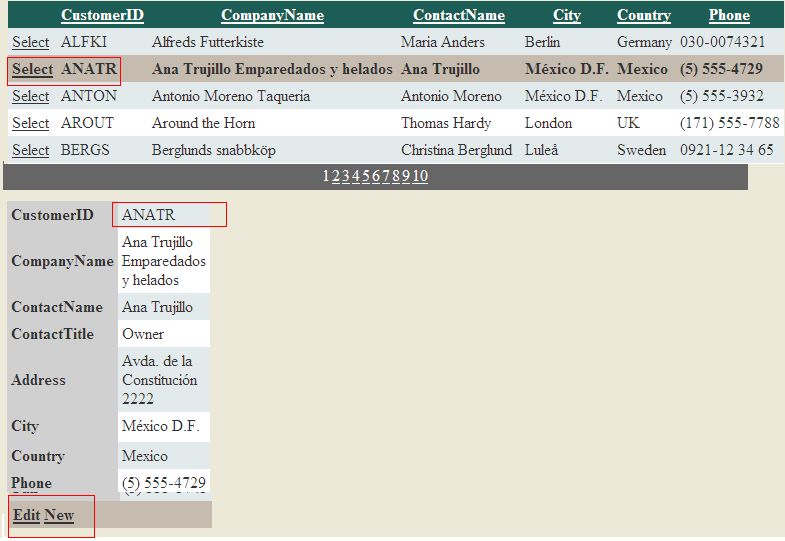
En vez de usar DetailView usaria javascript

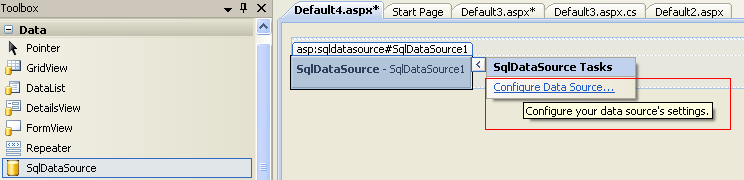
La ventaja de DetailView es que es mas rapido de programar, pero sera mas lento para ejecutarse.



In this exercise, you create a new website and add the database and data source control. You then add the data-bound controls and configure them accordingly.

1. Add the northwnd.mdf file to your **App\_Data** directory.

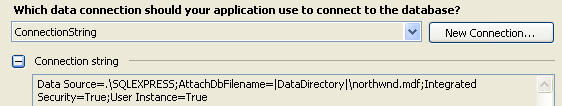
2. Open Default.aspx, add a **SqlDataSource** control to the page from the Toolbox; name it **SqlDataSourceReadList**. This control will read data for display by the GridView control.



3. In Design view of the Default.aspx page, click the smart tag in the upper-right corner of the SqlDataSource control to launch the Configure Data Source Wizard.

a. On the first page, set the connection to the northwnd.mdf file in the App\_Data directory and click Next.

b. When prompted, save the connection string as **ConnectionStringNorthwind**, and then click Next again. In this case in the ConnectionString is already the connection to northwnd.mdf



c. On the Configure Select Statement page, select the Customers table from the Name list box. Select the

- CustomerID,

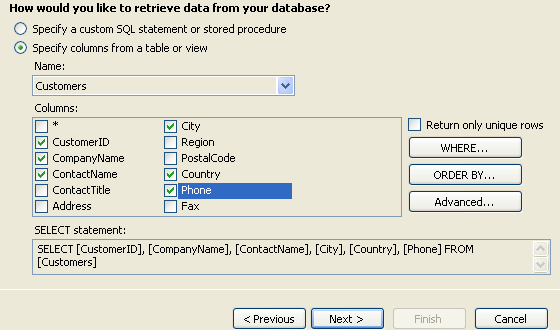
- CompanyName,

- ContactName,

- City,

- Country,

- Phone.



Click Next, and then click Finish to close the wizard. Your SqlDataSource markup should look similar to the following.

<asp:SqlDataSource ID="**SqlDataSource1**" runat="server"

ConnectionString="<%$ ConnectionStrings:ConnectionString %>"

SelectCommand="SELECT [CustomerID], [CompanyName], [ContactName],

[City], [Country], [Phone] FROM [Customers]">

</asp:SqlDataSource>

5. Drag a GridView control onto the Default.aspx page. Using either Design or Source view, configure the GridView control as follows:

■ DataSourceId = SqlDataSourceReadList, created previously.

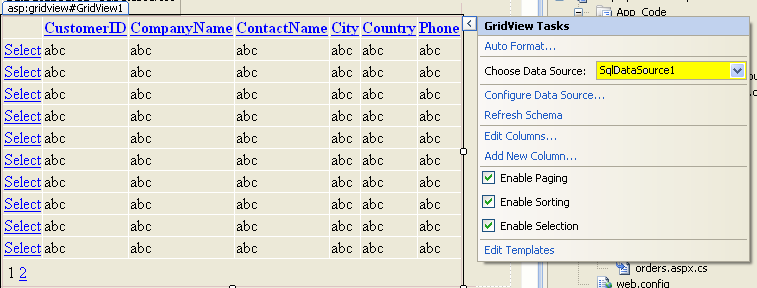
■ AllowPaging = True

■ AllowSorting = True

■ AutoGenerateColumns = False

■ Configure the CustomerID, CompanyName, ContactName, City, Country, and Phone fields to be displayed.

■ Add a CommandField to allow a user to select a row of data (you can do so in Design view by clicking Enable Selection).



Your markup should look similar to the following.

<asp:GridView ID="GridView1" runat="server"

AllowPaging="True" AllowSorting="**True**"

AutoGenerateColumns="**False**"

**DataKeyNames="CustomerID"**

DataSourceID="**SqlDataSource1**"

width="700px">

<Columns>

**<asp:CommandField ShowSelectButton="True" />**

<asp:BoundField DataField="CustomerID" HeaderText="ID" ReadOnly="True"

SortExpression="CustomerID" />

<asp:BoundField DataField="CompanyName" HeaderText="Company" SortExpression="CompanyName" />

<asp:BoundField DataField="ContactName" HeaderText="Contact" SortExpression="ContactName" />

<asp:BoundField DataField="City" HeaderText="City" SortExpression="City" />

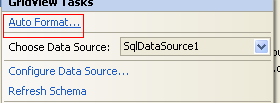
<asp:BoundField DataField="Country" HeaderText="Country" SortExpression="Country" />

<asp:BoundField DataField="Phone" HeaderText="Phone" SortExpression="Phone" />

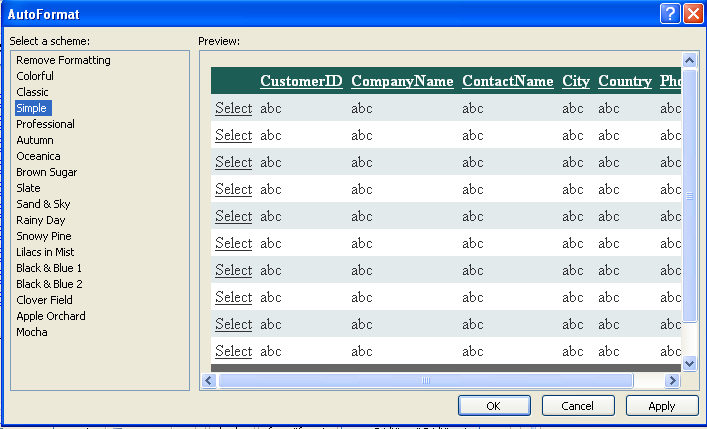
</Columns>

</asp:GridView>

6. If you want, select the GridView in Design view and click the AutoFormat link on the task pane (from the smart tag).



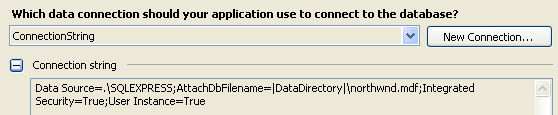
Select Professional or another formatting option.



7. Run the website; you should be able to page through data, sort data, and select a row.

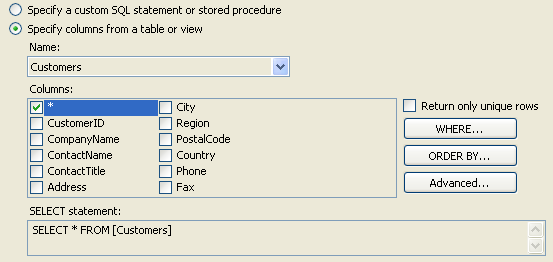
8. Next, add another **SqlDataSource** control to the Default.aspx page; name it **SqlDataSourceUpdate**. Configure this control as before, by using the Configure Data Source Wizard.

a. On the first page, select **ConnectionStringNorthwind** and click Next.



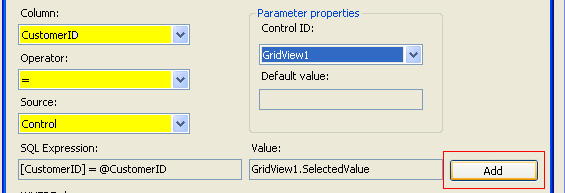
b. The next step is to configure the SELECT statement to pick up the CustomerID parameter from the selected row on the GridView control. To do this follow these steps:

* On the Configure The Select Statement page, select the Customers table.
* This time, select each field in the table.

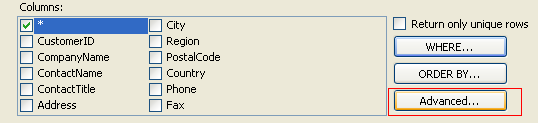


* Then click the **Where** button to launch the Add WHERE Clause dialog box.
* Set the column to **CustomerID**. Set the Operator to **=** (the equal sign).
* Set the Source to **Control**.
* Under Parameter Properties, set the Control ID to **GridView1**.

Click the Add button, and then click OK to close the dialog box.



c. Click the Advanced button. In the Advanced Sql Generation Options dialog box, select the Generate **INSERT, UPDATE, And DELETE** Statements option. Click OK to close this dialog box. Click Next, and then click Finish to close the wizard.



Your SqlDataSource control’s markup should look as follows.

<asp:SqlDataSource ID="SqlDataSourceUpdate" runat="server"

ConnectionString="<%$ ConnectionStrings:ConnectionString %>"

DeleteCommand="DELETE FROM [Customers] WHERE [CustomerID] = @CustomerID"

InsertCommand="INSERT INTO [Customers] ([CustomerID], [CompanyName], [ContactName], [ContactTitle], [Address], [City], [Region], [PostalCode], [Country], [Phone], [Fax]) VALUES (@CustomerID, @CompanyName, @ContactName, @ContactTitle, @Address, @City, @Region, @PostalCode, @Country, @Phone, @Fax)"

SelectCommand="SELECT \* FROM [Customers] WHERE ([CustomerID] = @CustomerID)"

UpdateCommand="UPDATE [Customers] SET [CompanyName] = @CompanyName, [ContactName] = @ContactName, [ContactTitle] = @ContactTitle, [Address] = @Address, [City] = @City, [Region] = @Region, [PostalCode] = @PostalCode, [Country] = @Country, [Phone] = @Phone, [Fax] = @Fax WHERE [CustomerID] = @CustomerID">

<SelectParameters>

<asp:ControlParameter ControlID="GridView1" Name="CustomerID"

PropertyName="SelectedValue" Type="String" />

</SelectParameters>

<DeleteParameters>

<asp:Parameter Name="CustomerID" Type="String" />

</DeleteParameters>

<UpdateParameters>

<asp:Parameter Name="CompanyName" Type="String" />

<asp:Parameter Name="ContactName" Type="String" />

<asp:Parameter Name="ContactTitle" Type="String" />

<asp:Parameter Name="Address" Type="String" />

<asp:Parameter Name="City" Type="String" />

<asp:Parameter Name="Region" Type="String" />

<asp:Parameter Name="PostalCode" Type="String" />

<asp:Parameter Name="Country" Type="String" />

<asp:Parameter Name="Phone" Type="String" />

<asp:Parameter Name="Fax" Type="String" />

<asp:Parameter Name="CustomerID" Type="String" />

</UpdateParameters>

<InsertParameters>

<asp:Parameter Name="CustomerID" Type="String" />

<asp:Parameter Name="CompanyName" Type="String" />

<asp:Parameter Name="ContactName" Type="String" />

<asp:Parameter Name="ContactTitle" Type="String" />

<asp:Parameter Name="Address" Type="String" />

<asp:Parameter Name="City" Type="String" />

<asp:Parameter Name="Region" Type="String" />

<asp:Parameter Name="PostalCode" Type="String" />

<asp:Parameter Name="Country" Type="String" />

<asp:Parameter Name="Phone" Type="String" />

<asp:Parameter Name="Fax" Type="String" />

</InsertParameters>

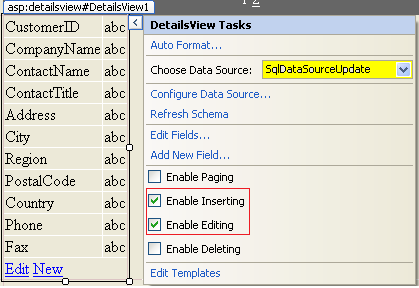
</asp:SqlDataSource>

9. Add a **DetailsView** control to the page; place it under the **GridView** control.

10. Switch to Design view and configure the **DetailsView** control by using the smart tag and related task list.

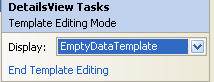
11. Set the control’s data source to **SqlDataSourceUpdate**.

12. Enable **inserting and editing**. (Deleting requires management of a foreign key constraint, so leave that cleared for this example.)

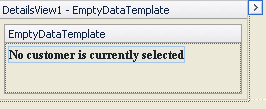


13. Click Edit Templates from the task list of the DetailsView control.

14. Select the **EmptyData Template** from the Display list in the task list.

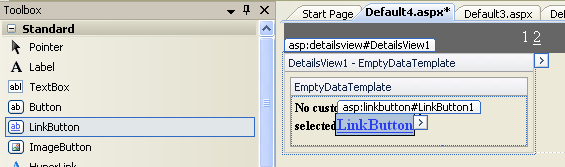


15. In the template, type **No customer is currently selected**.

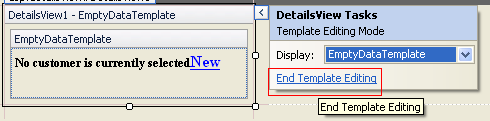


16. Add a **LinkButton** control to the template. Set the LinkButton control’s properties

* **CausesValidation** to false
* CommandName to **New**,
* Text to **New**.



17. In the DetailView Tasks window, click End Template Editing.



Your DetailsView markup should look as follows.

<**asp:DetailsView** ID="DetailsView1" runat="server"

AutoGenerateRows="False"

DataKeyNames="CustomerID"

DataSourceID="SqlDataSourceUpdate"

Height="50px"

Width="125px">

<EmptyDataTemplate>

<b>

<span style="font-size:9.0pt;font-family:Segoe-Bold;

mso-fareast-font-family:&quot;Times New Roman&quot;;mso-bidi-font-family:Segoe-Bold;

color:black;mso-ansi-language:EN-US;mso-fareast-language:EN-US;mso-bidi-language:

AR-SA">**No customer is currently selected**</span>

<asp:LinkButton ID="LinkButton1" runat="server" CausesValidation="False"

CommandName="New">New

</asp:LinkButton>

</b>

</EmptyDataTemplate>

<Fields>

<asp:BoundField DataField="CustomerID" HeaderText="CustomerID" ReadOnly="True"

SortExpression="CustomerID" />

<asp:BoundField DataField="CompanyName" HeaderText="CompanyName" SortExpression="CompanyName" />

<asp:BoundField DataField="ContactName" HeaderText="ContactName" SortExpression="ContactName" />

<asp:BoundField DataField="Address" HeaderText="Address" SortExpression="Address" />

<asp:BoundField DataField="City" HeaderText="City" SortExpression="City" />

<asp:BoundField DataField="Country" HeaderText="Country" SortExpression="Country" />

<asp:BoundField DataField="Phone" HeaderText="Phone" SortExpression="Phone" />

<asp:CommandField ShowEditButton="True" ShowInsertButton="True" />

</Fields>

</asp:DetailsView>

18. If you want, select the DetailsView control in Design view and click the AutoFormat link in the task pane (from the smart tag). Select Professional or another formatting option.

19. Next, add code to update the GridView when a record has been inserted or edited in the DetailsView control. To do so, add event handlers for both the **ItemUpdated** and **ItemInserted** events of the DetailsView control. Inside each event, rebind the GridView control. The following code shows an example.

protected void DetailsView1\_**ItemUpdated**(object sender, DetailsViewUpdatedEventArgs e)

{

GridView1.DataBind();

}

protected void DetailsView1\_**ItemInserted**(object sender, DetailsViewInsertedEventArgs e)

{

GridView1.DataBind();

}

20. Add a title to the top of the page for **Manage Customers**. Add another title for **Customer Details**.

21. Run the webpage. Notice that the empty DetailsView control allows you to add a new record. Select a row from the GridView. Notice that it appears in the DetailsView section.

