutoGenerateSelectButton="True" DataKeyNames="id"
    DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="name" HeaderText="name"
        SortExpression="name" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [id], [name] FROM [departments]"></asp:SqlDataSource>
  <asp:GridView ID="GridView2" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource2">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="fname"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="lname"
        SortExpression="lname" />
      <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
        SortExpression="marriage" />
      <asp:BoundField DataField="salary" HeaderText="salary"
        SortExpression="salary" />
      <asp:BoundField DataField="deptID" HeaderText="deptID"
        SortExpression="deptID" />
      <asp:BoundField DataField="photo" HeaderText="photo"
        SortExpression="photo" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource2" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees] WHERE ([deptID] = @deptID)">
    <SelectParameters>
      <asp:ControlParameter ControlID="GridView1" Name="deptID"
        PropertyName="SelectedValue" Type="Int32" />
    </SelectParameters>
  </asp:SqlDataSource>
</div>
```

نلاحظ عند تنفيذ الصفحة السابقة إمكانية اختيار سطر معين (ظهور Select على هيئة link) ضمن الأداة GridView1 حيث تم إسناد القيمة True للخاصية AutoGenerateSelectionButton , يتم معرفة اختيار المستخدم من خلال أحد الطرائق الأربعة التالية :

- ✓ SelectedDataKey() : تعيد غرض DataKey Object مرتبط بالسطر الذي تم اختياره , هذا الخيار مفيد في حال وجود أكثر من data key واحد .
- ✓ SelectedIndex() : تعيد رقم (فهرس) السطر الذي تم اختياره , يبدأ العد من الصفر باعتبار السطر الأول دليله صفر .
- ✓ SelectedValue() : تعيد قيمة data key المرتبط بالسطر الذي تم اختياره , في حال وجود أكثر من data key فإنها تعيد قيمة أول واحد فقط .
- ✓ SelectedRow() : تعيد غرض GridViewRow Object يمثل السطر الذي تم اختياره .

ترتيب البيانات

تدعم الأداة GridView إمكانية ترتيب البيانات المعروضة و بأسلوب تصاعدي Ascending أو تنازلي Descending , وذلك عبر تفعيل الخاصية AllowSorting من خلال إعطائها القيمة True . كما يمكن تفعيل خاصية الترتيب عبر النقر على Smart Tag للأداة GridView ومن ثم تفعيل الخيار Enable Sorting .



لنقم بإضافة أداة GridView وربطها بجدول الموظفين employee وتفعيل خيار ترتيب البيانات بحيث يصبح كود الصفحة شبيه بالتالي :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AllowSorting="true"
    DataSourceID="SqlDataSource1" />
</div>
```

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]" />
</div>
```

عند تنفيذ الصفحة السابقة ستلاحظ أن عناوين الحقول ستصبح كروابط تشعبية , عند النقر على عنوان أحد الحقول سيتم ترتيب بيانات الجدول بناء على بيانات ذلك العمود وبشكل تصاعدي , وعند النقر على عنوان نفس العمود للمرة الثانية فيستمر الترتيب بشكل معاكس لما كان عليه , أي بإسلوب تنازلي , في الصورة التالية قمت بترتيب بيانات الموظفين على أساس الراتب وبشكل تنازلي , أي من الراتب الأعلى إلى الأدنى .

id	fname	lname	marriage	salary	deptID	photo
3	إيمان	أحمد	<input type="checkbox"/>	1500	1	
5	بتول	الخلف	<input type="checkbox"/>	1300	2	
4	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	
2	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
6	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3	
7	لين	مراد	<input checked="" type="checkbox"/>	800	3	

تقسيم البيانات إلى عدة صفحات

عند العمل مع قواعد البيانات الضخمة والتي قد تحتوي على عدد كبير من السجلات ضمن جداولها يصبح من الضروري عرض الجداول على عدة صفحات , تدعم الأداة GridView الخاصية AllowPaging والتي تقوم بتقسيم البيانات إلى عدة صفحات , ونستطيع تحديد عدد الأسطر التي ستظهر في كل صفحة عبر الخاصية PageSize , بالعودة إلى المثال السابق , نقوم بتفعيل الخاصية AllowPaging (يمكن أيضا تفعيلها من خلال Smart Tag عبر الخاصية Enable Paging) , بعد ذلك نقوم بتحديد حجم الصفحة بـ 3 ليتم عرض ثلاثة أسطر فقط في كل صفحة من صفحات GridView كما توضح الصورة التالية :

id	fname	lname	marriage	salary	deptID	photo
2	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
3	إيمان	أحمد	<input type="checkbox"/>	1500	1	
4	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	

12

في حال لم يتم تحديد حجم الصفحة فإن القيمة الافتراضية لها هو 10 أسطر . كود الصفحة يكون كالتالي :

ASP.net كود

```

<div>
  <asp:GridView ID="GridView1" runat="server" AllowPaging="True" PageSize="3"
    DataSourceID="SqlDataSource1" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]" />
</div>

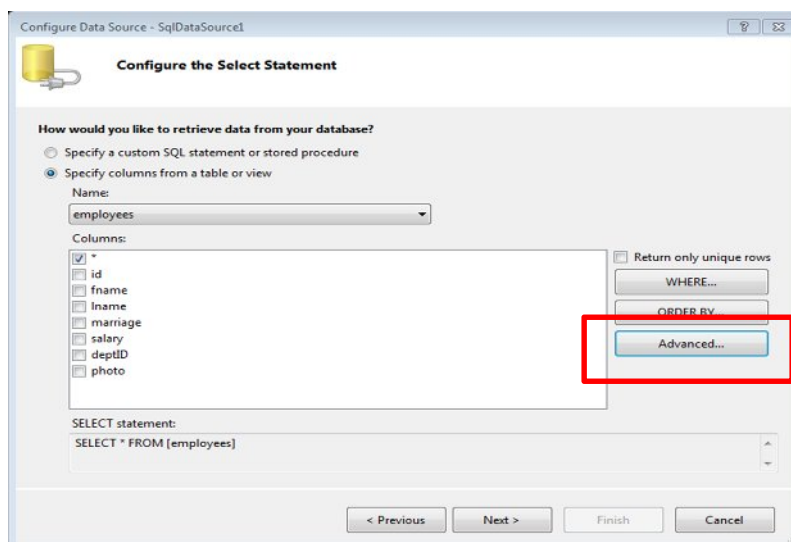
```

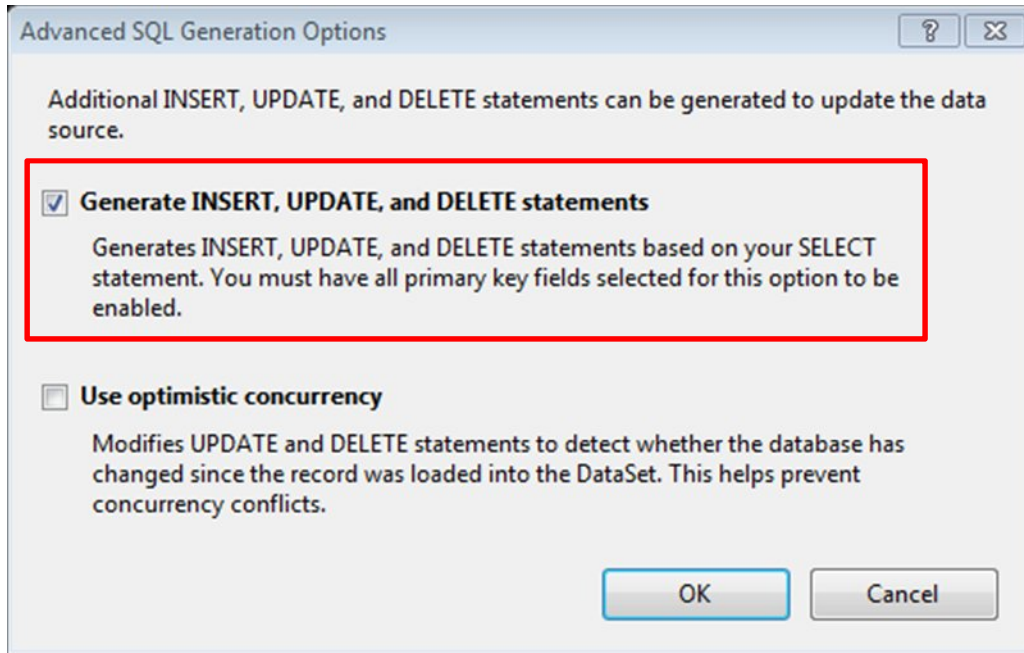
ملاحظة

إن تقسيم البيانات إلى صفحات عديدة كما في الأسلوب السابق سيؤدي إلى بطء تحميل الصفحة عند العمل مع قواعد البيانات الضخمة , وذلك لأنه يتم تحميل كامل سجلات الجدول إلى ذاكرة السيرفر ومن ثم إجراء عملية التقسيم إلى صفحات , فلو تم التعامل مع جدول يحتوي على مليون سطر فإنه سيتم تحميل المليون سطر إلى الذاكرة حتى لو تم عرض سطر واحد فقط في كل صفحة ضمن الأداة GridView , سنعرض في فصل لاحق طريقة أنسب لتقسيم البيانات و ذلك باستخدام الأداة ObjectDataSource .

حذف وتعديل البيانات

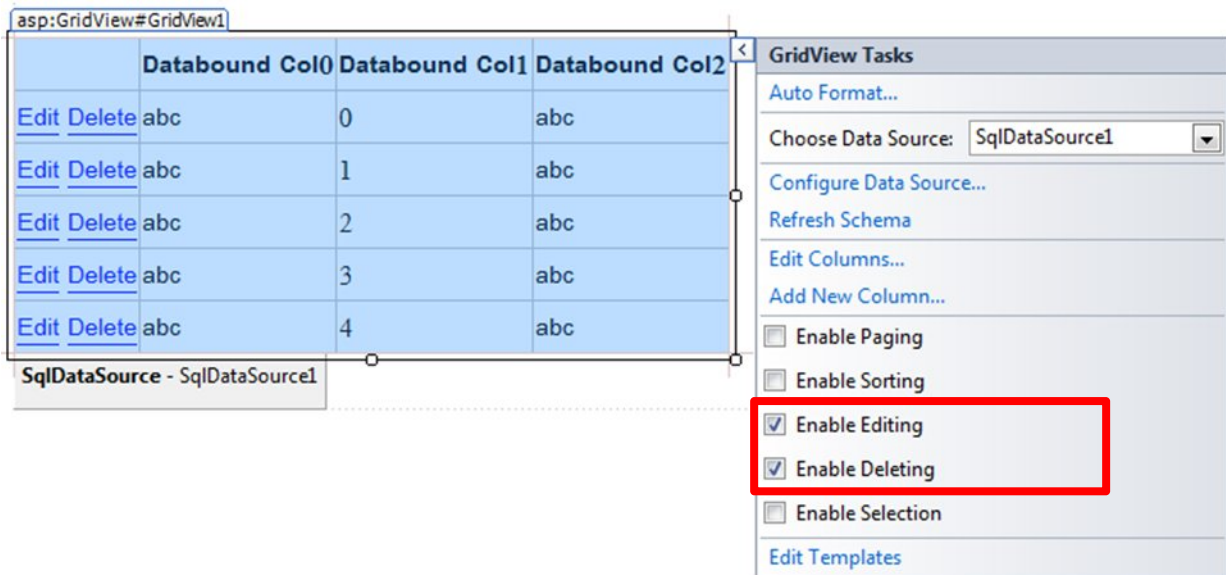
من المزايا الرائعة التي تقدمها الأداة GridView هي إمكانية حذف وتعديل البيانات دون الحاجة لكتابة سطر برمجي واحد , ولكن حتى تتمكن من إجراء هذه العمليات ينبغي أولاً تهيئة الأداة SqlDataSource لتسمح بها , وذلك عبر إضافة عبارات SQL التي تقوم بتنفيذ هذه العمليات (Update, Delete), بكل بساطة أنشئ صفحة جديدة وأضف إليها أداة GridView وابدأ بربطها مع الجدول Employees وعند وصولك للنافذة التالية قم بتفعيل خيار الحذف والتعديل كما هو موضح :





إن تفعيل الخيار الأول يسمح لنا بإجراء عمليات الإضافة التعديل والحذف على الجدول المحدد , شرط أن يحتوي الجدول على مفتاح أساسي , أما الخيار الثاني فهو مفيد لمنع حدوث تضارب في بيانات الجدول اذا تم تعديل نفس السطر من قبل مستخدمين اثنين أو أكثر في نفس الوقت . على العموم بعد الانتهاء من خطوات الربط بشكل كامل نقوم بتفعيل خيارى التعديل والحذف في الأداة GridView حيث نضغط على

Smart Tag ونقوم بتفعيل الخيارين Enable Editing و Enable Deleting



كما أنه بالإمكان تفعيل الخيارين السابقين من خصائص الأداة GridView عبر تفعيل الخاصيتين `AutoGenerateDeleteButton` و `AutoGenerateUpdateButton` وبالتالي يصبح كود الصفحة شبيه بالتالي (لقد قمت بحذف الجزء المتعلق بإمكانية إضافة سطر – insert – إلى الجدول وذلك لأن الأداة GridView لاتسمح بهذه العملية) :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server"
    DataKeyNames="id" DataSourceID="SqlDataSource1"
    AutoGenerateDeleteButton="true" AutoGenerateEditButton="true" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ ConnectionStrings:ConnectionString %>"
    DeleteCommand="DELETE FROM [employees] WHERE [id] = @id"
    SelectCommand="SELECT [id],[fname],[lname] FROM [employees]"
    UpdateCommand="UPDATE [employees] SET [fname] = @fname, [lname] = @lname
      WHERE [id] = @id">
    <DeleteParameters>
      <asp:Parameter Name="id" Type="Int32" />
    </DeleteParameters>
    <UpdateParameters>
      <asp:Parameter Name="fname" Type="String" />
      <asp:Parameter Name="lname" Type="String" />
      <asp:Parameter Name="id" Type="Int32" />
    </UpdateParameters>
  </asp:SqlDataSource>
</div>
```


	id	fname	lname
Edit Delete	2	محمد	الحاج خلف
Edit Delete	3	إيمان	أحمد
Update Cancel	4	<input type="text" value="أحمد"/>	<input type="text" value="المصطفى"/>
Edit Delete	5	بتول	الخلف
Edit Delete	6	قصي	ابراهيم
Edit Delete	7	لين	مراد

لاحظ في الكود السابق وجود الخاصية `DataKeyNames` والتي تحدثنا عنها مسبقا , وقد تم إسناد قيمة المفتاح الأساسي لها (الحقل id) وهذه العملية ضرورية لنتمكن من إجراء عمليات الحذف والتعديل على الجدول .

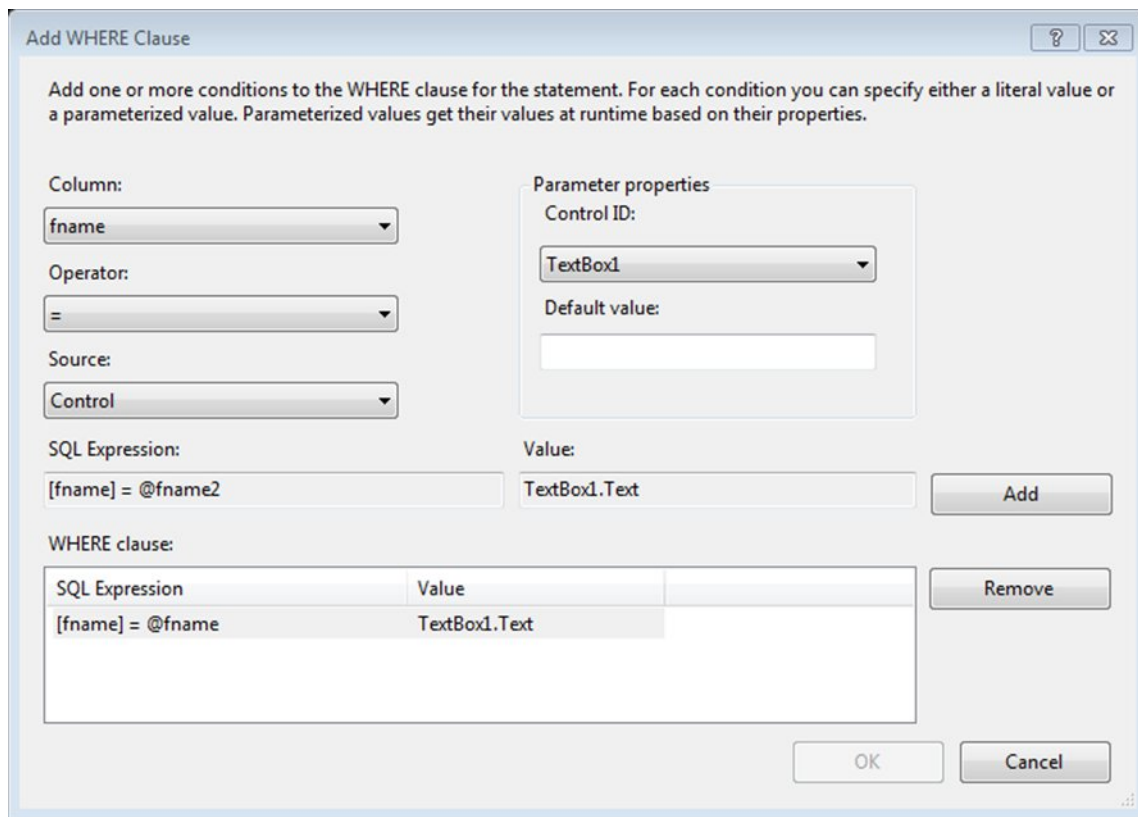
عدم وجود بيانات

بالإمكان تحديد سلوك الأداة GridView عندما لا يكون هناك بيانات مرجعة من مصدر البيانات , كأن يكون الجدول فارغ مثلا , يمكن في هذه الحالة إظهار عبارة للمستخدم تفيد بعدم وجود بيانات وذلك عبر الخاصية EmptyDataText , كما تتوفر خاصية ثانية أكثر غنى وهي EmptyDataTemplate والتي تتيح خيارات أوسع سنعرضها ضمن الأسطر القادمة .

لتوضيح عمل الخاصيتين السابقتين سنقوم ببناء صفحة للبحث عن موظف حسب اسمه , وفي حال عدم وجود موظف بهذا الاسم تقوم الأداة GridView بعرض النص "لا يوجد موظف بهذا الاسم" كما توضح الصورة التالية :



قم بإضافة صفحة جديدة وأضف عليها أداة Button وأداة TextBox وأداة GridView ثم قم بربطها مع جدول الموظفين Employees وأضف شرط أن يقوم بعرض الموظفين الذين لهم اسم كالنص المكتوب في الأداة TextBox كما في الصورة التالية :



نختار الأداة GridView1 و نقوم بإسناد القيمة " لا يوجد موظفين بهذا الاسم " إلى الخاصية EmptyDataText , والقيمة false للخاصية Visible , ثم نقوم بإضافة الكود التالي في حدث النقر على الزر search :

كود C#

```
GridView1.Visible = true;
```

كود VB

```
GridView1.Visible = true
```

أما كود الصفحة فيكون بالشكل التالي :

كود ASP.net

```
<div>
  <asp:TextBox ID="TextBox1" runat="server" />
  <asp:Button ID="Button1" runat="server" Text="Search" onclick="Button1_Click" />
  <hr />
  <asp:GridView ID="GridView1" runat="server" Visible="false"
    DataKeyNames="id" DataSourceID="SqlDataSource1"
    EmptyDataText="الاسم بهذا موظف يوجد لا" />

  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees] WHERE ([fname] = @fname)" >
    <SelectParameters>
      <asp:ControlParameter ControlID="TextBox1" Name="fname"
        PropertyName="Text" Type="String" />
    </SelectParameters>
  </asp:SqlDataSource>
</div>
```

تنسيق مظهر الأداة GridView

Header		
المعرف	الاسم	الراتب
1	محمد	1000
2	إيمان	1500
3	أحمد	1200
4	بتول	1300
Pager		
Footer		
1 2		

تحتوي الأداة GridView على مجموعة ضخمة من الخصائص التي تفيد بتنسيق مظهر هذه الأداة ومنها :

- ✓ HeaderStyle : تمكنا من تنسيق مظهر سطر عناوين الأعمدة .
- ✓ FooterStyle : تمكنا من تنسيق مظهر السطر السفلي (بشكل افتراضي يكون هذا السطر غير ظاهر).
- ✓ RowStyle : تمكنا من تنسيق مظهر جميع أسطر البيانات .
- ✓ AlternatingRowStyle : تمكنا من تنسيق مظهر أسطر البيانات الزوجية , وعند تحديد قيم لهذه الخاصية فإن هذه الأسطر الزوجية ستتجاهل القيم المحددة عبر الخاصية RowStyle .
- ✓ PagerStyle : تمكنا من تنسيق مظهر سطر التنقل بين صفحات الأداة GridView وذلك عند تفعيل الخاصية AllowPaging .
- ✓ SelectedRowStyle : تمكنا من تنسيق مظهر مختلف للسطر الحالي الذي تم اختياره .

جميع الخصائص السابقة تحتوي على الخاصية CssClass والتي تتيح إمكانية تحديد التنسيق ضمن كود CSS ومن ثم يتم تحديد اسم الصف المراد تطبيقه .

مثال :

يعرض هذا المثال كيفية تنسيق مظهر الأداة GridView عبر صفوف CSS والخاصية CssClass .

كود ASP.net

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="GridViewFormatting.aspx.cs"
    Inherits="GridViewFormatting" %>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>GridView Formatting</title>
    <style type="text/css">
        .grid
        {   font:16px Arial, Sans-Serif;   }
        .grid td, .grid th
        {   padding:10px;   }
        .header
        {   text-align:left;
            color:white;
            background-color:blue;
        }
        .row td
        {   border-bottom:solid 1px blue;   }
        .alternating
        {   background-color:#eeeeee;   }
        .alternating td
        {   border-bottom:solid 1px blue;   }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div>

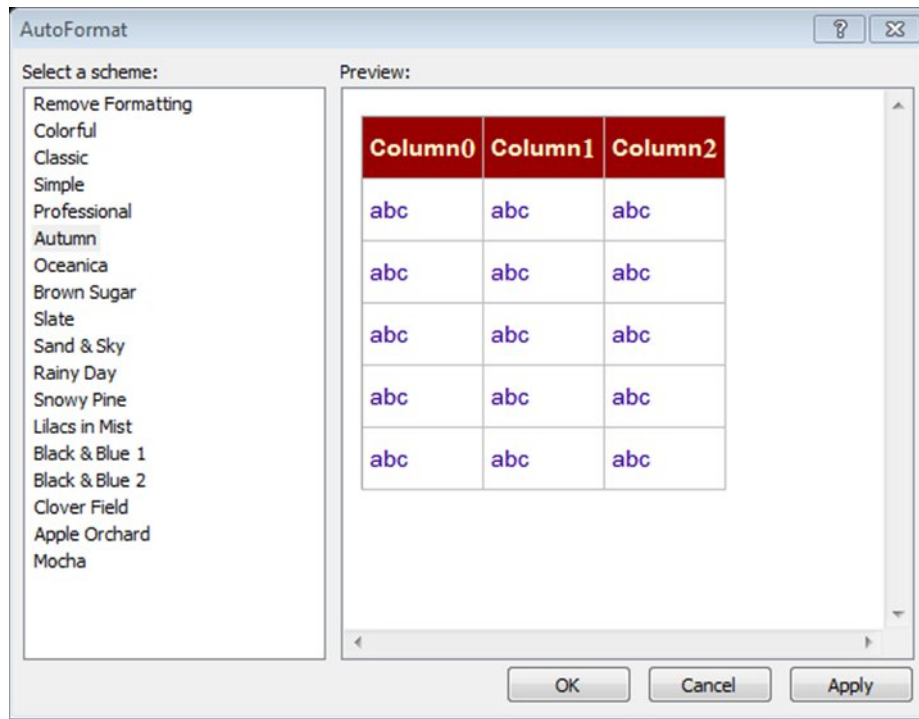
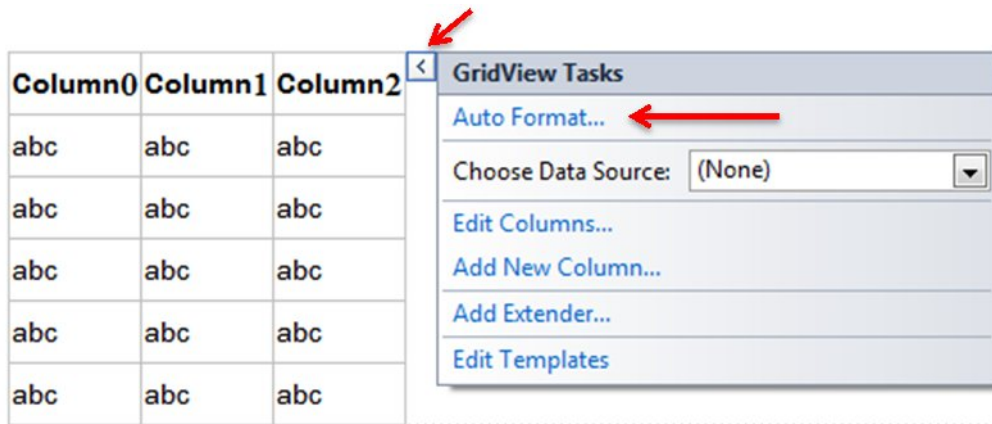
        <asp:GridView ID="GridView1" runat="server" AllowPaging="True" PageSize="5"
            DataSourceID="SqlDataSource1" ShowFooter="True"
            CssClass="grid"
            HeaderStyle-CssClass="header"
            RowStyle-CssClass="row"
            AlternatingRowStyle-CssClass="alternating" />

        <asp:SqlDataSource ID="SqlDataSource1" runat="server"
            ConnectionString="<%"$ ConnectionStrings:MyConnectionString %">"
            SelectCommand="SELECT [Id], [Name], [Mark] FROM [MyTable]" />

    </div>
    </form>
</body>
</html>

```

كما توجد مجموعة من التنسيقات الجاهزة والتي يمكن تطبيقها بكل سهولة , حيث نضغط بالبداية على الـ smart tag للأداة GridView ثم نختار Auto Format ستظهر نافذة تحتوي على مجموعة التنسيقات نختار منها مايناسبنا لتطبيقه , كما نستطيع التعديل على هذه التنسيقات إن أردنا ذلك .



أما بالنسبة للسطرين Header , Footer فيمكننا التحكم في ظهورهما أو إخفائهما وذلك عبر الخاصيتين ShowHeader , ShowFooter على التوالي .

تخصيص الحقول في الأداة GridView

في جميع الفقرات السابقة كنا ندع الأداة GridView تقوم بتوليد الحقول وعنوانتها بشكل تلقائي وبدون تدخل من قبلنا , ولكن هذا ليس بالأسلوب الأفضل وقد لا يلبي حاجات المبرمج , حيث إن عناوين الحقول تأخذ قيمها كأسماء الحقول في جدول قاعدة البيانات وهذا أمر غير مستحسن , كما أن البيانات تعرض دائما على شكل نص فماذا لو أردنا أن يتم عرض صورة ؟ كيف نستطيع إضافة أداة DropDownList داخل الأداة GridView , كيف نضيف أدوات التحقق إلى الأداة GridView أثناء عملية التعديل ؟ كل هذه الأسئلة بالإضافة إلى أمور أخرى ستتم مناقشتها ضمن هذه الفقرة والفقرات القادمة .

تدعم الأداة GridView سبعة أنماط للحقول وهي :

- ✓ BoundField : تمكنا من عرض البيانات كنص عادي .
- ✓ CheckBoxField : تمكنا من عرض البيانات كأداة اختيار CheckBox .
- ✓ ButtonField : تمكنا من عرض البيانات كزر (Button, LinkButton, ImageButton).
- ✓ HyperLinkField : تمكنا من عرض البيانات كرابط تشعبي .
- ✓ ImageField : تمكنا من عرض البيانات كصورة .
- ✓ TemplateField : تمكنا من تخصيص أسلوب عرض البيانات وفق أي شكل نرغب به .
- ✓ CommandField : تمكنا من إظهار روابط لعمليات الاختيار والحذف والتعديل .

لكل نمط من الأنماط السابقة مجموعة كبيرة من الخصائص , سنقوم بشرحها مع تطبيق أمثلة على كل منها ضمن الفقرات التالية .

استخدام نمط الحقول BoundFiled

وهو أبسط نمط من أنماط الحقول , حيث يتم عرض البيانات كنص عادي (وهو النمط الذي نشاهده حتى الآن) , وفي أسلوب التعديل يتم عرض البيانات ضمن أدوات TextBox في نمط SingleLine .

خصائص النمط BoundField :

- ✓ DataField : تسمح بتخصيص اسم برمجي للحقل .
- ✓ DataFormatString : تسمح بتطبيق تنسيق على البيانات المعروضة . كتتنسيق العملات أو أرقام الهواتف وغيره .
- ✓ ApplyFormatInEditMode : تسمح بتطبيق تنسيق معين على النص في حالة التعديل حيث تستخدم DataFormatString .
- ✓ ShowHeader : لعرض أو إخفاء عنوان الحقل .
- ✓ HeaderText : تسمح بتخصيص عنوان للحقل .
- ✓ HeaderStyle : تسمح بتطبيق تنسيق على عنوان الحقل .
- ✓ HeaderImageUrl : تمكنا من عرض صورة كعنوان للحقل بدلا من النص .
- ✓ HtmlEncode : لتحديد إمكانية عرض تنسيقات HTML أم لا , حيث تكون هذه التنسيقات مخزنة مع النص في قاعدة البيانات .
- ✓ ItemStyle : تسمح بتحديد تنسيق على البيانات المعروضة , كحجم الخط ولونه وغيره .
- ✓ NullDisplayText : تمكنا من تحديد نص بديل ليتم عرضه في حال كانت القيمة المرجعة هي Null .
- ✓ ReadOnly : لمنع إمكانية تعديل قيمة الحقل حتى في حالة التعديل (مثل حقل المفتاح الرئيسي) .
- ✓ SortExpression : تسمح بإمكانية ترتيب بيانات الجدول وفق هذا الحقل .

✓ Visible : لعرض أو إخفاء هذا الحقل .

نلاحظ أن معظم هذه الخصائص تتعلق بمظهر البيانات وليست خصائص ذات طابع برمجي .

مثال :

افتح الجدول employees ضمن قاعدة البيانات وعدل قيمة الحقل salary إلى NULL لأحد الموظفين لتصبح بيانات الجدول مشابهة للتالي :

id	fname	lname	marriage	salary	deptId	photo
1	محمد	الحاج خلف	False	1000	1	~/images/1.jpg
2	إيمان	أحمد	False	1500	1	~/images/2.jpg
3	أحمد	المصطفى	True	1200	2	~/images/3.jpg
4	بتول	الخلف	False	1300	2	~/images/4.jpg
5	قصي	ابراهيم	True	900	3	~/images/5.jpg
6	لين	مراد	True	NULL	3	~/images/6.jpg

المطلوب هو عرض بيانات الجدول السابق ضمن متصفح الانترنت بالأسلوب التالي :

الرقم	الاسم	الكنية	متزوج؟	الراتب	القسم	صورة شخصية
1	محمد	الحاج خلف	<input type="checkbox"/>	\$1,000.00	1	~/images/1.jpg
2	أحمد	إيمان	<input type="checkbox"/>	\$1,500.00	1	~/images/2.jpg
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	\$1,200.00	2	~/images/3.jpg
4	بتول	الخلف	<input type="checkbox"/>	\$1,300.00	2	~/images/4.jpg
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	\$900.00	3	~/images/5.jpg
6	لين	مراد	<input checked="" type="checkbox"/>	غير محدد	3	~/images/6.jpg

حيث تم إعطاء الحقول عناوين مختلفة عما هي في قاعدة البيانات , بالإضافة إلى أن قيم الحقل salary قد أخذت تنسيق العملة وتم ظهور العبارة "غير محدد" بدلا من NULL , أما كود الصفحة فهو :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="الرقم" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="الاسم"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="الكنية"
        SortExpression="lname" />
    </Columns>
  </asp:GridView>
</div>
```

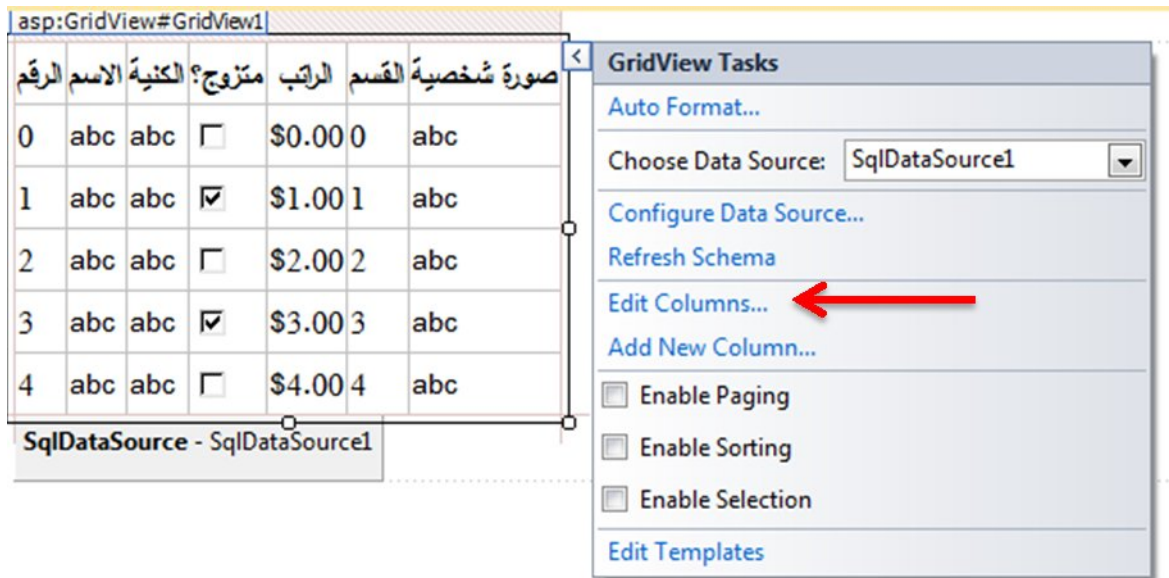
```

<asp:CheckBoxField DataField="marriage" HeaderText="متزوج?"
SortExpression="marriage" />
<asp:BoundField DataField="salary" HeaderText="الراتب"
SortExpression="salary" DataFormatString="{0:c}"
NullDisplayText="غير محدد" />
<asp:BoundField DataField="deptId" HeaderText="القسم"
SortExpression="deptId" />
<asp:BoundField DataField="photo" HeaderText="صورة شخصية"
SortExpression="photo" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%= ConnectionStrings:ConnectionString %>"
SelectCommand="SELECT * FROM [employees]"></asp:SqlDataSource>
</div>

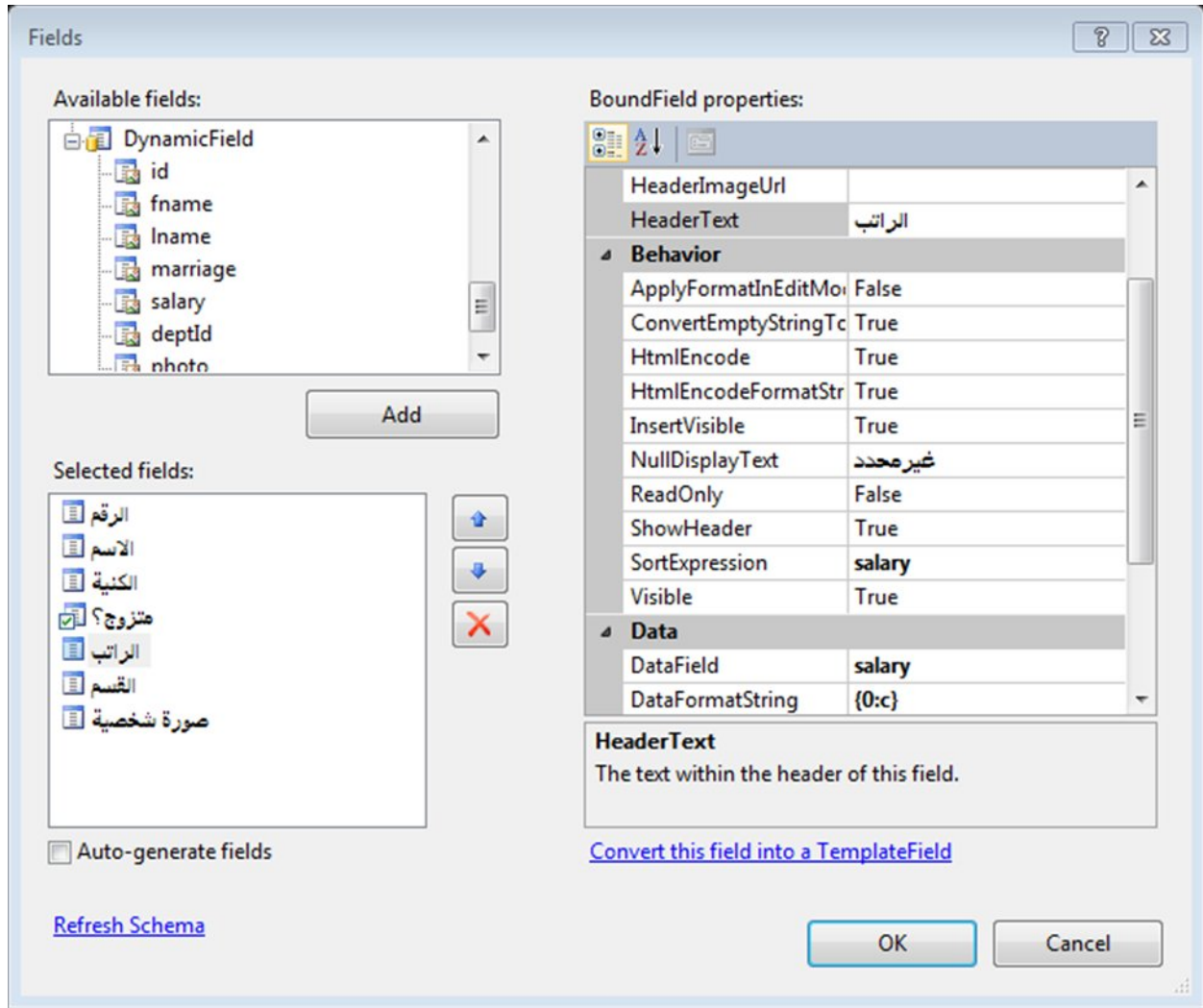
```

توفر لنا منصة العمل Visual Studio أسلوب سهل جدا للتعديل على خصائص الحقول , فأنت عزيزي المبرمج غير مضطر لكتابة الخصائص السابقة بشكل يدوي , إنما بالإمكان الاستعانة بالنافذة Fields لفعل ذلك عبر الخطوات التالية :

ضمن الصفحة التي نعمل عليها انتقل إلى وضع التصميم و انقر على الأداة GridView التي أضفناها مسبقا , سيظهر سهم صغير (يدعى smart tag) انقر عليه و اختر Edit Columns



ستفتح الآن نافذة خصائص الحقول والتي تدعى Fields , قم بتحديد الحقل salary ولاحظ ظهور جميع خصائصه مع القيم الجديدة التي عدلنا عليها , كما توضح الصورة التالية :



إن هذه النافذة تسهل التعامل مع خصائص الحقول ولكن لا تعتمد عليها بالكامل وتنسى أسلوب التعديل على الخصائص من خلال الكود بشكل مباشر .

استخدام نمط الحقول CheckBoxFields

يستخدم هذا النمط مع بيانات الحقول التي تدعم قيمتين فقط كالقيم المنطقية True , False حيث يتم عرضها على شكل أداة التحكم CheckBox .

إن الصف CheckBoxField يرث من الصف BoundField وهذا يعني أن النمط CheckBoxField يدعم جميع الخصائص المدعومة في النمط BoundField والتي ناقشناها مسبقا , بالإضافة إلى الخاصية التالية :

✓ Text : تسمح بإظهار نص بجانب كل CheckBox ضمن الحقل .

خلال المثال الذي قمنا بتطبيقه ضمن الفقرة السابقة نجد أن الحقل "متزوج؟" (marriage) هو من النمط CheckBoxField (علما أننا لم نقم بتحديد هذا النمط !) بشكل تلقائي فإن جميع الحقول التي لها نمط البيانات Boolean ضمن قاعدة البيانات يتم إظهارها على شكل حقل من النمط CheckBoxField ضمن أدوات عرض البيانات .

صورة شخصية	القسم	الراتب	متزوج؟	الكنية	الاسم الرقم
~/images/1.jpg	1	\$1,000.00	<input type="checkbox"/>	الحاج خلف	محمد 1
~/images/2.jpg	1	\$1,500.00	<input type="checkbox"/>	أحمد إيمان	2
~/images/3.jpg	2	\$1,200.00	<input checked="" type="checkbox"/>	المصطفى أحمد	3
~/images/4.jpg	2	\$1,300.00	<input type="checkbox"/>	الخلف بتول	4
~/images/5.jpg	3	\$900.00	<input checked="" type="checkbox"/>	ابراهيم قصي	5
~/images/6.jpg	3	غير محدد	<input checked="" type="checkbox"/>	مراد لين	6

ولو راجعنا كود الصفحة سنجد هذا النمط مستخدم مع الحقل "marriage" :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="الرقم" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="الاسم"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="الكنية"
        SortExpression="lname" />
      <asp:CheckBoxField DataField="marriage" HeaderText="متزوج؟"
        SortExpression="marriage" />
      <asp:BoundField DataField="salary" HeaderText="الراتب"
        SortExpression="salary" DataFormatString="{0:c}"
        NullDisplayText="غير محدد" />
      <asp:BoundField DataField="deptId" HeaderText="القسم"
        SortExpression="deptId" />
      <asp:BoundField DataField="photo" HeaderText="صورة شخصية"
        SortExpression="photo" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]"></asp:SqlDataSource>
</div>
```

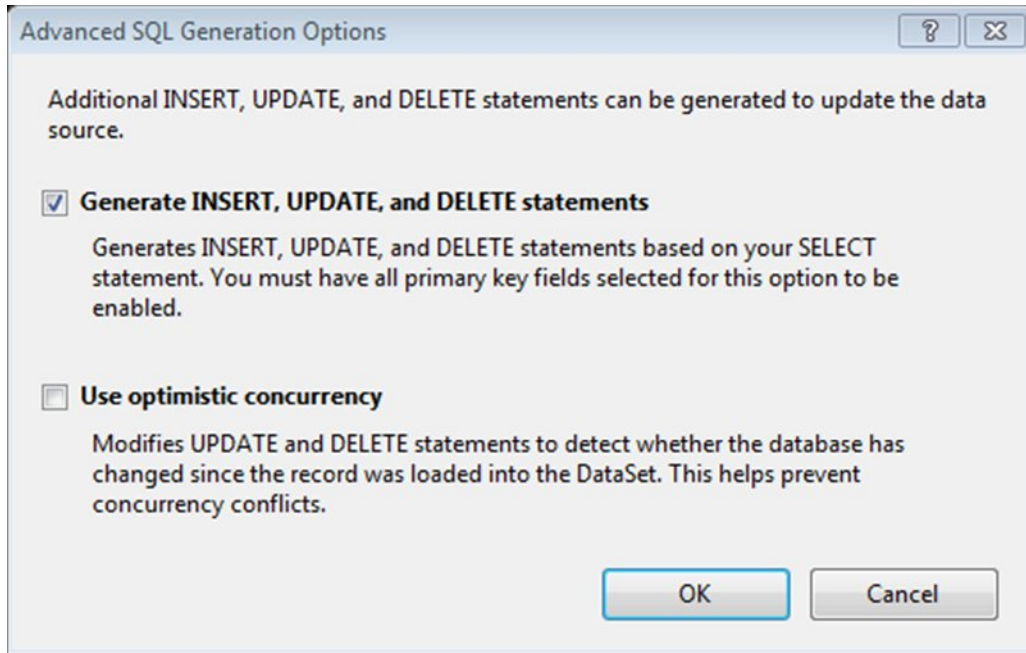
نلاحظ أن الأداة CheckBox تكون في حالة Disable في وضعية العرض , أما في وضعية التعديل فيتم تفعيلها وبالتالي يمكن تغيير القيمة المقابلة لها ضمن جدول قاعدة البيانات .

استخدام CommandFields

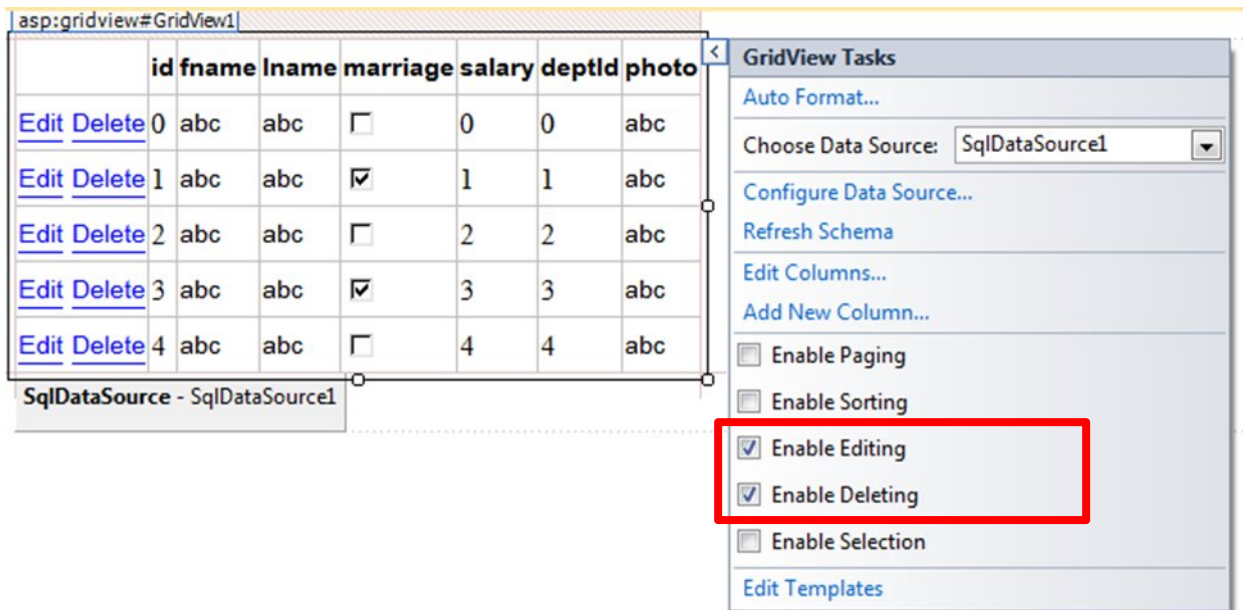
نستخدم نمط الحقول CommandFields لتخصيص مظهر أزرار الاختيار والتعديل والتحديث والحذف وإلغاء الأمر (Edit, Delete, Update, Cancel, and Select) ويتم هذا في حال تحديد ظهور هذه الخيارات , حيث تكون غير ظاهرة في الحالة الافتراضية .

سنقوم بالبداية بتطبيق مثال يوضح عمل هذا النمط من الحقول ثم نستعرض بعد ذلك مجموعة الخصائص المرتبطة به .

قم بإنشاء صفحة جديدة وأضف أداة GridView و اربطها مع قاعدة و قم بتفعيل خيارات الإضافة والحذف والتعديل عند ضبط إعدادات الاتصال بالجدول (تم توضيح هذه الخطوات ضمن فصل التعامل مع الأداة SqlDataSource) كما توضح الصور التالية :



بعد الانتهاء من إعداد باقي خطوات الاتصال بالجدول , نضغط على smart tag ثم نفعّل الخيارين Enable Deleting و Enable Editing



بالإمكان الآن تشغيل الصفحة السابقة وإجراء عمليات التعديل والحذف على البيانات وستكون الصفحة كالتالي :

	id	fname	lname	marriage	salary	deptId	photo
Edit Delete	1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	~/images/1.jpg
Edit Delete	2	إيمان	أحمد	<input type="checkbox"/>	1500	1	~/images/2.jpg
Edit Delete	3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	~/images/3.jpg
Edit Delete	4	بتول	الخلف	<input type="checkbox"/>	1300	2	~/images/4.jpg
Edit Delete	5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3	~/images/5.jpg
Update Cancel	6	لين	مراد	<input checked="" type="checkbox"/>		3	~/images/6.jpg

أما كود الصفحة فيكون كالتالي :

ASP.net كود

```
<div>
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataKeyNames="id" DataSourceID="SqlDataSource1">
<Columns>
<asp:CommandField ShowDeleteButton="True" ShowEditButton="True" />
<asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
ReadOnly="True" SortExpression="id" />
<asp:BoundField DataField="fname" HeaderText="fname"
SortExpression="fname" />
<asp:BoundField DataField="lname" HeaderText="lname"
SortExpression="lname" />
<asp:CheckBoxField DataField="marriage" HeaderText="marriage"
SortExpression="marriage" />
<asp:BoundField DataField="salary" HeaderText="salary"
SortExpression="salary" />
<asp:BoundField DataField="deptId" HeaderText="deptId"
SortExpression="deptId" />
<asp:BoundField DataField="photo" HeaderText="photo"
SortExpression="photo" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
DeleteCommand="DELETE FROM [employees] WHERE [id] = @id"
SelectCommand="SELECT * FROM [employees]"
UpdateCommand="UPDATE [employees] SET [fname] = @fname, [lname] = @lname,
[marriage] = @marriage, [salary] = @salary,
[deptId] = @deptId, [photo] = @photo WHERE [id] = @id" >
<DeleteParameters>
<asp:Parameter Name="id" Type="Int32" />
</DeleteParameters>
<UpdateParameters>
<asp:Parameter Name="fname" Type="String" />
<asp:Parameter Name="lname" Type="String" />
<asp:Parameter Name="marriage" Type="Boolean" />
<asp:Parameter Name="salary" Type="Int32" />
<asp:Parameter Name="deptId" Type="Int32" />
</UpdateParameters>
</asp:SqlDataSource>
</div>
```

```

        <asp:Parameter Name="photo" Type="String" />
        <asp:Parameter Name="id" Type="Int32" />
    </UpdateParameters>
</asp:SqlDataSource>

</div>

```

لاحظ بأنني قمت في الكود السابق بحذف عبارة الإضافة Insert ضمن الأداة SqlDataSource1 وذلك لأن الأداة GridView لا تدعم هذه العملية بشكلها الافتراضي .

في الكود السابق فإن نمط الحقول CommandField هو الذي أتاح لنا الزرين Edit , Delete وذلك عبر تفعيل الخاصيتين ShowDeleteButton و ShowEditButton .

نمط الحقول CommandFields يدعم الخصائص التالية :

- ✓ ButtonType : لتحديد نوع الأداة التي ستظهر ضمن هذا الحقل ويمكن أن تأخذ إحدى القيم Button, Image, Link .
- ✓ ShowSelectButton : يستخدم لعرض زر الاختيار Select .
- ✓ SelectText : لتحديد نص يظهر على زر الاختيار .
- ✓ SelectImageUrl : لتحديد صورة تظهر كزر الاختيار .
- ✓ ShowEditButton : يستخدم لعرض زر التعديل Edit .
- ✓ EditText : لتحديد نص يظهر على زر التعديل .
- ✓ EditImageUrl : لتحديد صورة تظهر كزر تعديل .
- ✓ ShowDeleteButton : يستخدم لعرض زر الحذف Delete .
- ✓ DeleteText : لتحديد نص يظهر على زر الحذف .
- ✓ DeleteImageUrl : لتحديد صورة تظهر كزر حذف .
- ✓ ShowInsertButton : يستخدم لعرض زر الإضافة Insert .
- ✓ InsertText : لتحديد نص يظهر على زر الإضافة .
- ✓ InsertImageUrl : لتحديد صورة تظهر كزر إضافة .
- ✓ ShowCancelButton : يستخدم لعرض زر إلغاء الأمر Cancel .
- ✓ CancelText : لتحديد نص يظهر على زر إلغاء الأمر .
- ✓ CancelImageUrl : لتحديد صورة تظهر كزر إلغاء الأمر .
- ✓ CausesValidation : تمكنا من تجاوز أدوات التحقق عند الضغط على زر التعديل Edit .
- ✓ ValidationGroup : تمكنا من ربط زر التعديل مع مجموعة أدوات تحقق وذلك لنضمن أن البيانات الجديدة تخضع لشروط محددة .

بعد أن تعرفنا على خصائص نمط الحقول CommandField فإنه بإمكاننا التعديل على هذه الخصائص بحيث نصل إلى نتيجة مشابهة للتالي :

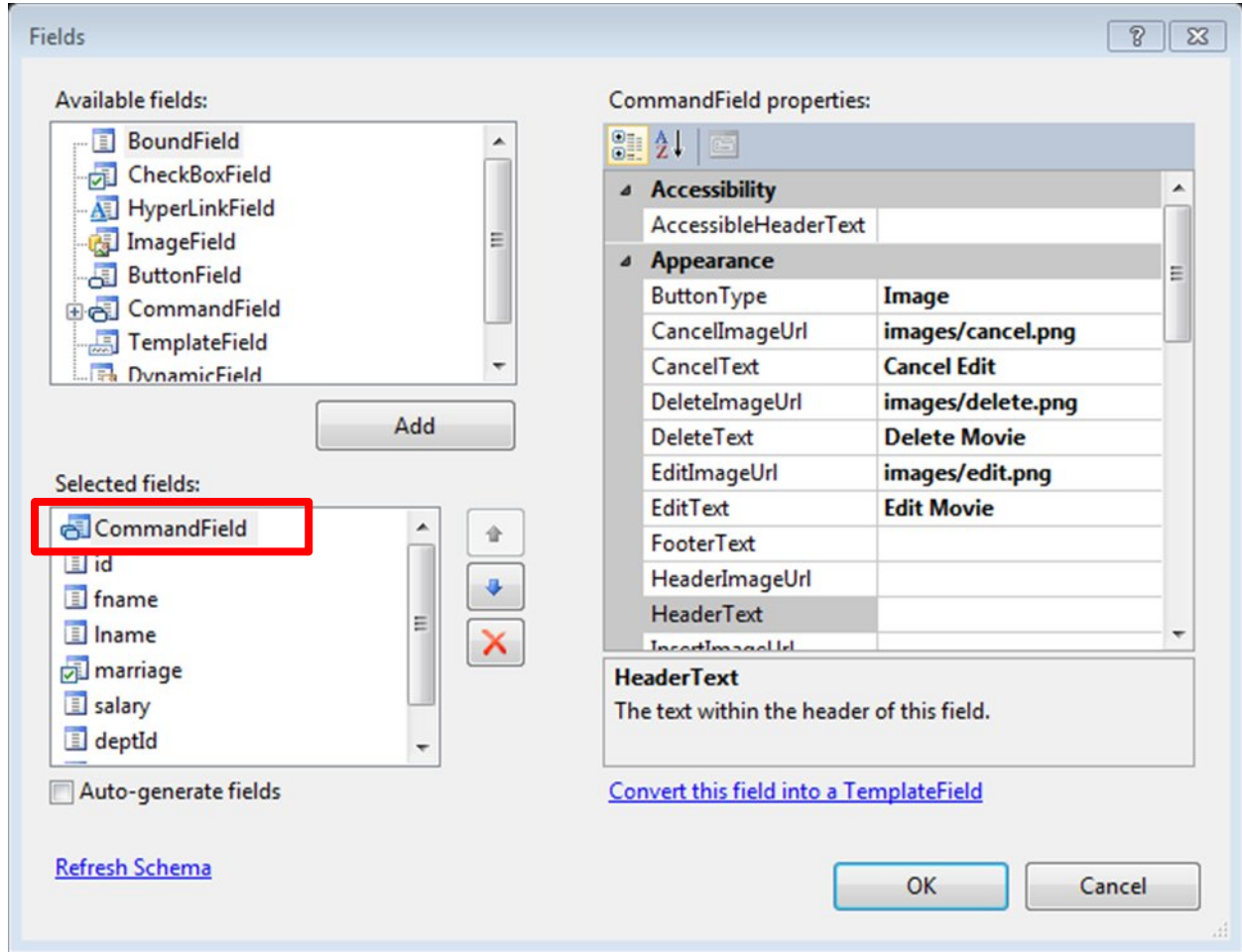
	id	fname	lname	marriage
 	1	محمد	الحاج خلف	<input type="checkbox"/>
 	2	إيمان	أحمد	<input type="checkbox"/>
 	3	أحمد	المصطفى	<input checked="" type="checkbox"/>
 	4	بتول	الخلف	<input type="checkbox"/>
 	5	قصي	ابراهيم	<input checked="" type="checkbox"/>
 	6	<input type="text" value="لين"/>	<input type="text" value="مراد"/>	<input checked="" type="checkbox"/>

أما الخصائص المعدلة فتظهر في الكود التالي :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:CommandField
        ButtonType="Image"
        ShowEditButton="true"
        EditText="Edit Movie"
        EditImageUrl="edit.png"
        UpdateText="Update Movie"
        UpdateImageUrl="save.png"
        ShowCancelButton="true"
        CancelText="Cancel Edit"
        CancelImageUrl="cancel.png"
        ShowDeleteButton="true"
        DeleteText="Delete Movie"
        DeleteImageUrl="delete.png"
      />
    :
    .
  </asp:GridView>
</div>
```

حيث يمكن الوصول إلى هذه الخصائص من خلال النقر على smart tag ثم نختار Edit Columns ثم نحدد الحقل CommandField



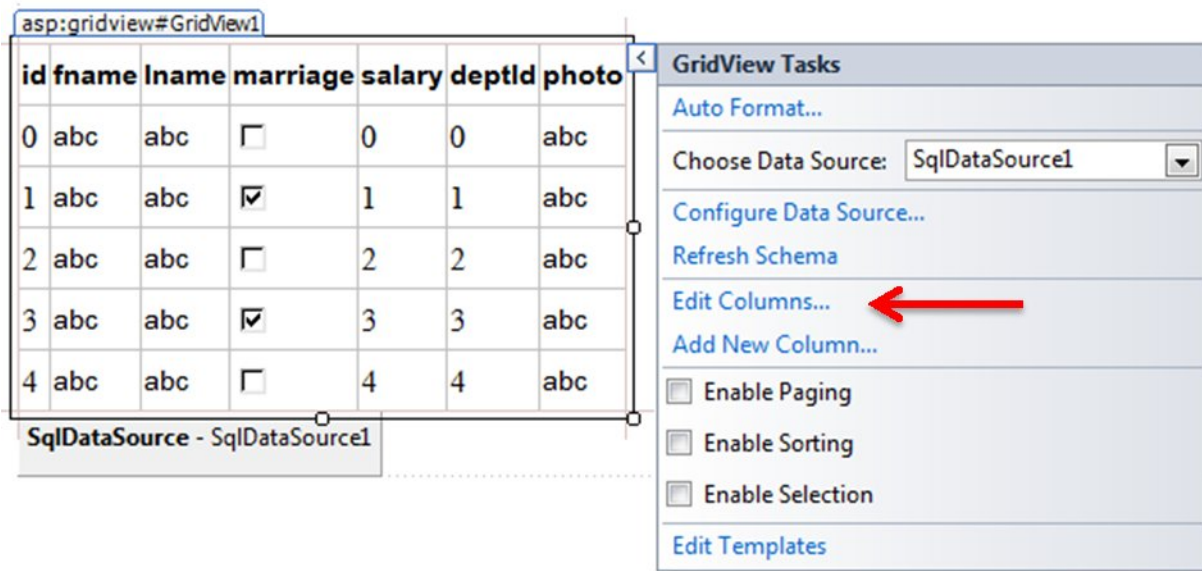
استخدام نمط الحقول ImageField

نستخدم نمط الحقول ImageField لعرض الصور داخل الأداة GridView حيث تكون موجودة على السيرفر و مسارات هذه الصور تكون مخزنة ضمن جدول ما في قاعدة البيانات .

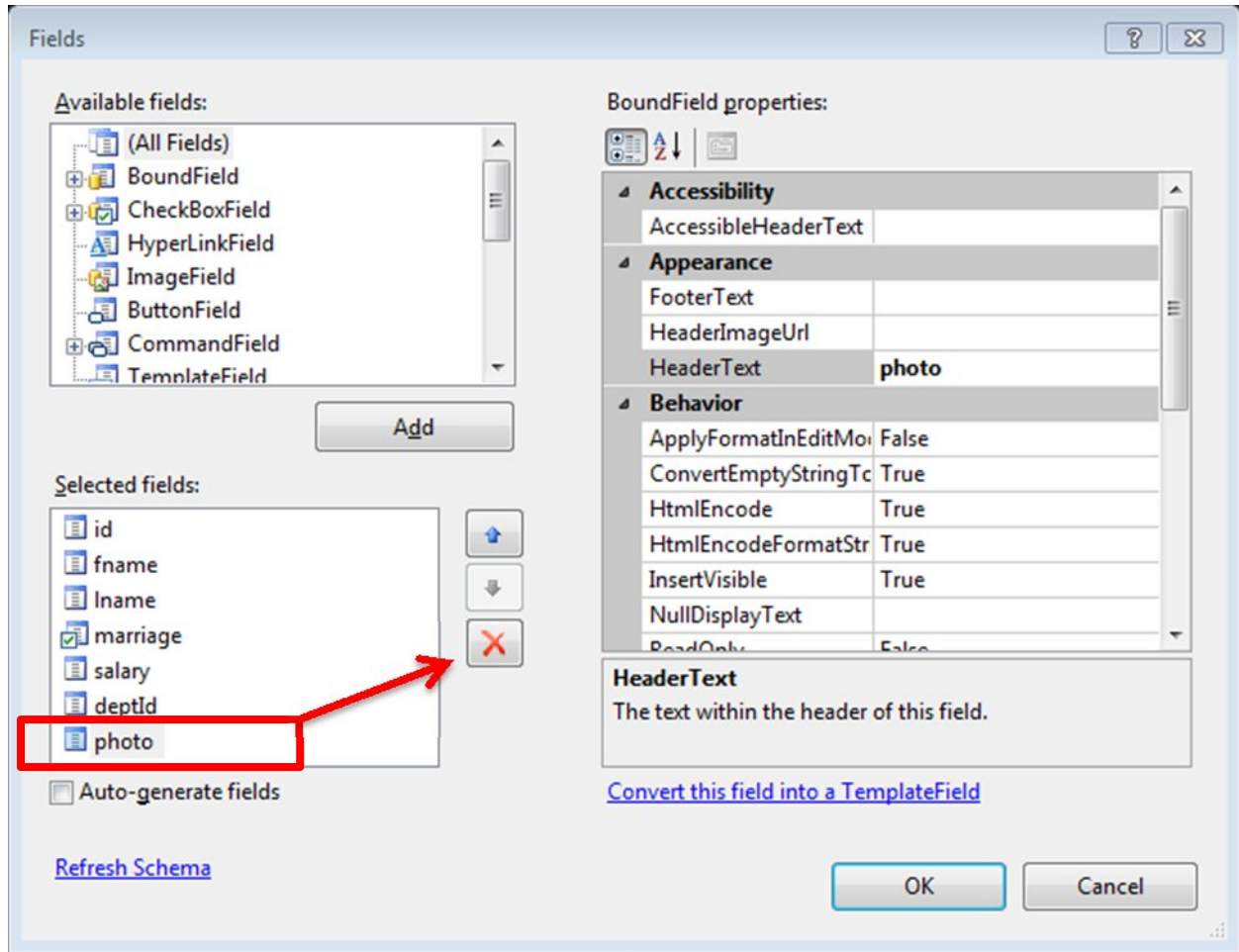
لنقم بداية بإنشاء صفحة جديدة و إضافة أداة GridView وربطها مع الجدول Employee حيث نقوم بإحضار وعرض كافة الحقول , قم بتنفيذ هذه الصفحة ضمن المتصفح , ستحصل على النتيجة التالية :

id	fname	lname	marriage	salary	deptId	photo
1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	~/images/1.jpg
2	إيمان	أحمد	<input type="checkbox"/>	1500	1	~/images/2.jpg
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	~/images/3.jpg
4	بتول	الخلف	<input type="checkbox"/>	1300	2	~/images/4.jpg
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3	~/images/5.jpg
6	لين	مراد	<input checked="" type="checkbox"/>	800	3	~/images/6.jpg

نلاحظ القيم المخزنة ضمن العمود photo وهي مسارات الصور ضمن المشروع الحالي , مانريده فعليا هو عرض هذه الصور وليس عرض مساراتها ! وهنا يأتي دور نمط الحقول ImageField , اضغط على smart tag ضمن الـ GridView و اختر Edit Columns

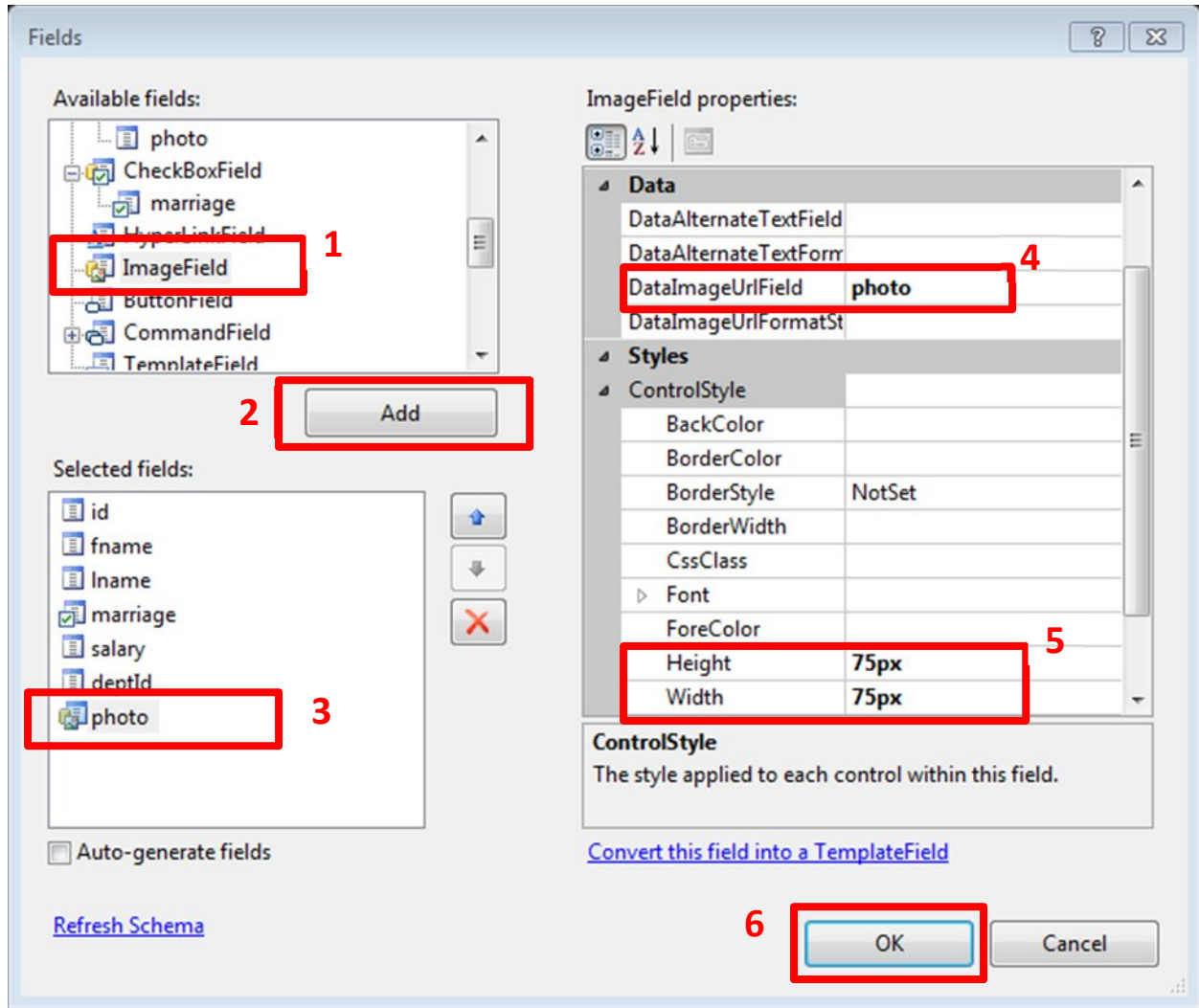


ستظهر النافذة Fileds , نقوم بحذف الحقل photo (لأنه بوضعه الافتراضي يكون من النمط BoundField وهو غير قادر على عرض الصور)



نقوم بعد ذلك بإضافة حقل جديد من النمط ImageField ونقوم بضبط بعض خصائصه كالتالي :

DataImageUrlField نعطيها القيمة photo (وهنا تكمن عملية الربط الفعلية مع مصدر البيانات الذي سيحدد مسار الصور) , كما نقوم بضبط أبعاد الصور التي سيتم عرضها حيث قمنا بجعلها 75 * 75 بيكسل , تظهر هذه الإعدادات في الشكل التالي :



قم الآن بتنفيذ الصفحة السابقة , وستظهر النتيجة التالية :

id	fname	lname	marriage	salary	deptId	
1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
2	إيمان	أحمد	<input type="checkbox"/>	1500	1	
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2	
4	بتول	الخلف	<input type="checkbox"/>	1300	2	
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3	
6	لين	مراد	<input checked="" type="checkbox"/>	800	3	

تم ظهور الصور وهذا مانريده ولكن نلاحظ عدم ظهور عنوان للحقل الجديد ! نعود إلى النافذة Fileds السابقة , نحدد الحقل الجديد الذي قمنا بإضافته و نعطي الخاصية HeaderText القيمة photo أو أي عنوان مناسب لك . بإمكانك عزيزي القارئ استكشاف باقي الخصائص ووظيفة كل منها , حيث سأكتفي ضمن الصفحات القادمة بذكر الخصائص الهامة فقط لهذا النمط من الحقول .

كود الصفحة :

كود ASP.net

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
  <Columns>
    <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
      ReadOnly="True" SortExpression="id" />
    <asp:BoundField DataField="fname" HeaderText="fname"
      SortExpression="fname" />
    <asp:BoundField DataField="lname" HeaderText="lname"
      SortExpression="lname" />
```

```

<asp:CheckBoxField DataField="marriage" HeaderText="marriage"
SortExpression="marriage" />
<asp:BoundField DataField="salary" HeaderText="salary"
SortExpression="salary" />
<asp:BoundField DataField="deptId" HeaderText="deptId"
SortExpression="deptId" />
<asp:ImageField DataImageUrlField="photo" HeaderText="photo">
<ControlStyle Height="75px" Width="75px" />
</asp:ImageField>
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
SelectCommand="SELECT * FROM [employees]">
</asp:SqlDataSource>

```

نمط الحقول ImageField يدعم الخصائص التالية :


- ✓ AlternateText : لتحديد نص بديل يتم عرضه في حال عدم ظهور الصورة
- ✓ DataAlternateTextField : لتحديد اسم حقل ضمن جدول في قاعدة البيانات يتم جلب النص البديل منه في حال عدم ظهور الصورة
- ✓ DataAlternateTextFormatString : لتطبيق تنسيق ما على النص البديل
- ✓ DataImageUrlField : لتحديد اسم حقل ضمن مصدر البيانات بحيث يحتوي على مسار الصورة
- ✓ NullImageUrl : لتحديد مسار لصورة بديلة يتم عرضها في حال كانت القيمة المحضرة من مصدر البيانات للخاصية DataImageUrlField تساوي NULL




ملاحظة

إن الصور التي يتم إنشاؤها سيتم ترجمتها نهاية المطاف إلى وسوم بلغة HTML وبالتالي لا يخفى على أحد أهمية وجود النص البديل alt="" , فبعض الأشخاص يوقف عرض الصور في متصفحه الخاص فعندها سيظهر هذا النص , كما أنه يرفع من ترتيب موقعك ضمن نتائج البحث في محركات البحث . إذاً عزيزي المبرمج لاتهمل إعطاء قيمة للنص البديل في كل مرة تستخدم فيها الصور .

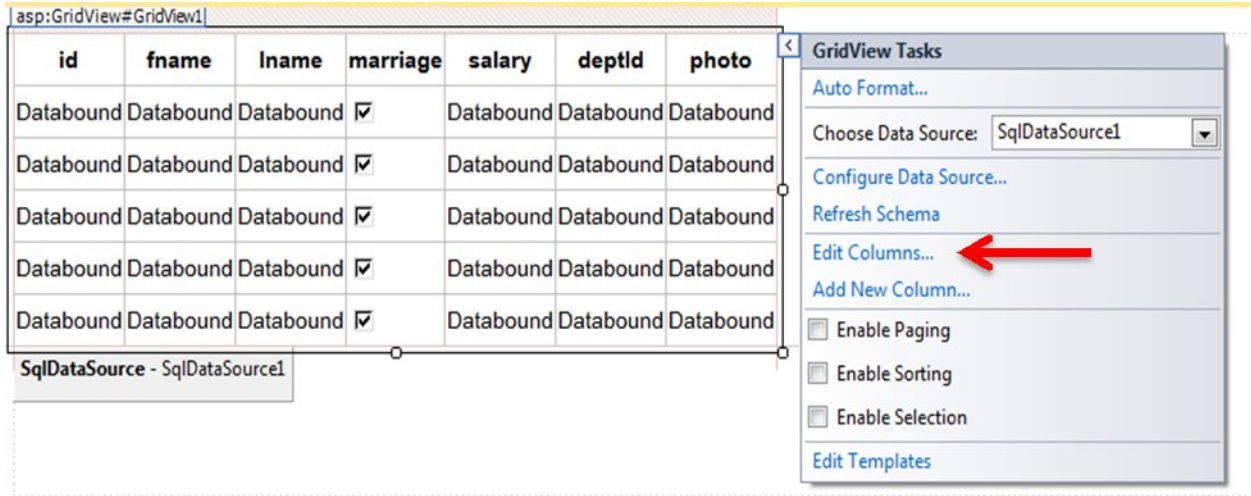
استخدام نمط الحقول ButtonFields

يستخدم النمط ButtonField لعرض أداة Button ضمن الجدول GridView وبالتالي نستطيع برمجة هذا الزر لتنفيذ كود معين عند النقر عليه , مثلا في جدول الموظفين employees سنقوم بعرض بيانات الموظفين و إضافة عمود جديد من النمط ButtonField ضمن الـ GridView وعند النقر عليه يتم تحميل صورة الموظف إلى جهاز المتصفح كما توضح الصورة التالية :

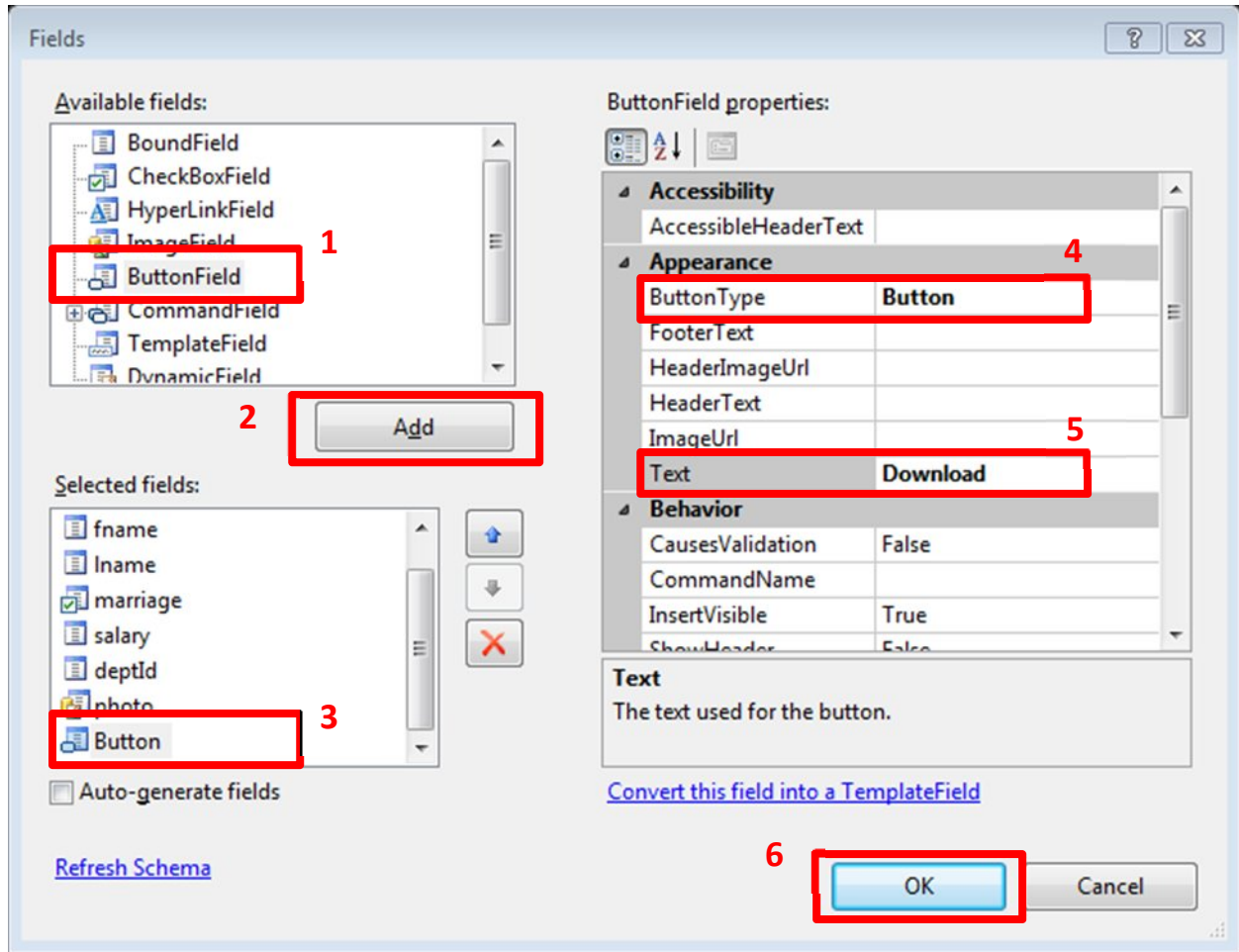


id	fname	lname	marriage	salary	deptId	photo	
1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1		Download
2	إيمان	أحمد	<input type="checkbox"/>	1500	1		Download
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200	2		Download
4	بتول	الخلف	<input type="checkbox"/>	1300	2		Download
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900	3		Download
6	لين	مراد	<input checked="" type="checkbox"/>	800	3		Download

نضيف صفحة جديدة لنطبق المثال عليها , أضف GridView وقم بضبط إعداداته واتصاله بقاعدة البيانات كما تعلمنا في الفقرات السابقة ليصبح ممثالا للشكل الظاهر في الصورة بالأعلى (باستثناء ظهور الأزرار Download) , بعد تجهيز ماسبق نأتي الآن إلى محور حديثنا هنا وهو إضافة حقل من النمط ButtonField , أنقر على smart tag ضمن ال GridView واختر Edit Columns



ستظهر النافذة Fileds , ضمن القسم Available Fields اختر النمط ButtonField واضغط على Add , ثم قم بتحديد الحقل الجديد الذي تم إضافته في القسم Selected Fields , بعد ذلك انتقل إلى القسم ButtonField properties و قم بتعديل قيمة الخاصية ButtonType إلى القيمة Button , تعديل الخاصية Text لإعطائها القيمة Download هذه الخطوات تتوضح من خلال الصورة التالية :



ملاحظة : بإمكاننا إعطاء الخاصية ButtonType القيمة Link أو Image حيث سيتم معاملتهم باعتبارهم أدوات LinkButton و ImageButton وهما مشتقان عموما من الأداة Button وقد تعرفنا عليهم بالتفصيل ضمن فصل سابق .

بعد الضغط على OK ضمن النافذة السابقة سيتم إضافة الحقل الجديد من النمط ButtonField و أصبح بإمكاننا البدء بكتابة الكود ضمن الأزرار التي أضيفت .

يتم كتابة الكود فعليا ضمن الحدث RowCommand التابع للأداة GridView .

تجربة بسيطة :

سنقوم بإضافة أداة Lable إلى الصفحة الحالية التي نعمل عليها ونعطي الخاصية Text القيمة 0 (صفر) وعند النقر على أحد الأزرار التي ضمن ال GridView يزيد هذا العدد بمقدار واحد , قم بتحديد الأداة GridView ثم ضمن الحدث GridView1_RowCommand أكتب الكود التالي :

كود C#

```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    Label11.Text = (int.Parse(Label11.Text) + 1).ToString();
}
```

كود VB

```
Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
GridViewCommandEventArgs)
    Label11.Text = (Integer.Parse(Label11.Text) + 1).ToString()
End Sub
```

أما كود الصفحة فيكون بالشكل :

كود ASP.net

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataKeyNames="id" DataSourceID="SqlDataSource1"
onrowcommand="GridView1_RowCommand" >
  <Columns>
    <asp:BoundField DataField="id" HeaderText="id" ReadOnly="True"
SortExpression="id" />
    <asp:BoundField DataField="fname" HeaderText="fname"
SortExpression="fname" />
    <asp:BoundField DataField="lname" HeaderText="lname"
SortExpression="lname" />
    <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
SortExpression="marriage" />
    <asp:BoundField DataField="salary" HeaderText="salary"
SortExpression="salary" />
    <asp:BoundField DataField="deptId" HeaderText="deptId"
SortExpression="deptId" />
    <asp:ImageField DataImageUrlField="photo" HeaderText="photo">
      <ControlStyle Height="75px" Width="75px" />
    </asp:ImageField>
    <asp:ButtonField ButtonType="Button" Text="Download" />
  </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
SelectCommand="SELECT * FROM [employees]" >
</asp:SqlDataSource>

<br />
<asp:Label ID="Label11" runat="server" Text="0" />
```

قم بتشغيل الصفحة الآن و قم بالنقر على الأزرار ضمن الـ GridView ولاحظ زيادة قيمة الأداة Lable .
 إذا و حتى الآن فإن جميع الأزرار تقوم بتنفيذ نفس الكود تماما (وهو زيادة العداد بقيمة واحد) و لكن ما نريده نحن هو تحميل الصورة المجاورة للزر و بالتالي يجب معرفة مسار الصورة التي تظهر في نفس سطر الزر الذي تم النقر عليه , نعود إلى كود الصفحة نضيف العمود photo إلى DataKeyNames ضمن GridView1 وهذه العملية حتى نتمكن من الوصول إلى قيم هذا العمود , نحدد للخاصية CommandName التابعة للنمط ButtonField القيمة "download" , وبالتالي يصبح كود الصفحة :

كود ASP.net

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
  DataKeyNames="id,photo" DataSourceID="SqlDataSource1"
  onrowcommand="GridView1_RowCommand">
  <Columns>
    <asp:BoundField DataField="id" HeaderText="id" ReadOnly="True"
      SortExpression="id" />
    <asp:BoundField DataField="fname" HeaderText="fname"
      SortExpression="fname" />
    <asp:BoundField DataField="lname" HeaderText="lname"
      SortExpression="lname" />
    <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
      SortExpression="marriage" />
    <asp:BoundField DataField="salary" HeaderText="salary"
      SortExpression="salary" />
    <asp:BoundField DataField="deptId" HeaderText="deptId"
      SortExpression="deptId" />
    <asp:ImageField DataImageUrlField="photo" HeaderText="photo">
      <ControlStyle Height="75px" Width="75px" />
    </asp:ImageField>
    <asp:ButtonField ButtonType="Button" Text="Download"
      CommandName="download" />
  </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
  ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
  SelectCommand="SELECT * FROM [employees]" >
</asp:SqlDataSource>

<br />
<asp:Label ID="Label1" runat="server" Text="0" />
```

أما الكود الذي يقوم بالوصول إلى مسار الصورة وتحميلها على جهاز المتصفح فسيكون ضمن الحدث GridView1_RowCommand بالشكل التالي :

كود C#

```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "download")
    {
        int index = int.Parse((string)e.CommandArgument);
        string filename = GridView1.DataKeys[index].Values["photo"].ToString();

        if (filename != "")
        {
            string path = MapPath(filename);
            byte[] bts = System.IO.File.ReadAllBytes(path);
            Response.Clear();
            Response.ClearHeaders();
            Response.AddHeader("Content-Type", "Application/octet-stream");
            Response.AddHeader("Content-Length", bts.Length.ToString());
            Response.AddHeader("Content-Disposition", "attachment; filename=" +
                filename);
            Response.BinaryWrite(bts);
            Response.Flush();
            Response.End();
        }
    }
}
```

كود VB

```
Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
GridViewCommandEventArgs)
    If e.CommandName = "download" Then
        Dim index As Integer = Integer.Parse(CType(e.CommandArgument, String))
        Dim filename As String = GridView1.DataKeys(index).Values("photo").ToString()

        If filename <> "" Then
            Dim path As String = MapPath(filename)
            Dim bts() As Byte = System.IO.File.ReadAllBytes(path)
            Response.Clear()
            Response.ClearHeaders()
            Response.AddHeader("Content-Type", "Application/octet-stream")
            Response.AddHeader("Content-Length", bts.Length.ToString())
            Response.AddHeader("Content-Disposition", "attachment; filename=" +
                filename)
            Response.BinaryWrite(bts)
            Response.Flush()
            Response.End()
        End If
    End If
End Sub
```

إن المتحول e يمتلك مجموعة من الخصائص منها CommandName و CommandArgument والذين قمنا بتحديد قيمهما ضمن الأداة GridView , بداية اختبرنا قيمة CommandName للتحقق من أن الزر الذي تم النقر عليه هو أحد أزرار الـ download (ضمن هذا المثال بإمكاننا إلغاء هذا الشرط لأنه أساسا لا يوجد سوى عمود واحد من النمط ButtonFields ولكن لو تم وضع أكثر من عمود من هذا النمط عندئذ لابد من فحص قيمة CommandName لمعرفة أي العمودين تم النقر فيه) . قمنا بعد ذلك بالحصول على رقم السطر الذي تم نقر الزر فيه وتخزينه في المتحول index ثم حصلنا على قيمة الخلية التي هي تقاطع العمود photo مع السطر ذو الدليل index , وذلك لمعرفة مسار الصورة على السيرفر وأخيرا قمنا بتنزيلها على جهاز المستخدم .

استخدام نمط الحقول HyperLinkField

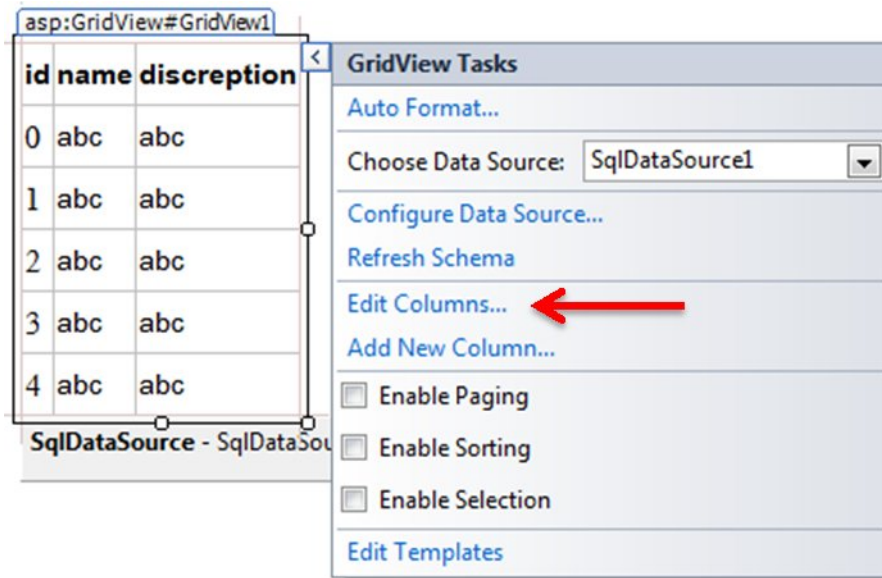
نستخدم نمط الحقول HyperLinkField لبناء روابط تشعبية داخل الأداة GridView , وبإمكاننا جعل هذه الروابط ديناميكية بحيث يقوم كل منها بعرض نص مختلف عن الآخر و ينقلنا إلى صفحة مختلفة عن غيره وبإمكاننا من ناحية أخرى جعل هذه الروابط ثابتة النص وتنقلنا جميعها إلى صفحة ثابتة .

سنقوم بإنشاء صفتين , الصفحة الأولى (Departments.aspx) نعرض فيها أقسام الشركة وعند النقر على اسم قسم ما يتم الانتقال إلى صفحة أخرى (employees.aspx) لعرض الموظفين التابعين لهذا القسم كما توضح الصور التالية :

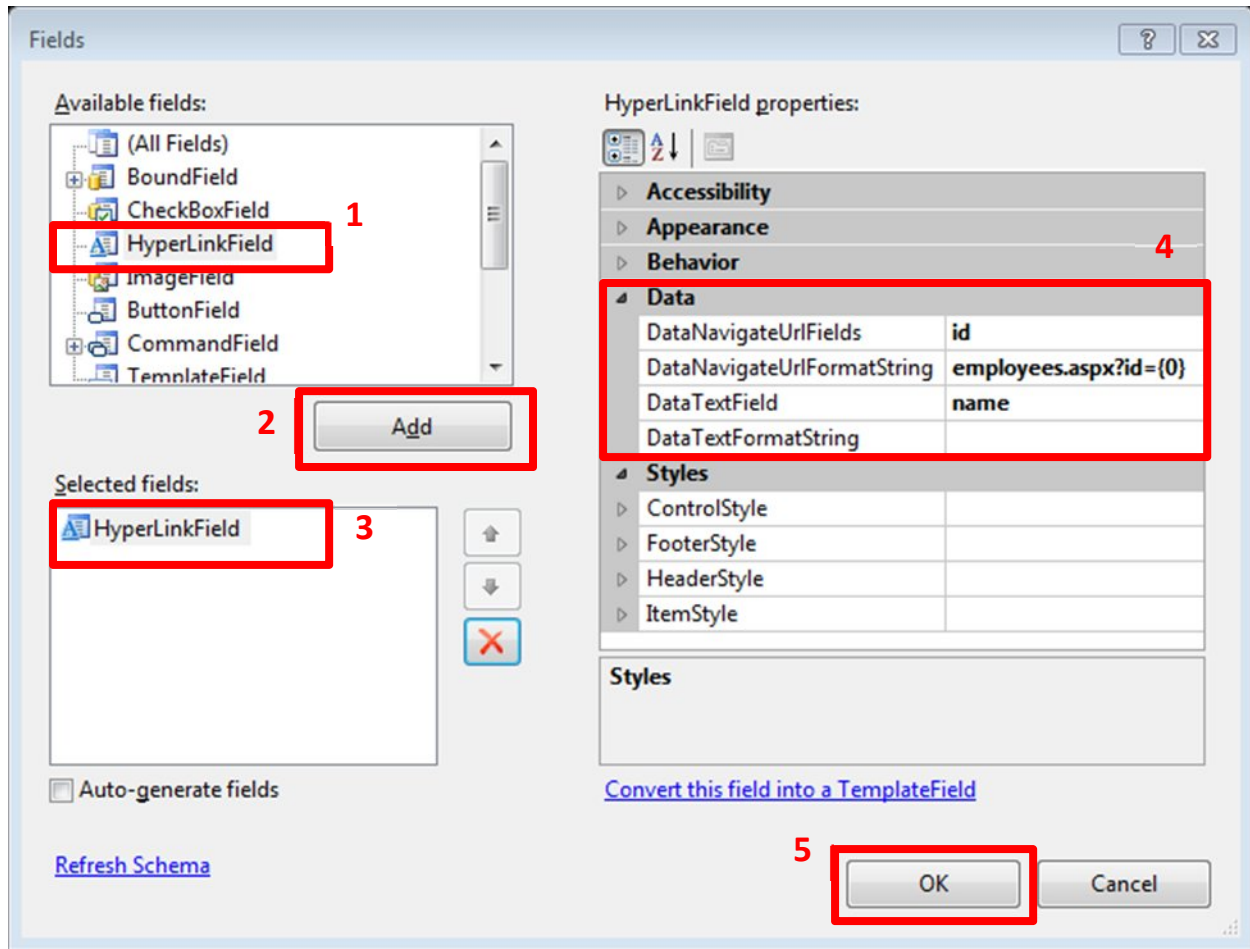
اسم القسم
قسم الإدارة
قسم المحاسبة
قسم العلاقات

id	fname	lname	marriage	salary	deptId	photo
1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
2	إيمان	أحمد	<input type="checkbox"/>	1500	1	

لنقم بداية ببناء الصفحة Departments.aspx , أضف عليها أداة GridView اربطها مع الجدول departments . ثم نضغط على الـ smart tag الخاص بالأداة GridView ونختار Edit Columns



ستظهر النافذة Fields , من القسم selected fields قم بحذف جميع الحقول الموجودة , بعد ذلك اختر HyperLinkField من القسم Available fields واضغط على Add , نقوم الآن بضبط بعض خصائص هذا الحقل الجديد . نعطي الخاصية DataNavigateUrlField القيمة id , كما نعطي الخاصية DataNavigateUrlFormatString القيمة employees.aspx?id={0} هذه العبارة تمثل اسم الصفحة التي سيتم الانتقال إليها عند النقر على أحد الروابط التشعبية التي سيولدها هذا الحقل , نلاحظ أنه يحتاج لقيمة للبارامتر {0} هذه القيمة يتم أخذها من قيمة الخاصية DataNavigateUrlField التي عدلناها قبل قليل (سنتأتي على شرح هذه الخصائص بعد قليل) , أخيرا نعطي الخاصية DataTextField القيمة name وهي تمثل النص الذي سيتم عرضه كرابط تشعبي . الصورة التالية توضح الخطوات التي قمنا بها



إلى هنا نكون قد انتهينا من إعداد هذا الحقل ولكن قبل إغلاق النافذة السابقة بإمكانك تعديل الخاصية HeaderText لإعطاء عنوان لهذا الحقل وليكن "اسم القسم".

كود الصفحة:

```

ASP.net كود
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:HyperLinkField
        DataNavigateUrlFields="id"
        DataNavigateUrlFormatString="employees.aspx?id={0}"
        DataTextField="name"
        HeaderText="القسم اسم" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"

```

```

ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
SelectCommand="SELECT * FROM [departments]"></asp:SqlDataSource>
</div>

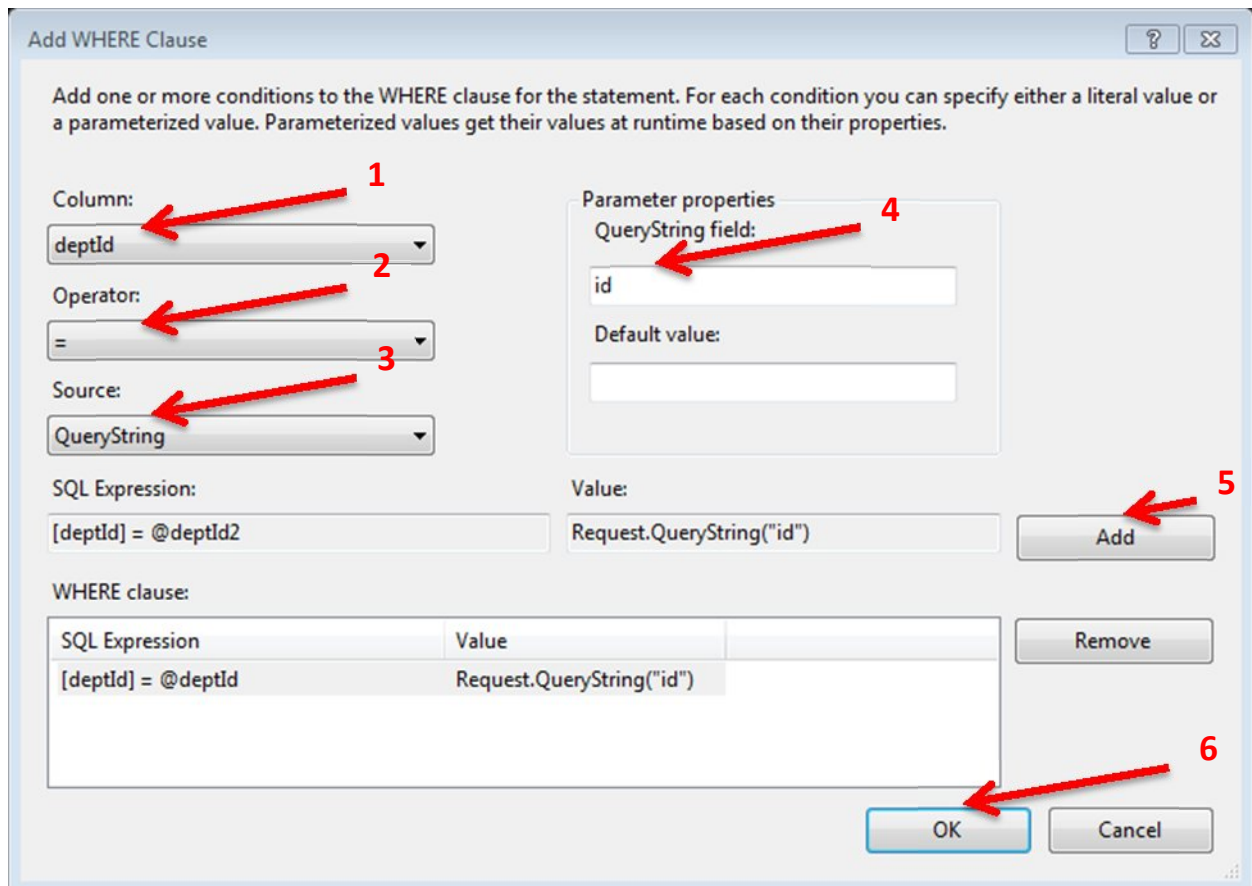
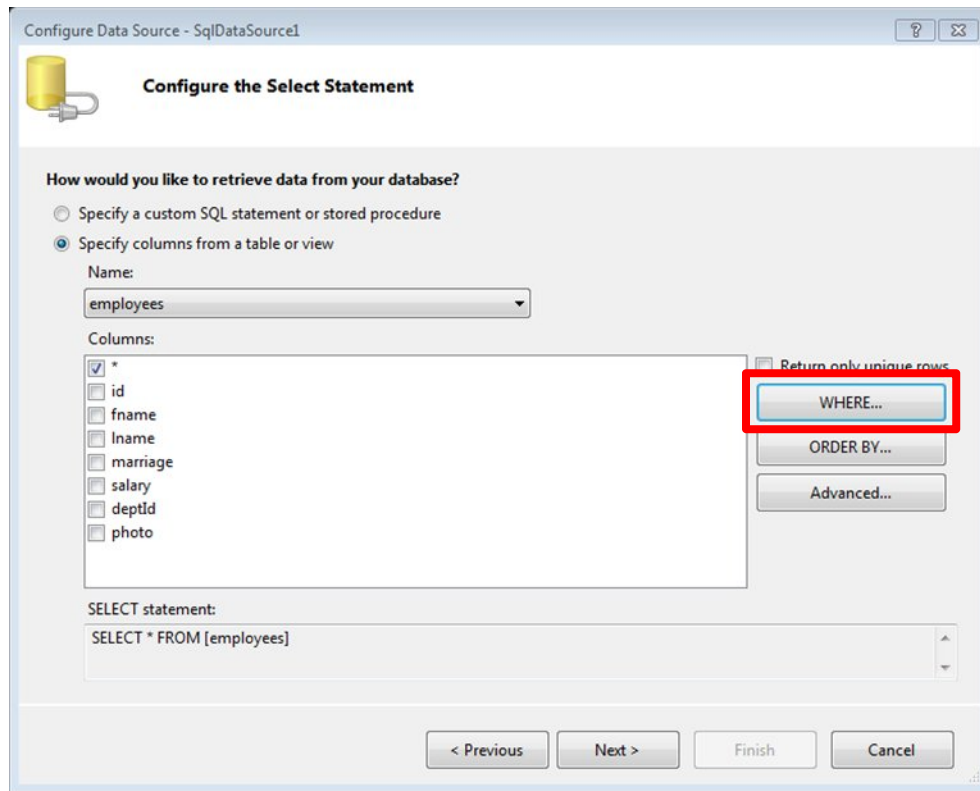
```

نمط الحقول HyperLinkField يدعم الخصائص التالية :

- ✓ DataNavigateUrlFields : لتحديد حقل (أو مجموعة حقول) من مصدر البيانات ليتم استخدامه مع الخاصية DataNavigateUrlFormatString
- ✓ DataNavigateUrlFormatString : لتحديد صيغة النص المراد بناؤه لتمثيل الرابط الذي سيتم الانتقال إليه
- ✓ DataTextField : لتحديد حقل من مصدر البيانات يتم استخدامه كنص ظاهر للرابط
- ✓ DataTextFormatString : لتحديد صيغة النص المطلوب بناؤه ليظهر كنص للرابط
- ✓ NavigateUrl : لتحديد رابط ثابت يتم الانتقال إليه
- ✓ Target : لتحديد أسلوب عرض الصفحة المحددة في الرابط وهي تأخذ إحدى القيم , blank , parent , self , top كما يمكن أن تأخذ اسم لـ frame أو iframe
- ✓ Text : لتحديد نص ثابت يظهر كنص ظاهر للرابط

إلى هنا نكون قد انتهينا من الحديث عن نمط الحقول HyperLinkField بإمكانك عرض الصفحة ضمن المتصفح و مشاهدة النتيجة . بالطبع عند النقر على أحد الروابط فإنه سينقلك إلى الصفحة employees وهي غير موجودة بعد ولكن لاحظ العنوان المطلوب والقيمة التي يأخذها البارامتر id بحسب رقم القسم الذي اخترته .

عموما سنقوم الآن ببناء الصفحة employees.aspx لعرض الموظفين التابعين للقسم الذي تم اختياره من الصفحة departments . أضف أداة GridView واربطها مع الجدول employees من قاعدة البيانات ولكن أضف شرط على عبارة الاستعلام وهو أن يتم عرض الموظفين التابعين للقسم الذي تم تمرير رقمه ضمن عنوان الصفحة (ناقشنا كيفية إضافة هذه الشروط في فصل سابق) الصورة التالية توضح العمل المطلوب



نكمل باقي الخطوات كالمعتاد .

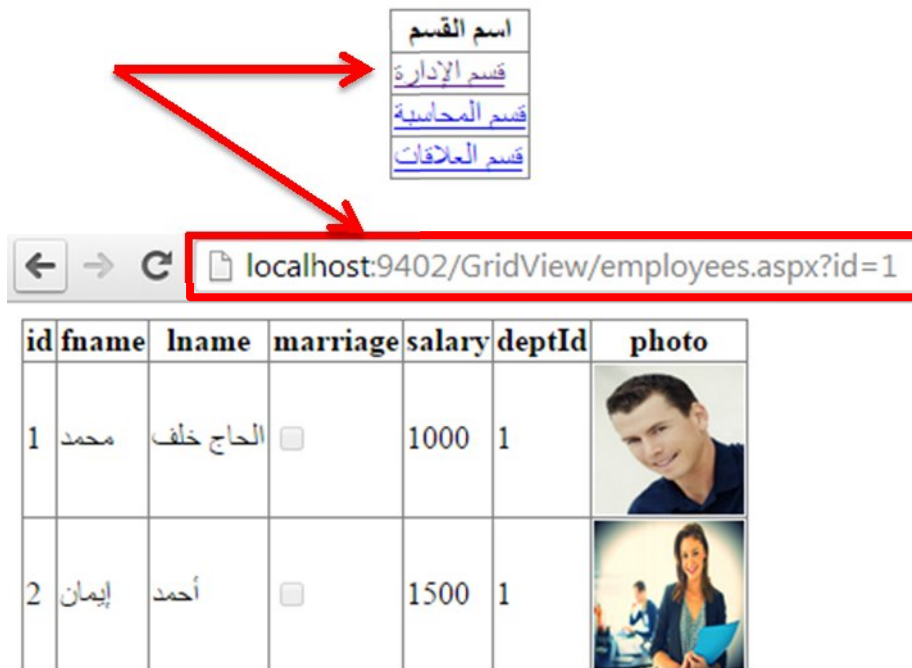
كود الصفحة (سأكتفي بعرض الجزء الخاص بعبارة الاستعلام أما كود الأداة GridView فهو تقليدي ولاجديد عليه في هذه الصفحة)

```

ASP.net كود
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
  ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
  SelectCommand="SELECT * FROM [employees] WHERE ([deptId] = @deptId)">
  <SelectParameters>
    <asp:QueryStringParameter Name="deptId" QueryStringField="id"
      Type="Int32" />
  </SelectParameters>
</asp:SqlDataSource>



```

قم الآن بإعادة اختبار التمرين كاملا , شغل الصفحة departments واختر قسم ما ليتم عرض الصفحة employees مع بيانات الموظفين التابعين لهذا القسم . مرة أخرى لاحظ كيفية تمرير البارامتر id ضمن عنوان الصفحة .



اسم القسم
قسم الإدارة
قسم المحاسبة
قسم العلاقات

localhost:9402/GridView/employees.aspx?id=1

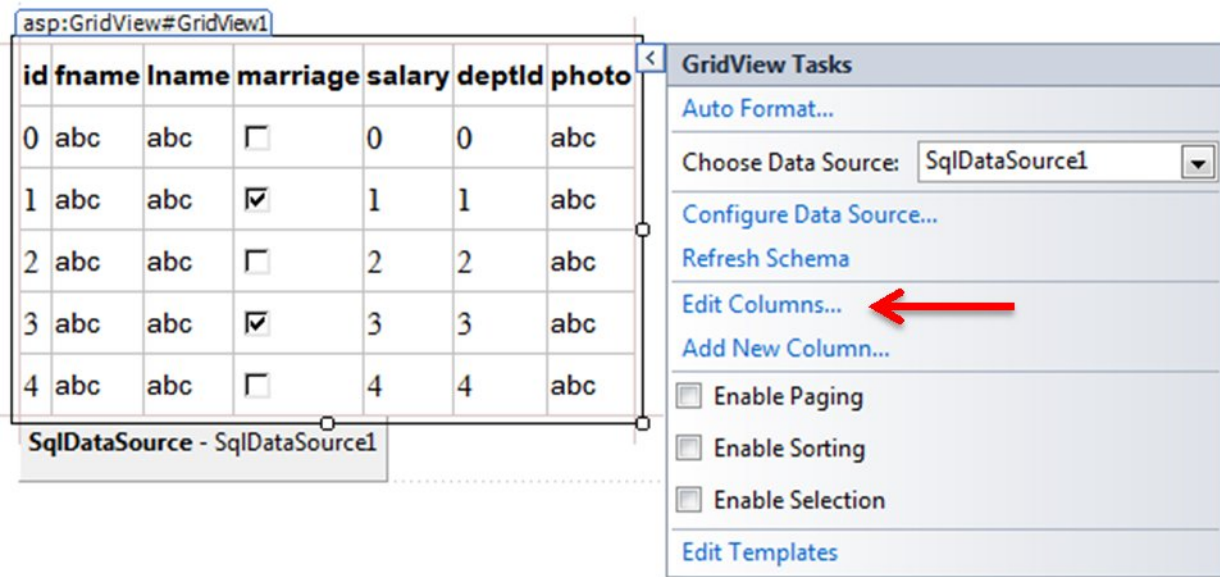
id	fname	lname	marriage	salary	deptId	photo
1	محمد	الحاج خلف	<input type="checkbox"/>	1000	1	
2	إيمان	أحمد	<input type="checkbox"/>	1500	1	

استخدام نمط الحقول TemplateField

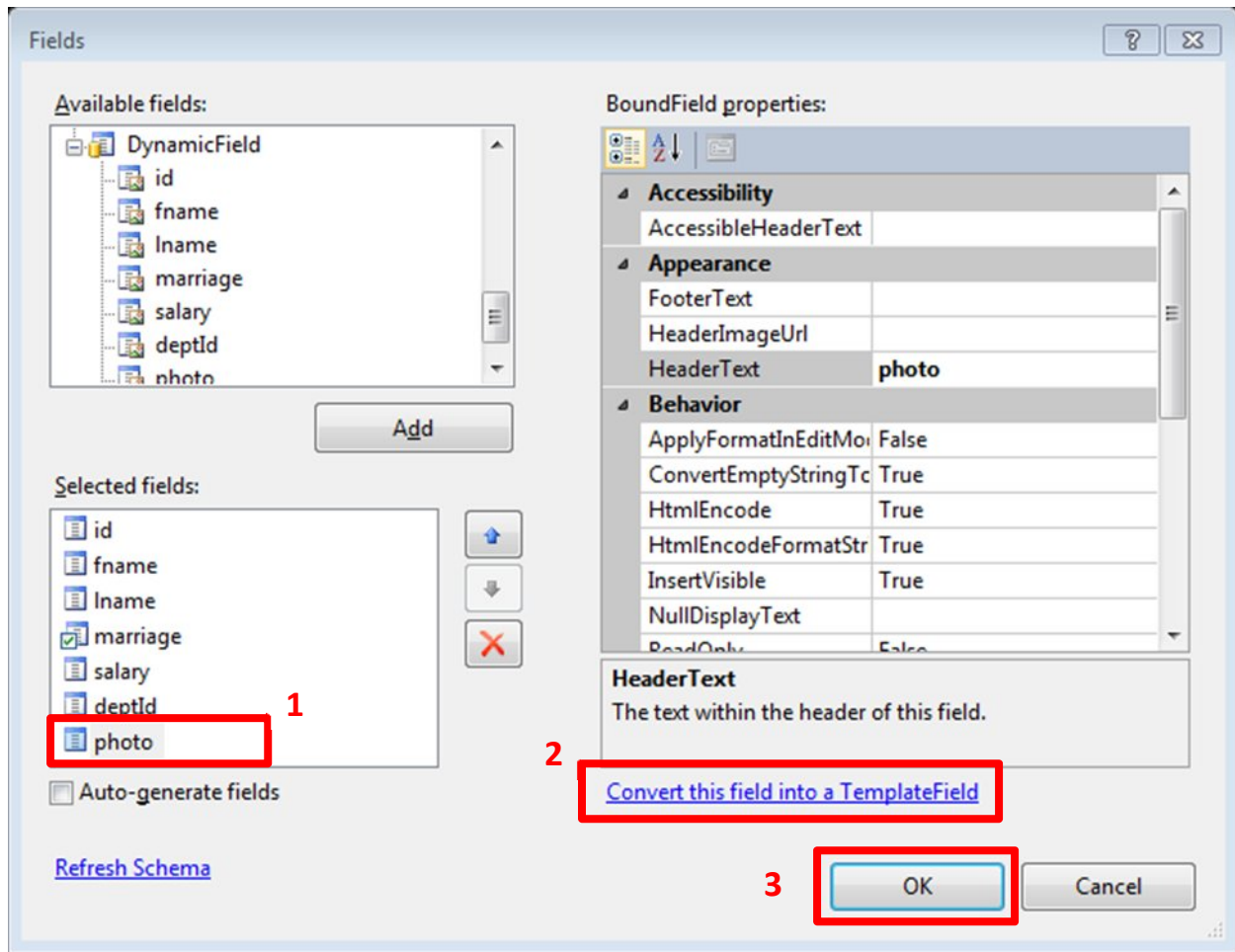
يمكننا هذا النمط من إضافة أي شيء نرغب به داخل عمود الـ GridView , فهو بمثابة قالب فارغ نستطيع من خلاله إنشاء الشكل الذي نريد , حيث بإمكاننا إضافة معظم أدوات ASP.NET أو HTML ليتم عرضها ضمن هذا النمط (Button, Image, HyperLink ...) . وبالتالي نستطيع القول إن تمكنت جيدا من هذا النمط فهو يغنيك عن جميع الأنماط السابقة التي تحدثنا عنها . لتوضيح أسلوب التعامل مع هذا النمط سنقوم بتطبيق ثلاث أمثلة مختلفة من ناحية الهدف و الصعوبة وذلك للإحاطة بأكبر قدر ممكن من مزايا هذا النمط.

المثال الأول :

سنقوم ضمن هذا المثال بتوضيح كيفية عرض صورة ضمن نمط الحقول TemplateField , قم بإضافة صفحة جديد و أضف عليها أداة GridView و اربطها مع الجدول employees ليتم عرض جميع بيانات الموظفين في الجدول , قم بالنقر على SmartTag و اختر EditColumns



ثم قم بتحديد العمود photo , ثم اضغط على Convert this field into a TemplateField , ثم اضغط على OK



وبهذا نكون قد حولنا العمود photo من النمط الافتراضي له (BoundField) إلى النمط TemplateField . ولو قمنا بفحص كود الصفحة سنجد الجزء التالي الخاص بالعمود photo

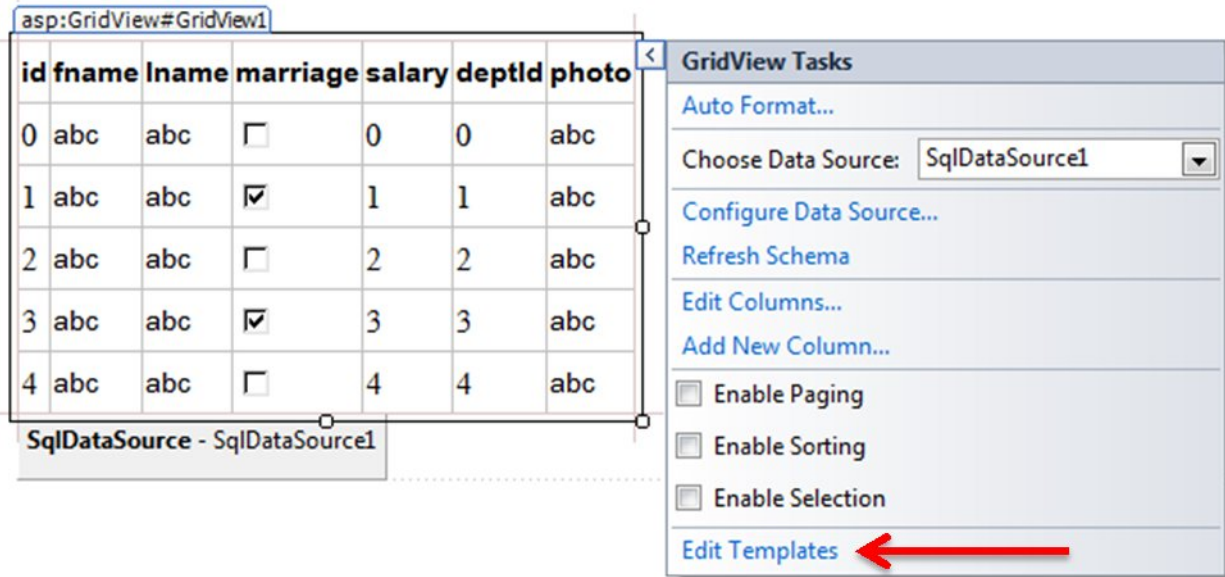
كود ASP.net

```
<asp:TemplateField HeaderText="photo" SortExpression="photo">
  <ItemTemplate>
    <asp:Label ID="Label1" runat="server" Text='<%= Bind("photo") %>' />
  </ItemTemplate>
  <EditItemTemplate>
    <asp:TextBox ID="TextBox1" runat="server" Text='<%= Bind("photo") %>' />
  </EditItemTemplate>
</asp:TemplateField>
```

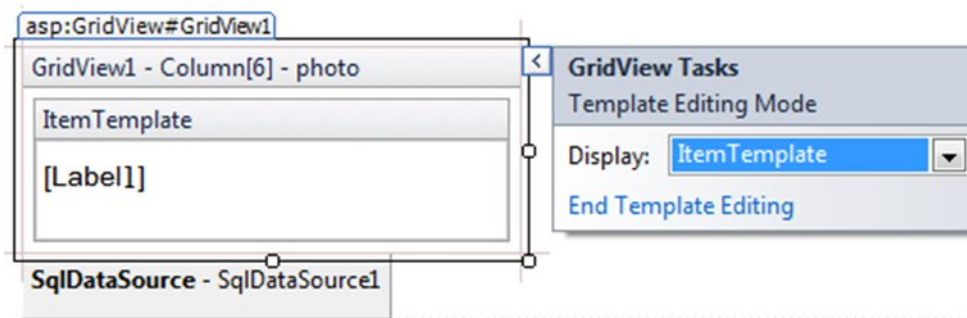
نلاحظ أن الحقل photo أصبح من النمط TemplateField و بداخله أداتين أو قالبين , القالب الأول ItemTemplate و بداخله يتم وضع المحتويات التي نرغب بعرضها عند تنفيذ الصفحة و تكون الأداة

GridView بوضعية عرض البيانات (وهي الحالة الافتراضية) , القالب الثاني EditItemTemplate وبداخله يتم وضع المحتويات التي سيتم عرضها عندما تكون الأداة GridView بوضعية تعديل البيانات .

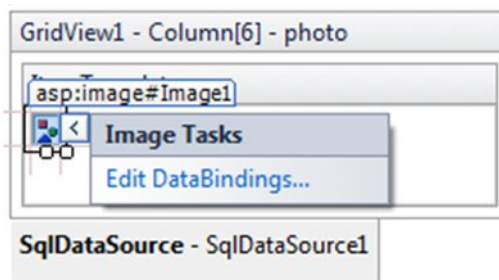
نعود الآن إلى الصفحة بوضعية التصميم و نقر على samrtTag ونختار Edit Templates



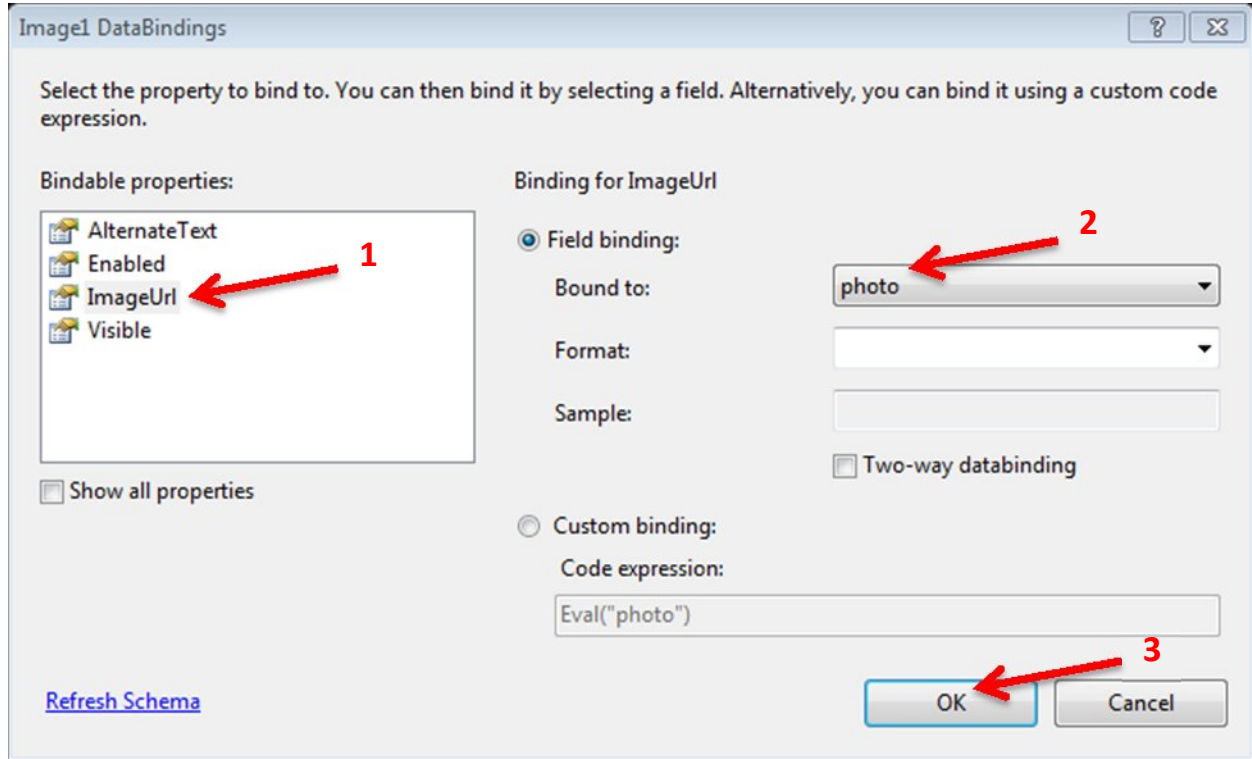
هذا الخيار يمكّننا من تعديل جميع الحقول التي من النمط TemplateField



نحذف الأداة Lable1 و نضع عوضاً عنها أداة Image , لنجعلها تقوم بعرض صورة الموظف

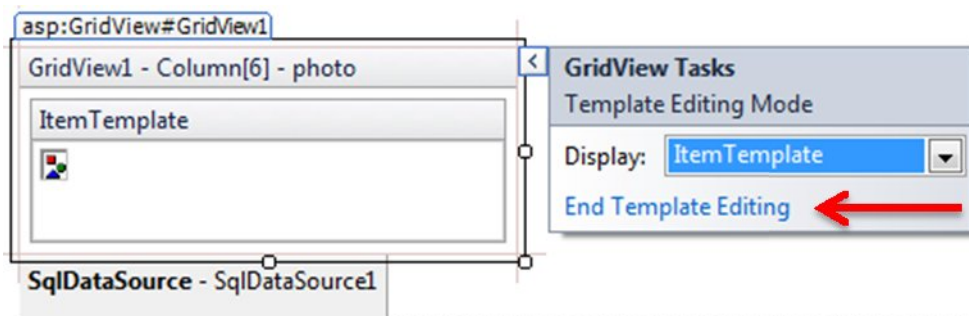


نضغط على SmartTag الخاص بالأداة Image1 التي ضفناها و نختار Edit DataBindings , هذه الخطوة لكي نحدد اسم العمود الذي يحتوي على مسارات الصور ضمن الجدول employees كما توضح الصورة التالية :



نلاحظ أنه بالإمكان تحديد عمود آخر يكون بمثابة مصدر بيانات للخاصية AlternateText والتي سيتم عرض قيمتها في حال عدم ظهور الصورة .

بعد الانتهاء من تحديد مصدر بيانات الصورة نضغط على SmartTag الخاص بالأداة GridView ونختار . End Template Editing



نعود إلى كود الصفحة و بالتحديد إلى كود الحقل photo لنجده أصبح كالتالي :

ASP.net كود

```
<asp:TemplateField HeaderText="photo" SortExpression="photo">
  <ItemTemplate>
    <asp:Image ID="Image1" runat="server" ImageUrl='<## Eval("photo") %>' />
  </ItemTemplate>
  <EditItemTemplate>
    <asp:TextBox ID="TextBox1" runat="server" Text='<## Bind("photo") %>' />
  </asp:TextBox>
  </EditItemTemplate>
</asp:TemplateField>
```

نلاحظ من خلال الكود السابق تغيير قيمة محتويات القالب ItemTemplate , حيث كان Label و قد أصبح Image , كان بإمكاننا بسهولة إجراء هذا التعديل على الكود مباشرة دون الحاجة للإستعانة بالنوافذ السابقة , إنما أردت توضيح كلا الأسلوبين ولك الحرية فيما تختار .

قم بتشغيل الصفحة السابقة و ستلاحظ عرض صور الموظفين دون الحاجة لاستخدام نمط الحقول ImageField و بأسلوب مماثل بإمكانك وضع أوات أخرى والاستغناء عن أنماط الحقول السابقة . قم بحذف القالب EditItemTemplate فلا داعي له في مثالنا هذا , كود الصفحة لهذا المثال كاملا :

ASP.net كود

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="fname"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="lname"
        SortExpression="lname" />
      <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
        SortExpression="marriage" />
      <asp:BoundField DataField="salary" HeaderText="salary"
        SortExpression="salary" />
      <asp:BoundField DataField="deptId" HeaderText="deptId"
        SortExpression="deptId" />
      <asp:TemplateField HeaderText="photo" SortExpression="photo">
        <ItemTemplate>
          <asp:Image ID="Image1" runat="server"
            ImageUrl='<## Eval("photo") %>' />
        </ItemTemplate>
      </asp:TemplateField>
    </Columns>
  </asp:GridView>
</div>
```

```

        </asp:TemplateField>
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%"$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]"></asp:SqlDataSource>
</div>

```

لاحظنا من خلال المثال السابق أن نمط الحقول TemplateField يحتوي على نمطين من القوالب أحدهما لوضعية عرض البيانات و الآخر يظهر عند وضعية التعديل , في الحقيقة يوجد أربعة أنماط أخرى حيث إن نمط الحقول TemplateField يدعم أنماط القوالب التالية :

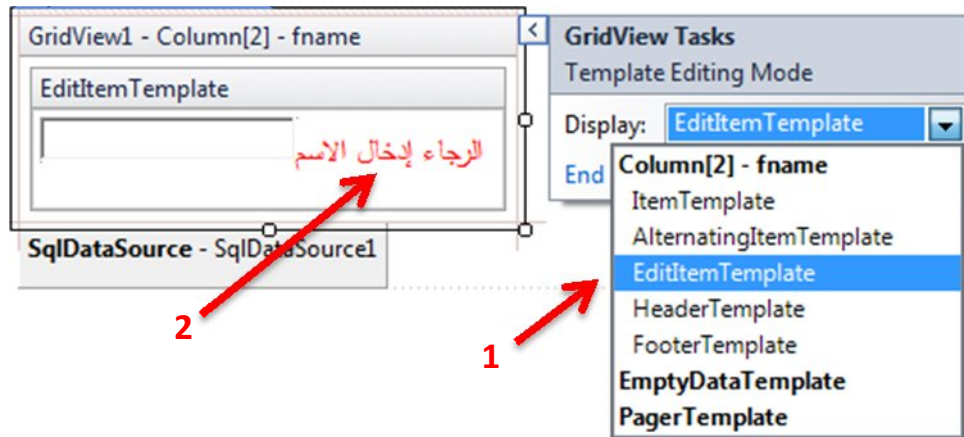
- ✓ ItemTemplate : تظهر محتويات هذا القالب في كل سطر بيانات يتم عرضه ضمن الأداة GridView
- ✓ EditItemTemplate : تظهر محتويات هذا القالب فقط عندما يكون السطر في حالة تعديل البيانات
- ✓ InsertItemTemplate : تظهر محتويات هذا القالب فقط عندما نرغب بإضافة سطر جديد و هذه العملية غير مدعومة ضمن الأداة GridView وبالتالي لا يوجد استخدام فعلي لهذا القالب ضمن الـ GridView ولكننا سنستخدمه مع أدوات أخرى ضمن فصول قادمة
- ✓ AlternatingItemTemplate : تظهر محتويات هذا القالب في الأسطر ذات الدليل الزوجي ضمن الـ GridView (السطر الثاني و الرابع والسادس ... وهكذا) , في حين أن الأسطر الفردية تبقى لتعرض محتويات القالب ItemTemplate , وإذا لم يتم ذكر هذا القالب فإن جميع الأسطر ستعرض محتويات القالب ItemTemplate (كما فعلنا ضمن المثال السابق)
- ✓ HeaderTemplate : محتويات هذا القالب تظهر كعنوان يظهر أعلى العمود
- ✓ FooterTemplate : محتويات هذا القالب تظهر كعنوان يظهر أسفل العمود

المثال الثاني

سنقوم ضمن هذا المثال بالتعامل مع نمط القوالب EditItemTemplate وذلك لإضافة أداة التحقق RequiredFieldValidator لضمان أن المستخدم لم يترك حقل البيانات فارغ عند إجراء عملية التعديل

	id	fname	lname
Update Cancel	1	<input type="text" value="الرجاء إدخال الاسم"/>	الحاج خلف
Edit	2	إيمان	أحمد
Edit	3	أحمد	المصطفى
Edit	4	بتول	الخلف
Edit	5	قصي	ابراهيم
Edit	6	لين	مراد

قم بإنشاء صفحة جديدة و أضف عليها أداة GridView مع تفعيل إمكانية تعديل وحذف البيانات , قم بعد ذلك بتحويل الحقل fname إلى النمط TemplateField , اضغط على SmartTag الأداة GridView واختر Edit Templates



قم بتحديد القالب EditItemTemplate , ثم أضف أداة التحقق RequiredFieldValidator بجانب الأداة TextBox الموجودة مسبقا , قم بظبط قيمة الخاصية ControlToValidate التابعة لأداة التحقق لربطها مع الـ TextBox (لمزيد من المعلومات عن أدوات التحقق راجع فصل "استخدام أدوات التحقق") . أخيرا اضغط على End Template Editing وقم بتشغيل الصفحة , قم بالضغط على Edit وحاول تعديل البيانات بعد أن تقوم بمسح قيمة الاسم الأول ولاحظ ظهور رسالة الاعتراض و عدم تنفيذ عملية التعديل .

كود الصفحة :

```

ASP.net كود
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1" >
    <Columns>
      <asp:CommandField ShowEditButton="True" />
      <asp:BoundField DataField="id" HeaderText="id" ReadOnly="True"
        SortExpression="id" />
    </Columns>
  </asp:GridView>
</div>

```



```

<asp:TemplateField HeaderText="fname" SortExpression="fname">
  <ItemTemplate>
    <asp:Label ID="Label1" runat="server"
      Text='<%= Bind("fname") %>'>
    </asp:Label>
  </ItemTemplate>
  <EditItemTemplate>
    <asp:TextBox ID="TextBox1" runat="server"
      Text='<%= Bind("fname") %>'></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
      runat="server" ControlToValidate="TextBox1"
      ErrorMessage="الاسم إدخال الرجاء" ForeColor="Red" >
    </asp:RequiredFieldValidator>
  </EditItemTemplate>
</asp:TemplateField>
<asp:BoundField DataField="lname" HeaderText="lname"
  SortExpression="lname" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
  ConnectionString="<%= $ ConnectionStrings:ConnectionString %>"
  SelectCommand="SELECT * FROM [employees]"
  UpdateCommand="UPDATE [employees] SET [fname] = @fname, [lname] = @lname
  WHERE [id] = @id">
  <UpdateParameters>
    <asp:Parameter Name="fname" Type="String" />
    <asp:Parameter Name="lname" Type="String" />
    <asp:Parameter Name="id" Type="Int32" />
  </UpdateParameters>
</asp:SqlDataSource>
</div>

```

المثال الثالث :

استخدام الأداة DropDownList داخل الأداة GridView

ضمن هذا المثال سنقوم باستخدام نمط الحول TemplateField لوضع أداة DropDownList داخل الأداة GridView حيث تظهر في حالة تحرير سطر البيانات لأجراء تعديل ما (راجع المثال السابق), كما توضح الصورة التالية :

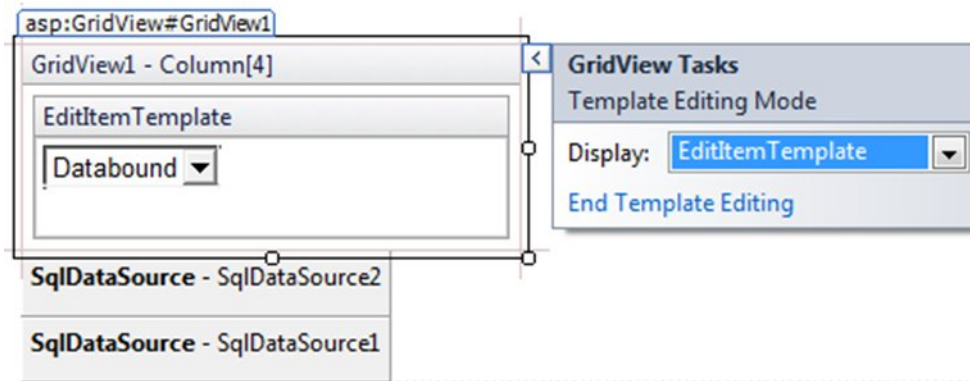
	id	fname	lname	
Edit	3	محمد	الحاج خلف	قسم العلاقات
Edit	1	إيمان	أحمد	قسم الإدارة
Edit	2	أحمد	المصطفى	قسم المحاسبة
Edit	2	يتول	الخلف	قسم المحاسبة
Edit	3	قصي	ابراهيم	قسم العلاقات
Update Cancel	3	لين	مراد	<div style="border: 1px solid black; padding: 2px;"> قسم العلاقات ▼ <ul style="list-style-type: none"> قسم الإدارة <li style="background-color: #e0e0e0;">قسم المحاسبة قسم العلاقات </div>

وبالتالي فإننا سنتعامل مع القالب EditItemTemplate . بداية قم بإضافة أداة GridView واربطها مع مصدر بيانات SqlDataReader1 بحيث ضمن عبارة الـ Select نقوم بربط الجدولين Departments و Employees مع بعضهما (عبر الحقل رقم القسم) وذلك لكي نتمكن من عرض اسم القسم ضمن جدول الـ GridView و قم طبعاً بتفعيل خيارات التعديل والحذف ضمن مصدر البيانات .

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1">
    <Columns>
      <asp:CommandField ShowEditButton="True" />
      <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
        ReadOnly="True" SortExpression="id" />
      <asp:BoundField DataField="fname" HeaderText="fname"
        SortExpression="fname" />
      <asp:BoundField DataField="lname" HeaderText="lname"
        SortExpression="lname" />
    </Columns>
  </asp:GridView>
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [employees].[id], [fname], [lname], [deptId], [name]
FROM [employees] inner join [departments] ON [employees].[deptId] = [departments].[id] "
    UpdateCommand="UPDATE [employees] SET [fname] = @fname, [lname] = @lname,
[deptId] = @deptId WHERE [id] = @id">
    <UpdateParameters>
      <asp:Parameter Name="fname" Type="String" />
      <asp:Parameter Name="lname" Type="String" />
      <asp:Parameter Name="deptId" Type="Int32" />
      <asp:Parameter Name="id" Type="Int32" />
    </UpdateParameters>
  </asp:SqlDataSource>
</div>
```

قم بعد ذلك بإضافة حقل جديد من النمط TemplateField وذلك من النافذة Fields والتي تظهر بعد النقر على smart tag الأداة GridView واختيار Edit Columns . بعد ذلك نختار Edit Template و نحدد القالب EditItemTemplate ثم نضع بداخله أداة DropDownList و نربطها مع جدول الأقسام Departments و لكن في عبارة الـ select قم بإعطاء الحقل Id اسم مستعار وليكن deptId وذلك حتى يتطابق مع اسم العمود المقابل له ضمن الجدول Employees لكي نتمكن من استخدام قيمته عند تعديل رقم القسم الخاص بموظف ما . أما ضمن القالب ItemTemplate فإننا سنضيف أداة Label نجعلها تعرض اسم القسم كما يوضح كود الصفحة :



ASP.net كود

```
<div>
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
  DataKeyNames="id" DataSourceID="SqlDataSource1">
  <Columns>
    <asp:CommandField ShowEditButton="True" />
    <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
      ReadOnly="True" SortExpression="id" />
    <asp:BoundField DataField="fname" HeaderText="fname"
      SortExpression="fname" />
    <asp:BoundField DataField="lname" HeaderText="lname"
      SortExpression="lname" />
    <asp:TemplateField>
      <ItemTemplate>
        <asp:Label ID="Label1" runat="server"
          Text='<%# Eval("name") %>' /></asp:Label>
      </ItemTemplate>
      <EditItemTemplate>
        <asp:DropDownList ID="DropDownList1" runat="server"
          DataSourceID="SqlDataSource2" DataTextField="name"
          DataValueField="deptId"
          SelectedValue='<%# Bind("deptId") %>' />
      </asp:DropDownList>
    </EditItemTemplate>
  </Columns>
</asp:GridView>
```

```

        </asp:TemplateField>
    </Columns>
</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource2" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [id] as deptId, [name] FROM [departments]">
</asp:SqlDataSource>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT [employees].[id], [fname], [lname], [deptId], [name]
        FROM [employees] inner join [departments]
        ON [employees].[deptId] = [departments].[id] "
    UpdateCommand="UPDATE [employees] SET [fname] = @fname, [lname] = @lname,
        [deptId] = @deptId WHERE [id] = @id">
    <UpdateParameters>
        <asp:Parameter Name="fname" Type="String" />
        <asp:Parameter Name="lname" Type="String" />
        <asp:Parameter Name="deptId" Type="Int32" />
        <asp:Parameter Name="id" Type="Int32" />
    </UpdateParameters>
</asp:SqlDataSource>
</div>

```

الأحداث التي تطلقها الأداة GridView

تطلق الأداة GridView مجموعة كبيرة من الأحداث Events , والتي يمكن تقسيمها إلى ثلاث مجموعات :
 المجموعة الأولى : الأحداث التي يتم إطلاقها بدءاً من ربط الأداة GridView مع مصدر البيانات وانتهاءً
 بعرض البيانات , وهذه الأحداث هي :

- ✓ DataBinding : يتم إطلاقه مباشرة قبل ربط الأداة GridView مع مصدر البيانات
- ✓ DataBound : يتم إطلاقه مباشرة بعد ربط الأداة GridView مع مصدر البيانات
- ✓ RowCreated : يتم إطلاقه عند كل عملية إنشاء لسطر جديد . لنفرض أن مصدر البيانات يحتوي على خمسين سجل , عندئذ يتم إطلاق هذا الحدث ضمن الـ GridView خمسين مرة (مرة واحدة لكل سطر يتم إنشاؤه) .
- ✓ RowDataBound : يتم إطلاقه عند ربط سطر من الـ GridView مع سجل بيانات من مصدر البيانات (مشابه للحدث RowCreated ولكن عند ربط السطر و ليس عند إنشائه) .

المجموعة الثانية : مجموعة الأحداث التي تتعلق باختيار أو تعديل سطر البيانات ضمن الأداة GridView وهي استجابة لفعل قام به المستخدم , وهذه الأحداث هي :

- ✓ RowUpdating : يتم إطلاقه قبل تعديل سطر البيانات مباشرة
 - ✓ RowUpdated : يتم إطلاقه بعد تعديل سطر البيانات مباشرة
 - ✓ RowCancelingEdit : يتم إطلاقه عندما يتم إلغاء عملية تعديل سطر البيانات
 - ✓ RowDeleting : يتم إطلاقه قبل حذف سطر البيانات مباشرة
 - ✓ RowDeleted : يتم إطلاقه بعد حذف سطر البيانات مباشرة
 - ✓ RowCommand : يتم إطلاقه كاستجابة لأي حدث انطلق من قبل أداة موجودة ضمن الأداة GridView , مثلا لو قمت بالضغط على زر موجود ضمن الأداة GridView عندئذ يتم إطلاق هذا الحدث وقد رأينا هذا الأمر ضمن مثال سابق في هذا الفصل .
 - ✓ SelectedIndexChanged : يتم إطلاق هذا الحدث عندما يقوم المستخدم باختيار سطر بيانات مختلف عن السطر الحالي , وهذا الحدث يتم إطلاقه قبل تغيير السطر المختار مباشرة
 - ✓ SelectedIndexChanged : يتم إطلاقه بعد القيام باختيار سطر ما مباشرة
- المجموعة الثالثة : الأحداث التي تتعلق بترتيب البيانات أو تقسيمها إلى عدة صفحات وهي :

- ✓ Sorting : يتم إطلاقه قبل ترتيب البيانات مباشرة
- ✓ Sorted : يتم إطلاقه بعد ترتيب البيانات مباشرة
- ✓ PageIndexChanging : يتم إطلاقه قبل تغيير الصفحة الحالية مباشرة (صفحة GridView)
- وهذا الحدث يعمل فقط في حالة تقسيم البيانات ضمن الأداة GridView إلى عدة صفحات .
- ✓ PageIndexChanged : مشابه للحدث السابق ولكن يتم إطلاقه بعد تغيير الصفحة الحالية مباشرة

تمييز الأسطر وفق شطر معين

سنقوم ضمن هذه الفقرة بتطبيق مثال لتوضيح عمل الحدث RowDataBound , حيث سنقوم بجعل أسطر الموظفين المتزوجين تظهر باللون الأصفر , كما توضح الصورة التالية :

id	fname	lname	marriage	salary
1	محمد	الحاج خلف	<input type="checkbox"/>	1000
2	إيمان	أحمد	<input type="checkbox"/>	1500
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200
4	بتول	الخلف	<input type="checkbox"/>	1300
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900
6	لين	مراد	<input checked="" type="checkbox"/>	

قم بإنشاء صفحة جديدة و أضف إليها أداة GridView و اربطها مع جدول الموظفين , بعد الانتهاء من ذلك قم بكتابة الكود التالي ضمن الحدث RowDataBound الخاص بالأداة GridView

كود C#

```
protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        bool marriage = (bool)DataBinder.Eval(e.Row.DataItem, "marriage");
        if (marriage)
            e.Row.BackColor = System.Drawing.Color.Yellow;
    }
}
```

كود VB

```
Protected Sub GridView1_RowDataBound(ByVal sender As Object, ByVal e As
GridViewRowEventArgs)
    If e.Row.RowType = DataControlRowType.DataRow Then
        Dim marriage As Boolean = CType(DataBinder.Eval(e.Row.DataItem, "marriage"),
Boolean)

        If marriage Then
            e.Row.BackColor = System.Drawing.Color.Yellow
        End If
    End If
End Sub
```

لاحظ بأن الكود السابق يعمل بعد ربط كل سطر ضمن الـ GridView مع سجل البيانات , وبالتالي نفحص قيمة الحقل "marriage" فإن كانت قيمته "True" يتم تلوين السطر الحالي بخلفية صفراء . أما السطر الذي وضعناه في أول الكود فهو يفحص أن السطر الحالي هو سطر بيانات و ذلك حتى يتم استثناء سطر العناوين من الكود السابق . أما الصف DataBinder فهو الذي يقوم بربط سجل البيانات من مصدر البيانات مع سطر ضمن الأداة GridView وذلك عبر الدالة Eval() , لذلك قمنا باستخدامها للوصول إلى قيمة الحقل "marriage" .

كود الصفحة :

كود ASP.net

```
<div>
    <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
        DataKeyNames="id" DataSourceID="SqlDataSource1"
        onrowdatabound="GridView1_RowDataBound" >
```

```

<Columns>
  <asp:BoundField DataField="id" HeaderText="id" ReadOnly="True"
    SortExpression="id" />
  <asp:BoundField DataField="fname" HeaderText="fname"
    SortExpression="fname" />
  <asp:BoundField DataField="lname" HeaderText="lname"
    SortExpression="lname" />
  <asp:CheckBoxField DataField="marriage" HeaderText="marriage"
    SortExpression="marriage" />
  <asp:BoundField DataField="salary" HeaderText="salary"
    SortExpression="salary" />
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
  ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
  SelectCommand="SELECT * FROM [employees]" />
</div>

```

ضمن المثال السابق استخدمنا DataItem , RowType , وهي خصائص تتبع للصف GridViewRow حيث إنه يحتوي على الخصائص التالية :

- ✓ Cells : تمثل مجموعة خلايا السطر الحالي حيث يمكن الوصول لقيمة كل خلية منها .
- ✓ DataItem : يمثل سجل البيانات الذي يتم ربطه مع سطر البيانات الحالي .
- ✓ RowIndex : يمثل دليل السطر الحالي .
- ✓ RowType : يمثل نمط السطر الحالي حيث يمكن أن يأخذ إحدى القيم التالية DataRow, Header, Footer, NullRow, Pager, Separator .

استخدام GridView داخل GridView آخر

هناك العديد من الحالات التي نحتاج فيها لوضع GridView داخل GridView آخر , مثلا لنفترض أننا نريد عرض الأقسام التي لدينا و تحت كل قسم يتم عرض الموظفين العاملين فيه , كما توضح الصورة :

قسم الإدارة				
id	fname	lname	marriage	salary
2	إيمان	أحمد	<input type="checkbox"/>	1500

قسم العلاقات				
id	fname	lname	marriage	salary
1	محمد	الحاج خلف	<input type="checkbox"/>	1000
5	قصي	ابراهيم	<input checked="" type="checkbox"/>	900
6	لين	مراد	<input checked="" type="checkbox"/>	

قسم المحاسبة				
id	fname	lname	marriage	salary
3	أحمد	المصطفى	<input checked="" type="checkbox"/>	1200
4	بتول	الخلف	<input type="checkbox"/>	1300

تكمّن الفكرة ببساطة بوضع الـ GridView الخارجي وربطه مع جدول الأقسام , وإضافة حقل من النمط TemplateField إليه ونضيف ضمنه GridView آخر و نربطه مع جدول الموظفين و نضيف عليه شرط أن يكون رقم القسم مساوي لقيمة متحول ما , هذا المتحول سنقوم بتغيير قيمته عند كل إطلاق حدث RowDataBound ضمن الـ GridView الخارجي بحيث نجعله يساوي قيمة رقم القسم الحالي .

كود الصفحة :

كود ASP.net

```
<div>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="SqlDataSource1" ShowHeader="False"
    onrowdatabound="GridView1_RowDataBound" >
    <Columns>
      <asp:TemplateField>
        <ItemTemplate>
          <center>
            <h2><%#Eval("name") %></h2>
            <asp:GridView ID="GridView2" runat="server"
              AutoGenerateColumns="False"
              DataKeyNames="id" DataSourceID="empDataSource">
              <Columns>
```



```

        <asp:BoundField DataField="id" HeaderText="id"
            InsertVisible="False"
            ReadOnly="True" SortExpression="id" />
        <asp:BoundField DataField="fname" HeaderText="fname"
            SortExpression="fname" />
        <asp:BoundField DataField="lname" HeaderText="lname"
            SortExpression="lname" />
        <asp:CheckBoxField DataField="marriage"
            HeaderText="marriage"
            SortExpression="marriage" />
        <asp:BoundField DataField="salary" HeaderText="salary"
            SortExpression="salary" />
    </Columns>
</asp:GridView>
</center>
<asp:SqlDataSource ID="empDataSource" runat="server"
    ConnectionString="<%= $ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [employees]
        WHERE deptId = @deptId">
    <SelectParameters>
        <asp:Parameter Name="deptId" />
    </SelectParameters>
</asp:SqlDataSource>
</ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= $ConnectionStrings:ConnectionString %>"
    SelectCommand="SELECT * FROM [departments] ORDER BY [name]">
</asp:SqlDataSource>
</div>

```

أما ضمن الحدث RowDataBound الخاص بالأداة GridView1 فنكتب الكود التالي :

كود C#

```

protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        int departmentId = (int)DataBinder.Eval(e.Row.DataItem, "id");
        SqlDataSource empDataSource =
            (SqlDataSource)e.Row.FindControl("empDataSource");
        empDataSource.SelectParameters["deptId"].DefaultValue =
            departmentId.ToString();
    }
}

```

كود VB

```

Protected Sub GridView1_RowDataBound(ByVal sender As Object, ByVal e As
GridViewRowEventArgs)
    If e.Row.RowType = DataControlRowType.DataRow Then
        Dim departmentId As Integer = CType(DataBinder.Eval(e.Row.DataItem, "id"),
Integer)

        Dim empDataSource As SqlDataSource =
            CType(e.Row.FindControl("empDataSource"), SqlDataSource)
        empDataSource.SelectParameters("deptId").DefaultValue =
            departmentId.ToString()

    End If
End Sub

```

حيث يتم تغيير قيمة البارامتر deptId عند كل عملية ربط سجل جديد من الجدول Departments مع سطر من الأداة GridView .

الخاتمة

إلى هنا نأتي إلى نهاية هذا الفصل الطويل والمتعدد الأفكار للتعامل مع الأداة GridView , إن الكثير من المواضيع التي ناقشناها يمكن أن يتم تطبيقها باستخدام أدوات البيانات الأخرى مثل DataList و Repeater وغيرهم من الأدوات التي سنناقشها بالتفصيل ضمن فصول قادمة . إن أدوات التحكم بالبيانات تمثل جزء كبير من عالم تطوير مواقع وتطبيقات الانترنت , وهذا الفصل سيكون قاعدة نعتمد عليها أثناء الفصول القادمة حتى لانكرر نفس الأفكار , لذلك أرجو تطبيق واستيعاب جميع الأفكار والأمثلة التي تمت مناقشتها . والله ولي التوفيق .

الفصول التي صدرت حتى الآن :

- 1- الفصل الأول : استخدام أدوات التحكم القياسية
- 2- الفصل الثاني : استخدام أدوات التحكم المتقدمة
- 3- الفصل الثالث : استخدام أدوات التحقق
- 4- الفصل الرابع : استخدام صفحات الماستريينج
- 5- الفصل الخامس : استخدام الـ Themes
- 6- الفصل السادس : أساسيات التعامل مع قواعد البيانات
- 7- الفصل السابع : استخدام قوائم التكرار
- 8- الفصل الثامن : استخدام الأداة GridView

لتحميل جميع أجزاء السلسلة المتوفرة :

<https://www.mediafire.com/folder/u44z9omx9ajjv/asp.net>

http://www.4shared.com/folder/i34_ICo0/ASP.html

أو البحث في Google عن : محمد عمر الحاج خلف -أو- ASP.NET خطوة بخطوة حتى الاحتراف

للتواصل :

m-hajj Khalaf@hotmail.com

<http://www.facebook.com/mohammed.hajkhalaf>

محمد عمر الحاج خلف - سوريا