

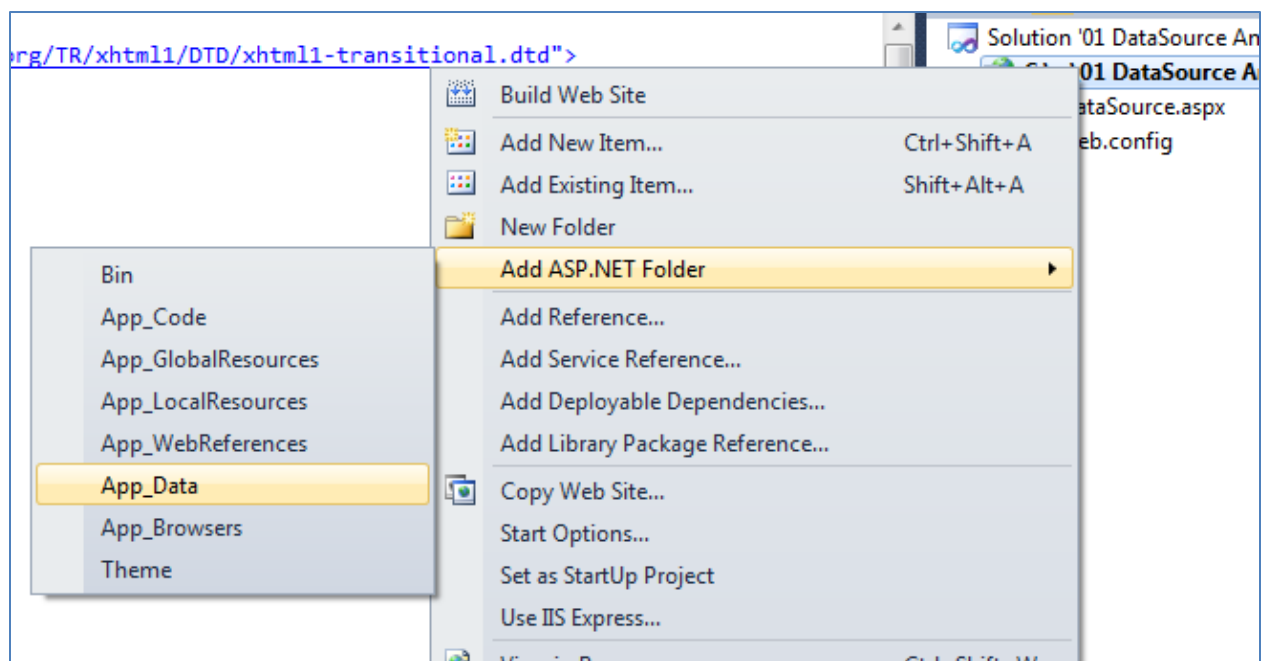
DATASOURCE

ניתן להשתמש במספר בסיסי נתונים ולכן יש מספר מחלקות שמקשרות בין הקוד שלנו בC# לבין בסיסי הנתונים. ישנם DataSource טבלאיים וישנם היררכיים.

[01 DataSource And GridView]

[01 DataSource.aspx]

הראשון שנראה היה ה SqlDataSource שמתחבר לבסיס נתונים SQL.
אפשרות פשוטה היא למשוך את קובץ ה MDF לתוך ספריית ה App_Data.



ניתן לראות את בסיסי הנתונים דרך ה VS.

יצירת ה DS:

1. מושך SqlDS
2. מצביע על ה connectionString הרצוי
3. בוחר את הטבלה והשדות המתאימים
4. finish

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

ניתן למצוא את ה connectionString בתוך הקובץ web.config

```
<connectionStrings>
  <add name="MovieConStr" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\VideoLibProjectV01.mdf;Integrated
Security=True;User Instance=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

ניתן לחבר פקד ל DS ע"י קוד בצורה מפורשת למוחלטת imperative או בצורה הצהרתית declarative למשל declarative:

```
<asp:DropDownList ID="DropDownList1" runat="server"
  DataSourceID="SqlDataSource1" DataTextField="MovieName"
  DataValueField="MovieID"
  onselectedindexchanged="DropDownList1_SelectedIndexChanged"
  AutoPostBack="True">
</asp:DropDownList>
```

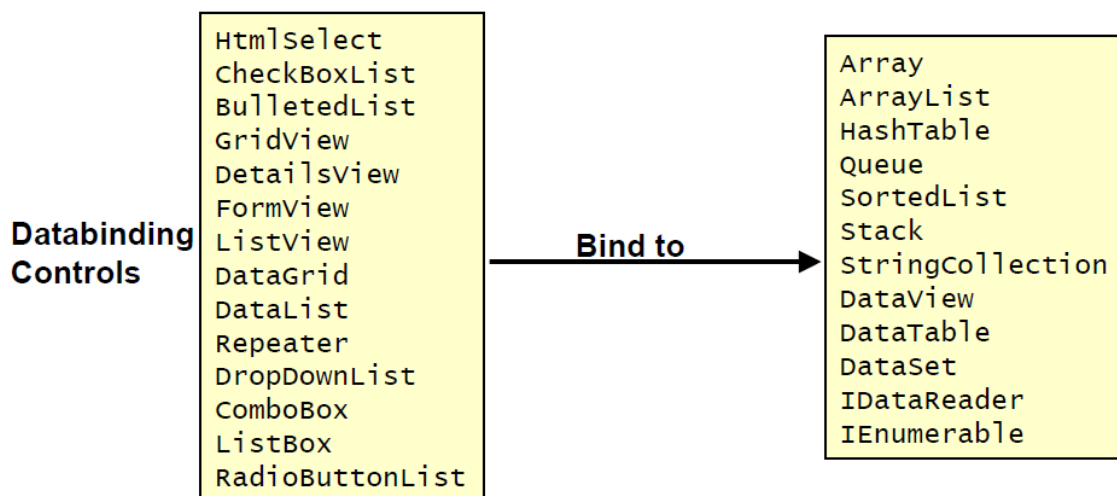
או בצורה imperative:

```
<asp:DropDownList ID="DropDownList2" runat="server"
  onselectedindexchanged="DropDownList2_SelectedIndexChanged"
  AutoPostBack="True">
</asp:DropDownList>
```

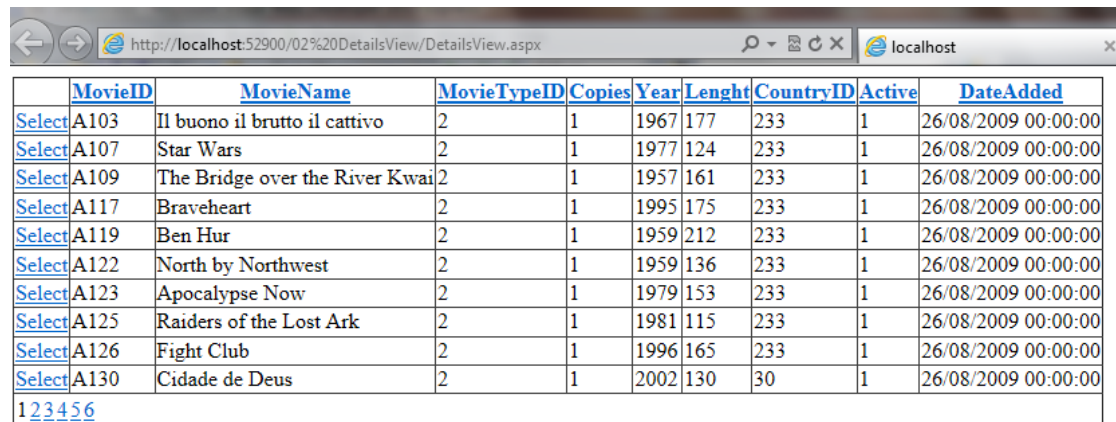
+

```
if (!IsPostBack)
{
  DropDownList2.DataSource = SqlDataSource1;
  DropDownList2.DataValueField = "MovieID";
  DropDownList2.DataTextField = "MovieName";
  DropDownList2.DataBind();
}
```

עוד אפשרויות שניתן לחבר כל עוד ישנו המימוש של IEnumerable



GRIDVIEW



	<u>MovieID</u>	<u>MovieName</u>	<u>MovieTypeID</u>	<u>Copies</u>	<u>Year</u>	<u>Lenght</u>	<u>CountryID</u>	<u>Active</u>	<u>DateAdded</u>
Select	A103	Il buono il brutto il cattivo	2	1	1967	177	233	1	26/08/2009 00:00:00
Select	A107	Star Wars	2	1	1977	124	233	1	26/08/2009 00:00:00
Select	A109	The Bridge over the River Kwai	2	1	1957	161	233	1	26/08/2009 00:00:00
Select	A117	Braveheart	2	1	1995	175	233	1	26/08/2009 00:00:00
Select	A119	Ben Hur	2	1	1959	212	233	1	26/08/2009 00:00:00
Select	A122	North by Northwest	2	1	1959	136	233	1	26/08/2009 00:00:00
Select	A123	Apocalypse Now	2	1	1979	153	233	1	26/08/2009 00:00:00
Select	A125	Raiders of the Lost Ark	2	1	1981	115	233	1	26/08/2009 00:00:00
Select	A126	Fight Club	2	1	1996	165	233	1	26/08/2009 00:00:00
Select	A130	Cidade de Deus	2	1	2002	130	30	1	26/08/2009 00:00:00

1 2 3 4 5 6

[02 GridView.aspx]

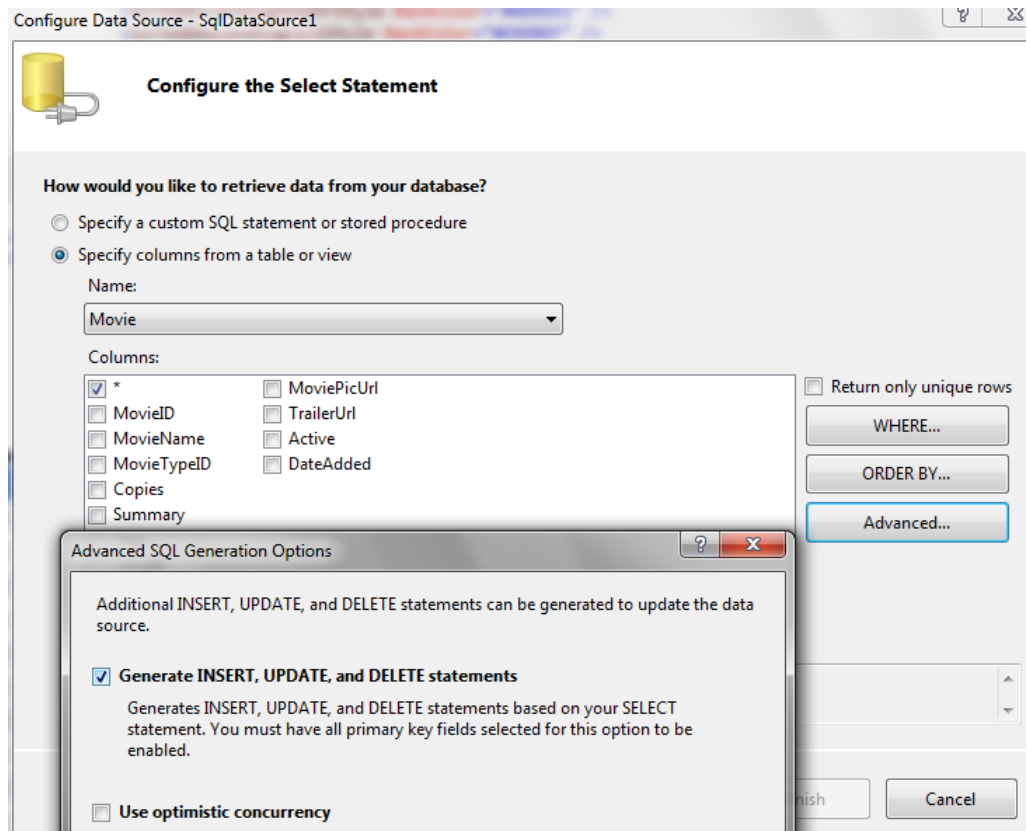
אחד הפקדים החזקים ביותר להצגה של מידע טבלאי הוא ה GridView אותו נציג עכשיו.

[02_2 GridView.aspx]

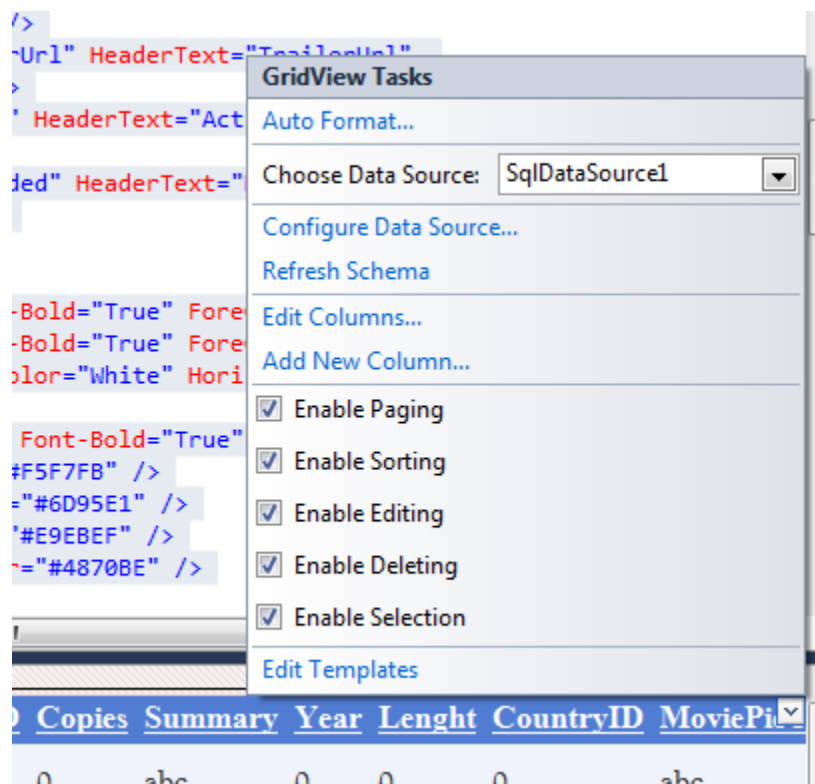
ניתן לערוך את אופן הצגתו, את העמודות שיוצגו, אפשר להוסיף עוד עמודות של עריכה...

[02_3 GridView.aspx]

ניתן עוד להרחיב את הפקד ולהשתמש בו לעריכת התוכן אבל קודם עלינו להרחיב את היכולות של ה DataSource



עכשיו נוכל לסמן גם את האפשרויות הבאות בתוך ה GV



- ברגע ששניתי את העמודות של ה GV אני חייב גם לערוך את הקוד של ה DS כי חלק מהעמודות של ה UpdateCommand וה InsertCommand כבר לא נימצאות על הדף ולכן לא ניתן לקרוא אותן מה שיביא לexception.
- ל GV יש
 - ImageField
 - CheckBoxField
 - HyperLinkField
- במקום כל מה שעשינו עד כה – ניתן היה פשוט למשוך טבלה מתוך ה VS כאשר בוחרים מתוך ה DatabaseExplorer. ואז נוצרים גם DS וגם GV עבורינו.
- כל מה שראינו עובד בצורה אוטומטית רק כל עוד לא שינינו את המאפיין של sqlDataSource

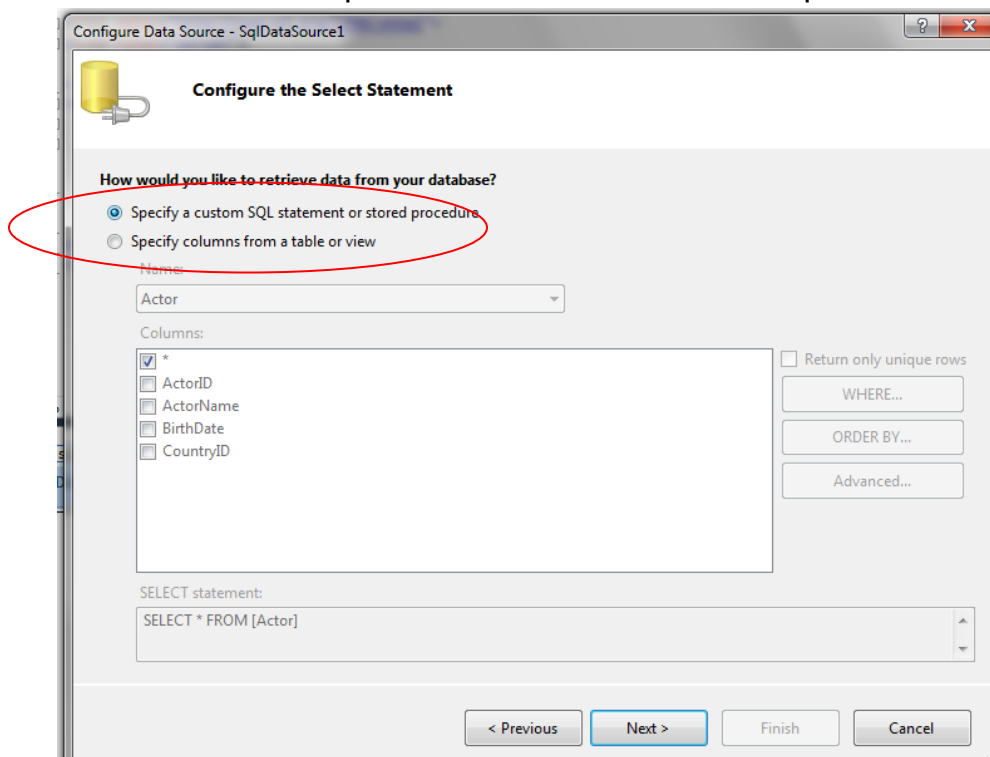
`DataSourceMode="DataReader"`

ברירת המחדל היא

`DataSourceMode="DataSet"`

[02_4 GridView.aspx]

בדוגמה הבאה נראה איך אפשר להציג יותר מטבלה אחת בתוך אותו GV



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

SELECT UPDATE INSERT DELETE

☒ SQL statement:

Query Builder...

☐ Stored procedure:

Query Builder

Movie
☐ * (All Columns)
☒ MovieID
☐ MovieName

Column	Alias	Table	Outp...	Sort Type	Sort O

SELECT
FROM Movie

Execute Query

Add Table

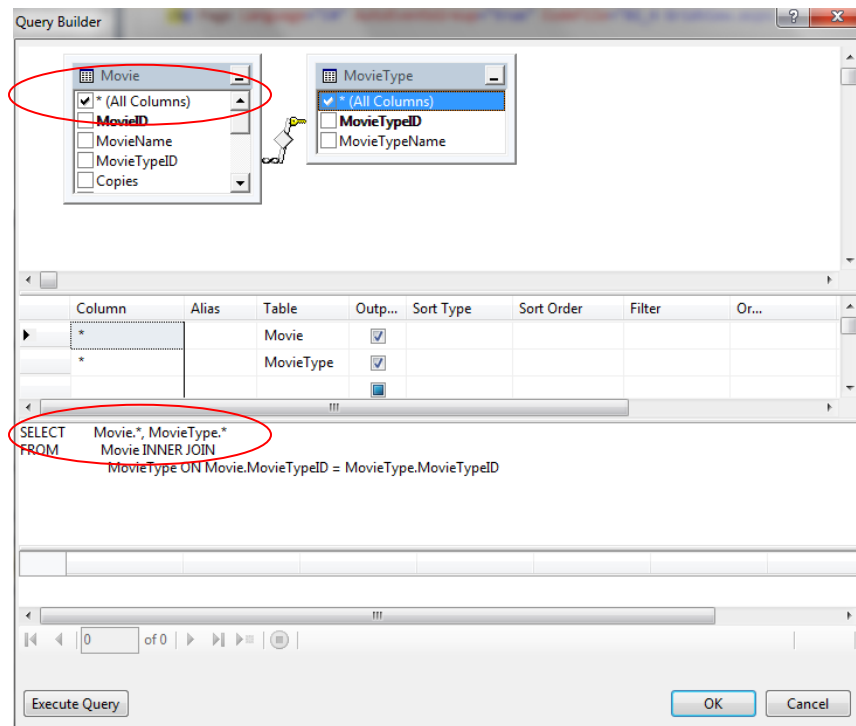
Tables Views Functions Synonyms

Actor
ActorMovie
Country
Member
Movie
MovieType
popular
Rent
Review
WaitingList

Refresh

Add

Close



[02 DetailsView]

לדוגמה: פה רואים שכאשר ניבחר סרט נוכל לראות עבורו בטבלה נוספת את כל ה reviews שעשו עליו, כאשר בכל פעם רואים רק review אחד.

ניתן גם לעשות Insertion !!!

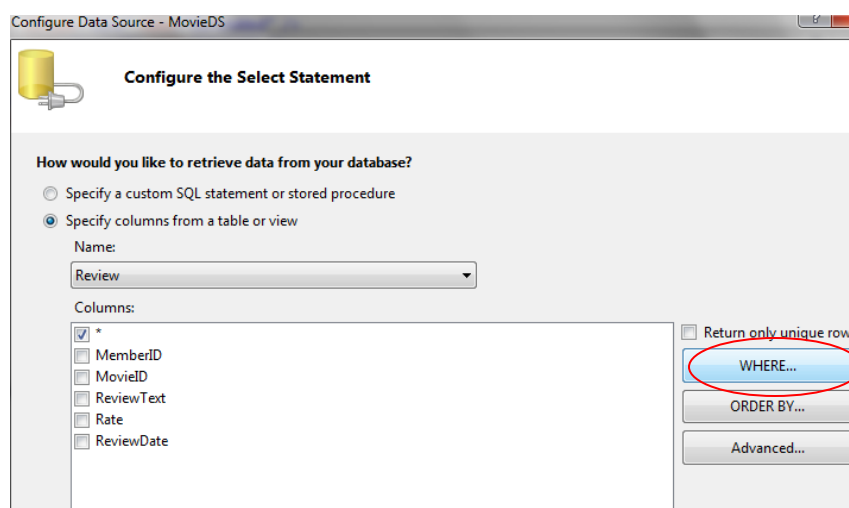
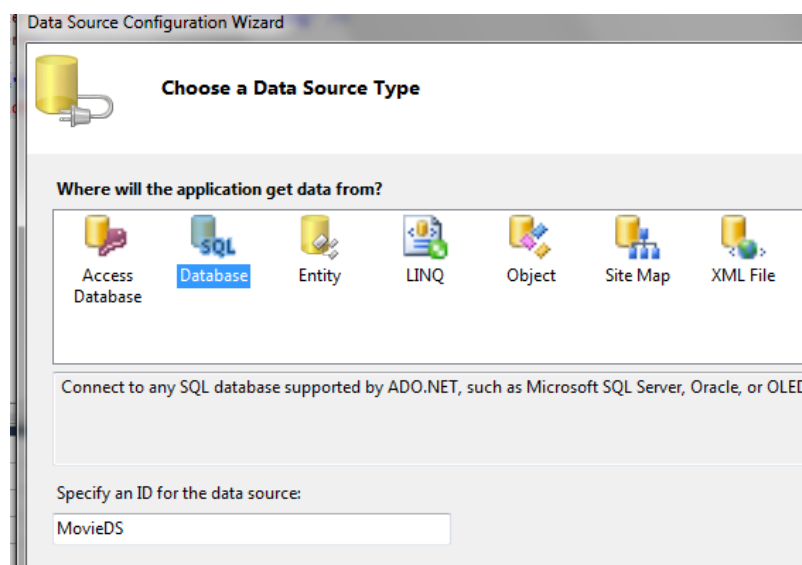
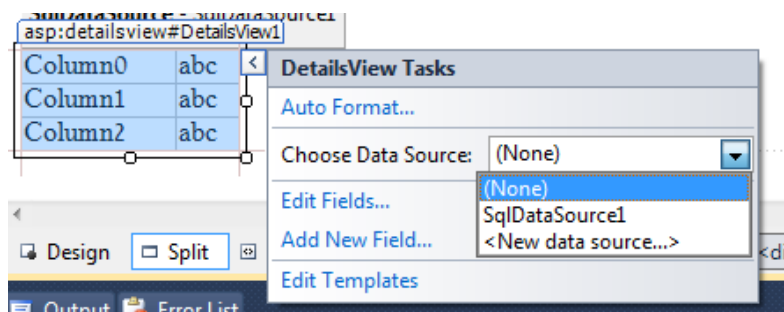
	MovieID	MovieName	MovieTypeID	Copies	Year	Lenght	CountryID	Active	DateAdded
Select	A103	Il buono il brutto il cattivo	2	1	1967	177	233	1	26/08/2009 00:00:00
Select	A107	Star Wars	2	1	1977	124	233	1	26/08/2009 00:00:00
Select	A109	The Bridge over the River Kwai	2	1	1957	161	233	1	26/08/2009 00:00:00
Select	A117	Braveheart	2	1	1995	175	233	1	26/08/2009 00:00:00
Select	A119	Ben Hur	2	1	1959	212	233	1	26/08/2009 00:00:00
Select	A122	North by Northwest	2	1	1959	136	233	1	26/08/2009 00:00:00
Select	A123	Apocalypse Now	2	1	1979	153	233	1	26/08/2009 00:00:00
Select	A125	Raiders of the Lost Ark	2	1	1981	115	233	1	26/08/2009 00:00:00
Select	A126	Fight Club	2	1	1996	165	233	1	26/08/2009 00:00:00
Select	A130	Cidade de Deus	2	1	2002	130	30	1	26/08/2009 00:00:00

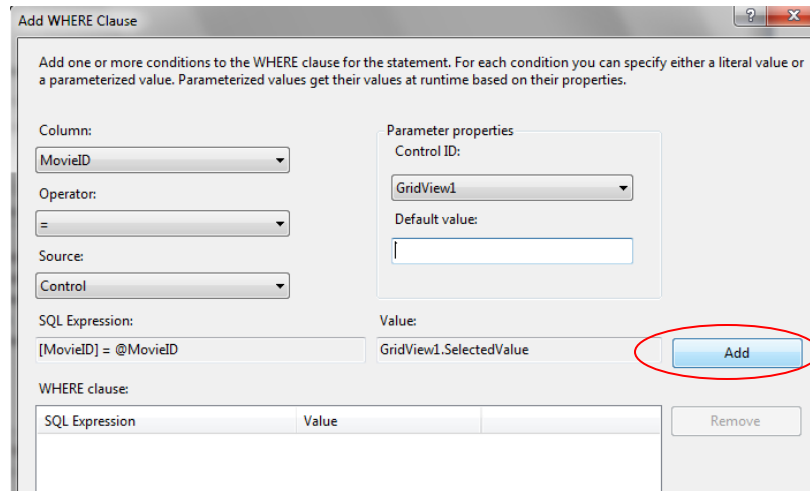
123456

MemberID	21
MovieID	A109
ReviewText	my text bla
Rate	3/ <div>ReviewText</div>
ReviewDate	

Insert Cancel

איך נעשה זאת:





לתת את EX1

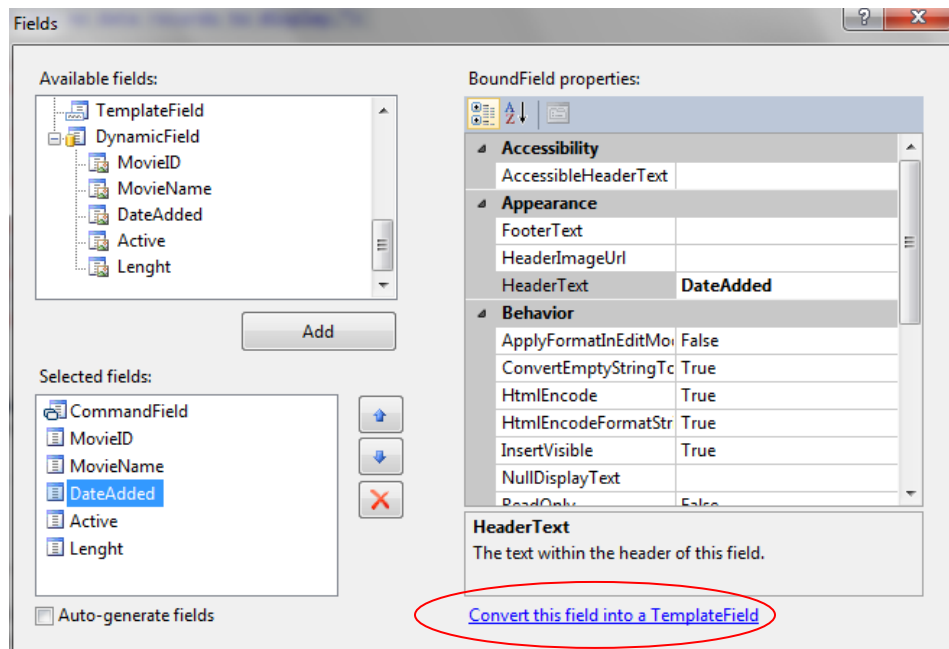
TWO WAY BINDING

[03 Eval And Bind 2-way Binding]

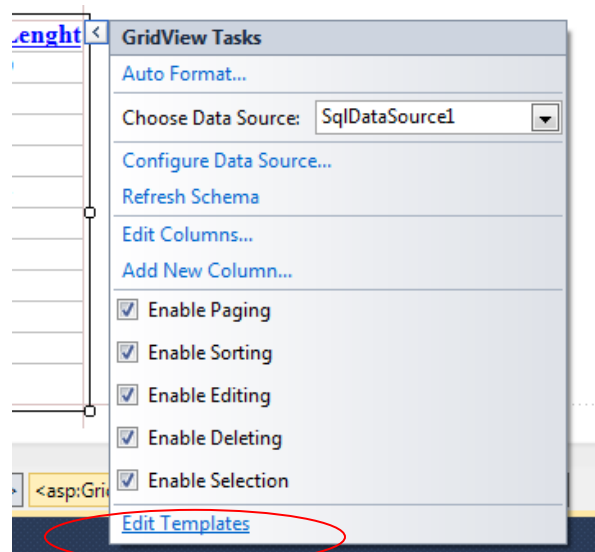
ראינו שבעזרת האשף או הMarkup ניתן לעצב את הפקדים ע"י `<asp:CommandField>` או `<asp:BoundField>` כמו כן ניתן גם לייצר שדות ע"י `<asp:TemplateField>` שבהם ניתן לעשות עיצוב יותר חופשי. על מנת להשתמש בהם כדאי להכיר את שתי הפונקציות המיוחדות שעוזרות להציג או לערוך את התוכן של השדות של הפקד. בעזרת הפונקציה `<%# Eval(FiledName)%>` ניתן עבור כל שורה לקבל את הערך של אותו השדה עבור קריאה בלבד (קיצור של evaluation). אם צריך קריאה וכתיבה חזרה ל DS אז ניתן להשתמש בפונקציה `<%# Bind(FieldName)%>`. הערכים הללו מחושבים כאשר ה DataBiding מתבצע. הם מחזירים ערך עבור כל שורה שנמצאת ב DS.

```
<%# data-binding expression %>
<%$ expressionPrefix: expressionValue %>
<%@ %> Directive Syntax
<%-- --%> Server-Side Comments
```

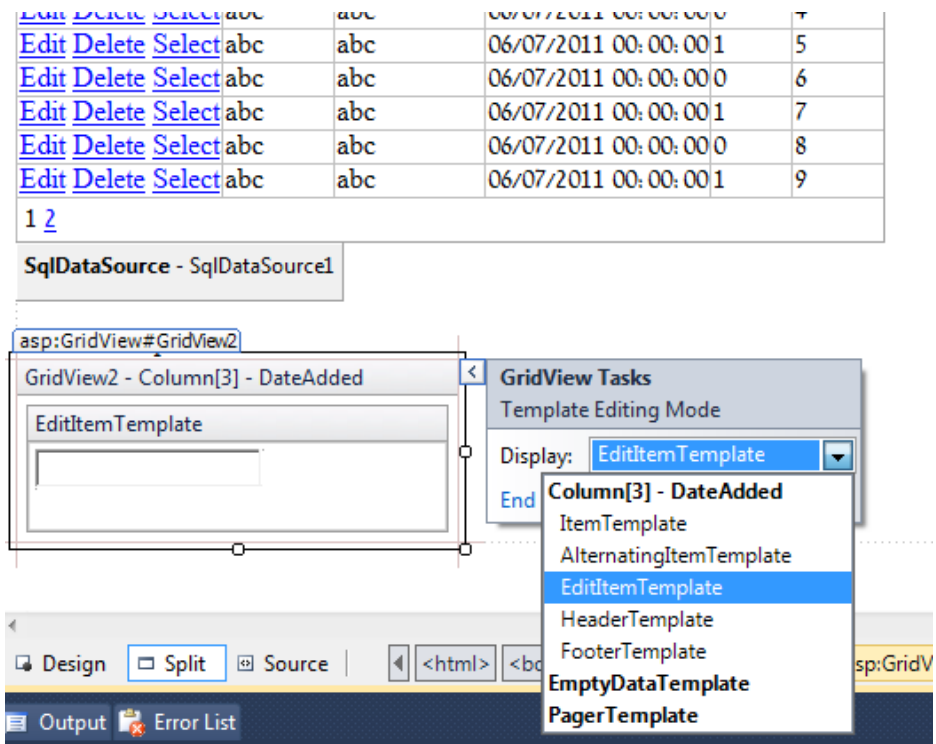
פשוט יותר יהיה לראות את הדוגמה הבאה:



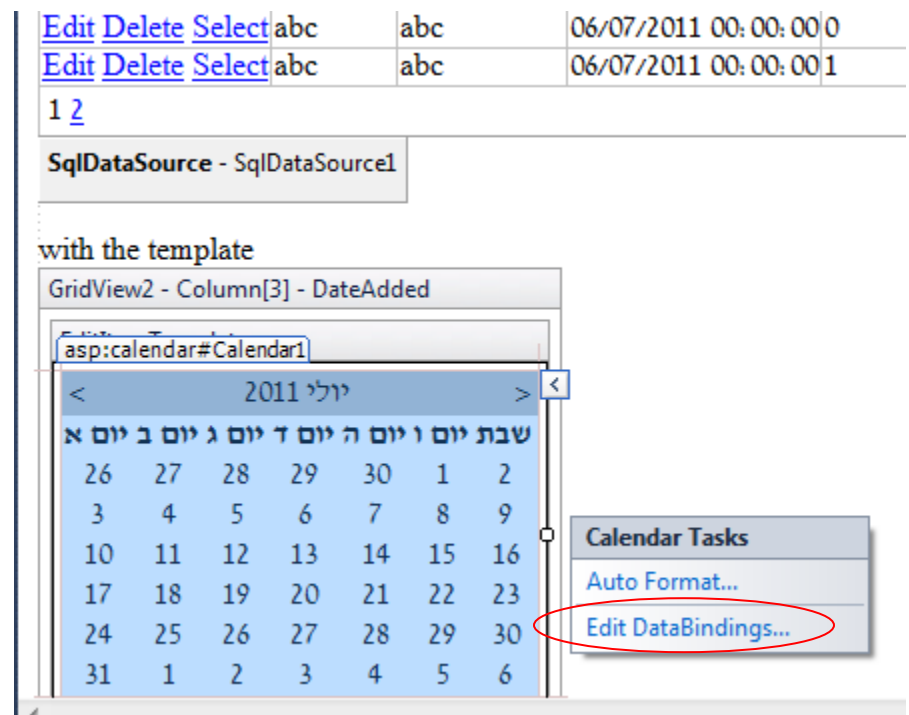
בחירת השדה והמרה ל TemplateFiled



בחירה של עריכה לכל ה TemplateField הקיימים



בחירה ספציפית עבור השדה וסוג התמלול למשל עבור הצגה/עריכה
עכשיו העידכון עצמו למשל: צריך למחוק את תיבת הטקסט ולהחליפה ב/ calendar



לבסוף קישור בין השדה לבסיס הנתונים. האם חד או דו צדדי?

Calendar1 DataBindings

Select the property to bind to. You can then bind it by selecting a field. Alternatively, you can bind it using a custom code expression.

Bindable properties:

- Enabled
- SelectedDate
- Visible
- VisibleDate

☐ Show all properties

Refresh Schema

Binding for SelectedDate

☒ Field binding:

Bound to: DateAdded

Format:

Sample:

☒ Two-way databinding

☐ Custom binding:

Code expression: Bind("DateAdded")

OK Cancel

Calendar1 DataBindings

Select the property to bind to. You can then bind it by selecting a field. Alternatively, you can bind it using a custom code expression.

Bindable properties:

- Enabled
- SelectedDate
- Visible
- VisibleDate

☐ Show all properties

Refresh Schema

Binding for VisibleDate

☒ Field binding:

Bound to: DateAdded

Format:

Sample:

☐ Two-way databinding

☐ Custom binding:

Code expression: Eval("DateAdded")

OK Cancel

וכן נוצר לנו הדבר הבא:

```

<asp:TemplateField HeaderText="DateAdded" SortExpression="DateAdded">
    <EditItemTemplate>
        <asp:Calendar ID="Calendar1" runat="server"
            SelectedDate='<%# Bind("DateAdded") %>' VisibleDate='<%# Eval("DateAdded") %>'>
        </asp:Calendar>
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label1" runat="server" Text='<%# Bind("DateAdded") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>

```

שימו לב שה – ItemTemplate קיבל גם כן Bind למרות שאפשר היה לשים Eval גם כן וזה היה עובד אותו דבר.

FORMVIEW

[04 FormView]

FormView דומה ל DetailsView אבל הוא לא מכיל עיצוב בכלל אלא המתכנת משתמש בTemplate בכדי לעצב את השדות כרצונו. מתי יהיה כדאי להשתמש בו אם כך? כאשר אנחנו צריכים בכל מקרה להוסיף פקדים לאילו שקיימים, למשל פקדי אימות, אז יותר פשוט להשתמש בFormView. (ב DetailsView הוספת פקדים נוספים לעיצוב נעשה כמו בדוגמה הקודמת שבה ראינו איך מעצבים שדה של GV באופן מותאם אישית. ניתן ליהיה לראות הרבה templateFiled עם מספר XXXItemTemplate בכל אחד מהם - מסורבל) נראה שבעזרת FromView זה יראה קצת יותר מסודר. עכשיו נראה דוגמה לשימוש בו:

אנחנו רוצים שוב להציג calendar במקום סתם תיבת טקסט.

היה צריך רק להעתיק את הקיים ב DateAdded שנמצא מעל ולשנות לפקד calendar כמו שרואים בשורות המסומנות. כל השאר נעשה אוטומטית ע"י שימוש ב EVAL ו BIND ולכן קל לערוך ב Markup

```

<asp:FormView ID="FormView1" runat="server" DataKeyNames="MovieID"
DataSourceID="SqlDataSource1">
  <EditItemTemplate>
    MovieID:
    <asp:Label ID="MovieIDLabel1" runat="server" Text='<%# Eval("MovieID") %>' />
    <br />
    MovieName:
    <asp:TextBox ID="MovieNameTextBox" runat="server"
      Text='<%# Bind("MovieName") %>' />
    <br />
    DateAdded:
    <asp:TextBox ID="DateAddedTextBox" runat="server"
      Text='<%# Bind("DateAdded") %>' />
    <br />
    DateAdded2:
    <asp:Calendar runat="server" ID="calendar" SelectedDate='<%# Bind("DateAdded") %>' VisibleDate='<%# Eval("DateAdded") %>' />
    <br />
    Lenght:
    <asp:TextBox ID="LenghtTextBox" runat="server" Text='<%# Bind("Lenght") %>' />
    <br />
  </EditItemTemplate>
</asp:FormView>

```

זזה יראה כך בסופו של דבר:

Leon

MovieID: A133

MovieName: Leon

DateAdded(was created automatically): 26/08/2009 00:00:00

DateAdded2(was created manually):

אוגוסט 2009						
שבת	יום ו'	יום ד'	יום ג'	יום ב'	יום א'	
1	31	30	29	28	27	26
8	7	6	5	4	3	2
15	14	13	12	11	10	9
22	21	20	19	18	17	16
29	28	27	26	25	24	23
5	4	3	2	1	31	30

Lenght: 110

Active: 1

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

זהו הפקד החדש ביותר (3.5) והמקיף ביותר, באמצעותו ניתן להחליף כל פקד אחד.

הוא מכיל שני מקטעים: `LayoutTemplate` ו `ItemTemplate`.

:LayoutTemplate

מכיל כל פקד שנרצה לצורה הכללית של התוצאה כמו DIV, TABLE וכו... אין לפקד הזה שום עיצוב

דיפולטיבי. בתוך החלק הזה ניתן להגדיר פקד בשם `asp:Placeholder` שיש לו ID ומטרתו היא רק

לסמן את המקום שלו בכדי שיוחלף ע"י ה ItemTemplate

[05 ListView]

בדוגמה הבא ניתן לראות כמה קל לעצב את התוצאה:

[01 ListViewDemo1.aspx]

```
<asp:ListView ID="LV1" runat="server" DataSourceID="SqlDataSource1" ItemPlaceholderID="myplaceholder">  
    <LayoutTemplate>  
        <h1>This Is My ListView</h1>  
        <asp:PlaceHolder runat="server" ID="myplaceholder"/>  
        <h2>Here is the End Of IT</h2>  
    </LayoutTemplate>  
    <ItemTemplate>  
        <div><%#Eval("MovieID") %>      --&nbsp&nbsp&nbsp  
            <asp:TextBox runat="server" ID="txt1" Text='<%#Eval("MovieName")%>' />  
        </div>  
    </ItemTemplate>  
</asp:ListView>
```

The diagram illustrates the relationship between the placeholder ID defined in the code-behind and the actual ASP.NET server control. One arrow originates from the "myplaceholder" value assigned to the "ItemPlaceholderID" attribute of the "<asp:ListView>" element at the top of the snippet. The other arrow points directly to the "<asp:PlaceHolder runat=

התוצאה:



This Is My ListView

A103	--	Il buono il brutto il cattivo
A107	--	Star Wars
A109	--	The Bridge over the River
A117	--	Braveheart
A119	--	Ben Hur
A122	--	North by Northwest
A123	--	Apocalypse Now
A125	--	Raiders of the Lost Ark

[02 ListViewWithCSSAndSort.aspx]

אחד הדברים המובהקים בהם הפקד נח יותר מאחרים הוא אפשרות לעצב אותו בנוחות ע"י CSS.

מכיוון שהפקד מייצר (דרך האשף) את כל הקוד ב MarkUp, בצורה מאוד נוחה ניתן לגשת לאיזורים

מסויימים ולהגדיר בהם את העצוב ה-CSS-י המצאים.

באופן אוטומטי ניתן להוסיף Edit, Delete, Insert, Select אבל מיון ניתן באמצעות הוספת כפתור עם

```
CommandName="Sort"
```

למשל כך:


```

<th runat="server">
  <asp:LinkButton runat="server" ID="lbMovieID" CommandName="Sort" CommandArgument="MovieID">MovieID Sort</asp:LinkButton>
  MovieID</th>
<th runat="server">
  <asp:LinkButton runat="server" ID="LinkButton1" CommandName="Sort" CommandArgument="MovieName">MovieName Sort</asp:LinkButton>
  MovieName</th>
<th runat="server">

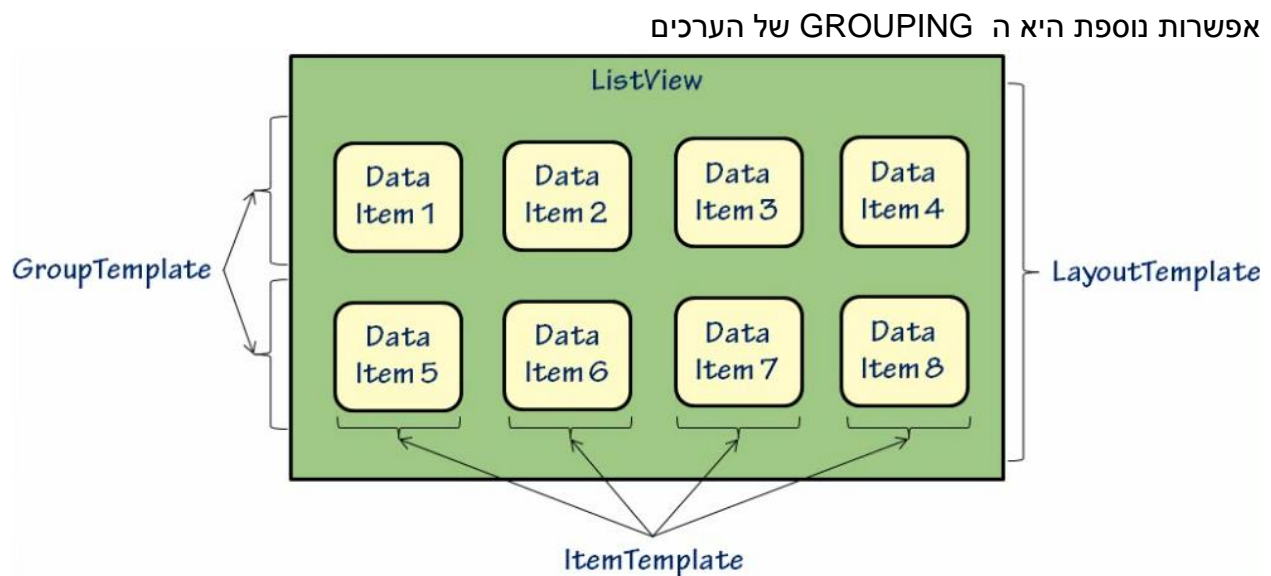
```

שימו לב שנוצר פקד שלם שנקרא DataPager שאחראי על המעברי עמודים. את הפקד הזה ניתן גם כן לעצב למשל ניתן להחליט כמה שורות יופיעו בכל עמוד ע"י `PageSize="5"`

```

<asp:DataPager ID="DataPager1" runat="server" PageSize="5">

```



הכי פשוט זה דרך האשף לבחור Layout כ Tiled. את מספר היחידות בשורה ניתן לעדכן באצעות:

```

<asp:ListView ID="ListView1" runat="server" DataKeyNames="MovieID"
  DataSourceID="SqlDataSource1" InsertItemPosition="LastItem"
  GroupItemCount="5">

```

את מספר השורות לא ניתן לעדכן ישירות אלה ניתן לעדכן את סה"כ האלמנטים שמוצגים ע"י

```

<asp:DataPager ID="DataPager1" runat="server" PageSize="15">

```