



## الفصل الرابع عشر: التعامل مع قواعد البيانات برمجياً

الصفحة	العنوان
3	<b>1. التعامل مع قواعد البيانات برمجياً</b>
4	<b>1.1</b> التعامل مع قواعد البيانات برمجياً
6	<b>2. مثال تطبيقي</b>
13	<b>3. عنصر التحكم Repeater</b>
14	<b>1.3</b> استخدام عنصر التحكم Repeater
17	<b>4. مجموعات البيانات Datasets</b>
18	<b>1.4</b> الصف SqlDataAdapter
18	<b>2.4</b> الصف DataSet
22	<b>5. مثال تطبيقي</b>

## الكلمات المفتاحية

.DataSet ، SqlDataAdapter ، SqlDataReader ، SqlCommand ، SqlConnection

## الملخص

نستعرض في هذا الفصل أهم طرق التعامل مع قواعد البيانات من خلال الكود البرمجي.

## الأهداف التعليمية:

يتعرف الطالب في هذا الفصل على:

- الصف SqlConnection.
- الصف SqlCommand.
- الصف SqlDataReader.
- الصف SqlDataAdapter.
- الصف DataSet.
- عنصر البيانات Repeater.

## المخطط

يضم فصل التعامل مع قواعد البيانات برمجياً 5 وحدات (Learning Objects) هي:

- التعامل مع قواعد البيانات برمجياً
- مثال تطبيقي
- عنصر التحكم Repeater
- مجموعات البيانات Datasets
- مثال تطبيقي

## التعامل مع قواعد البيانات برمجياً

### الأهداف التعليمية

- الصفوف الأساسية للتعامل مع قواعد البيانات.

## التعامل مع قواعد البيانات برمجياً

يُمكن برمجياً الاتصال مع قواعد البيانات وتنفيذ العمليات الأساسية عليها (استعلام Select، إضافة Insert، حذف Delete، تعديل Update).

نُعتمد في التعامل مع البيانات SQL Server على مجموعة من الصفوف التي توجد في كل من الفضائين System.Data و System.Data.SqlClient. لذا يجب استيراد هذين الفضائين في بداية ملف الكود الخلفي:

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

### الصف SqlConnection

- يُستخدم غرض من هذا الصف لوصل غرض من الصف SqlCommand مع قاعدة بيانات معينة.
- من أهم خصائص هذا الغرض الخاصية.ConnectionString والتي تُحدد سلسلة الإتصال مع قاعدة البيانات.
- من أهم طرق هذا الغرض الطريقة Open() التي تقوم بفتح اتصال مع قاعدة البيانات المحددة بسلسلة الاتصال، والطريقة Close() التي تقوم بإغلاق الاتصال المفتوح.
- يُستخدم غرض من هذا الصف لوصل غرض SqlConnection مع غرض SqlDataReader أو SqlDataAdapter.
- يسمح الغرض SqlCommand بتنفيذ عبارات SQL أو إجراءات مخزنة Stored Procedures على قاعدة البيانات. يتضمن هذا إعادة مجموعة تسجيلات (حيث نستخدم للوصول إليها غرض آخر كالغرض SqlDataReader أو الغرض SqlDataAdapter)، أو إعادة قيمة وحيدة، أو إعادة عدد العناصر المتأثرة بالاستعلامات التي لا تُعيد مجموعة تسجيلات.
- من أهم طرق هذا الغرض:
  - الطريقة ExecuteReader: تقوم بتنفيذ الأمر المعروف في الخاصية.CommandText والذي يُعيد مجموعة من التسجيلات.
  - الطريقة ExecuteNonQuery: تقوم هذه الطريقة بتنفيذ الأمر المُعرّف في الخاصية.CommandText (استعلام لا يُعيد صفوف مثل تعليمة Insert أو Update أو Delete) وتُعيد عدد التسجيلات التي تأثرت بالاستعلام.
  - الطريقة ExecuteScalar: تقوم بتنفيذ الأمر المعروف في الخاصية.CommandText والذي يُعيد قيمة وحيدة.

- يوفر غرض من هذا الصف طريقة سريعة وفعالة للوصول إلى مجموعة من التسجيلات بشكل أمامي وللقراءة فقط.
- من أهم طرق هذا الغرض:
  - الطريقة Read: تقوم بتحريك مؤشر التسجيلة إلى التسجيلة التالية. وتُعيد false عند الوصول لنهاية التسجيلات.
  - الطريقة GetOrdinal("ColName"): تُعيد فهرس العمود الذي اسمه "ColName".
  - الطريقة GetString (index): تُعيد قيمة العمود ذو الفهرس index كسلسلة نصية.

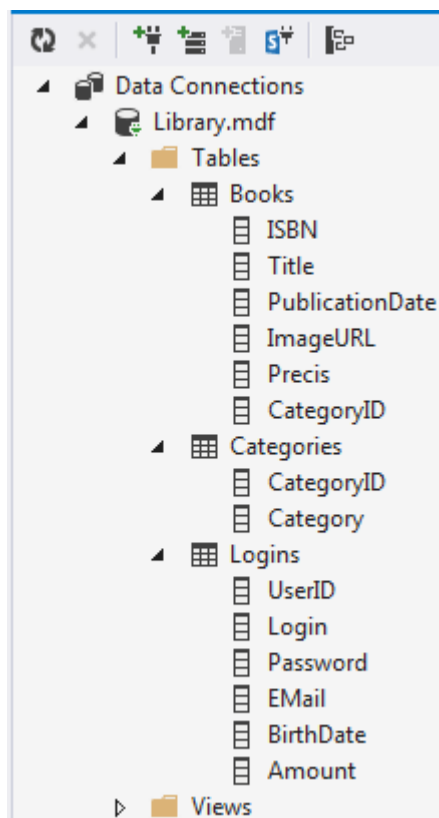
## مثال تطبيقي

### الأهداف التعليمية

- مثال على التعامل مع قواعد البيانات برمجياً.

ليكن لدينا قاعدة البيانات التالية Library.mdf والمنشأة باستخدام SQL Server Express. تحوي هذه القاعدة على ثلاثة جداول:

- جدول الكتب Books
- جدول فئات الكتب Categories
- جدول المستخدمين Logins



لنفرض أننا نريد قراءة بيانات جدول الكتب واستعراض عناوينها في قائمة برمجياً. يكون الملف.aspx في هذا المثال:

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

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>>Data Reader</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
Style="z-index: 100;
        left: 80px; position: absolute; top: 350px" Text="Data Reader
2" Width="192px" />
        <asp:ListBox ID="ListBox1" runat="server" Height="184px" Style="z-
index: 101; left: 80px;
        position: absolute; top: 80px" Width="193px"></asp:ListBox>
```



```

        <asp:Button ID="Button3" runat="server" OnClick="Button3_Click"
Style="z-index: 102;
        left: 80px; position: absolute; top: 312px" Text="Data Reader
1" Width="192px" />
        <asp:Button ID="Button4" runat="server" OnClick="Button4_Click"
Style="z-index: 103;
        left: 80px; position: absolute; top: 472px" Text="Clear"
Width="192px" />
        <asp:Label ID="Label1" runat="server" Style="z-index: 104; left:
80px; position: absolute;
        top: 48px" Text="Books Titles:" Width="184px"></asp:Label>
        <asp:Button ID="Button2" runat="server" Style="z-index: 102;
        left: 72px; position: relative; top: 416px"
Text="ExecuteNonQuery" Width="192px" OnClick="Button2_Click" />
        <asp:Button ID="Button5" runat="server" Style="z-index: 102;
        left: -126px; position: relative; top: 376px"
Text="ExecuteScalar" Width="192px" OnClick="Button5_Click" />
        <asp:Label ID="Label2" runat="server" Style="left: -318px;
position: relative; top: 256px"
        Width="184px"></asp:Label>
    </form>
</body>
</html>

```

نقوم في الزر الأول (Data Reader 1) بإنشاء اتصال مع قاعدة البيانات وتعريف أمر Select، ومن ثم تنفيذ هذا الأمر باستخدام الطريقة ExecuteQuery وتضمين البيانات الناتجة في SqlDataReader. نقوم بعد ذلك بتنفيذ حلقة على هذا القارئ لقراءة عنوان كل كتاب وإضافته إلى القائمة.

```

protected void Button3_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "SELECT * FROM Books";

    MySelectCommand.CommandText = sql;

    SqlDataReader DReader ;

    sqlCON.Open();
    // Execute the SqlCommand and get the SqlDataReader.
    DReader = MySelectCommand.ExecuteReader();
    this.ListBox1.Items.Clear();
    String s;
    int i;
    while (DReader.Read())
    {
        i = DReader.GetOrdinal("Title");
        s = DReader.GetString(i);
    }
}

```

```

        this.ListBox1.Items.Add(s);
    }
    DReader.Close();
    sqlCON.Close();
}

```

The screenshot shows a web application window. At the top, there is a label "Books Titles:" followed by a text box containing the following list of book titles: "Beginning ASP 3.0", "Windows for Dummies", "Word for Dummies", and "Excel for Dummies". Below the text box, there are five buttons arranged vertically: "Data Reader 1", "Data Reader 2", "ExecuteScalar", "ExecuteNonQuery", and "Clear".

يُمكن ربط القائمة مباشرة مع القارئ وذلك بإسناد اسم القارئ إلى الخاصية DataSource للقائمة، وإسناد اسم الحقل المعني إلى الخاصية DataTextField، ومن ثم تنفيذ الطريقة DataBind() على القائمة. كما تُبين الإجراءية التالية للزر Data Reader 2:

```

protected void Button1_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;
}

```

```
String sql ;
sql = "SELECT * FROM Books";

MySelectCommand.CommandText = sql;

SqlDataReader DReader ;

sqlCON.Open();
// Execute the SqlCommand and get the SqlDataReader.
DReader = MySelectCommand.ExecuteReader();
this.ListBox1.DataSource = DReader;
this.ListBox1.DataTextField = "Title";
this.ListBox1.DataBind();

DReader.Close();
sqlCON.Close();

}
```

تقوم إجرائية الزر ExecuteScalar بإظهار عدد الكتب الموجودة في الجدول:

```
protected void Button5_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "SELECT count(*) FROM Books";

    MySelectCommand.CommandText = sql;

    sqlCON.Open();

    int nL = int.Parse(MySelectCommand.ExecuteScalar().ToString());

    Label2.Text = nL.ToString() + " Books ";
    sqlCON.Close();
}
```

**Books Titles:**

Beginning ASP 3.0

Windows for Dummies

Word for Dummies

Excel for Dummies

**4 Books**

Data Reader 1

Data Reader 2

ExecuteScalar

ExecuteNonQuery

Clear

كما تقوم إجرائية الزر ExecuteNonQuery بتنفيذ استعلام تحديث على جدول الكتب وإظهار عدد التسجيلات التي تمّ تحديثها:

```
protected void Button2_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand MySelectCommand = new SqlCommand();
    MySelectCommand.Connection = sqlCON;

    String sql ;
    sql = "Update Books set CategoryId=1";

    MySelectCommand.CommandText = sql;

    sqlCON.Open();

    int nL= int.Parse( MySelectCommand.ExecuteNonQuery().ToString());
    Label2.Text = nL.ToString() + " Records has been updated ";
    sqlCON.Close();
}
```

}

**Books Titles:**

Beginning ASP 3.0  
Windows for Dummies  
Word for Dummies  
Excel for Dummies

**4 Records has been updated**

Data Reader 1

Data Reader 2

ExecuteScalar

ExecuteNonQuery

Clear

## عنصر التحكم Repeater

### الأهداف التعليمية

- استخدام عنصر التحكم Repeater.

## استخدام عنصر التحكم Repeater

يُستخدم عنصر التحكم Repeater لعرض مجموعة من البيانات بشكل تكراري. يتم تحديد مظهر الإظهار بتعريف مجموعة من القوالب.

نقوم في المثال التالي بعرض كتب فئة معينة يُحددها المستخدم من قائمة الفئات باستخدام عنصر التحكم Repeater.

Search By Category: **Computer** ▼

### Search Results:

---



#### *C How to Program*

ISBN: 0764544020

Published: February 2005

*This book is for beginners who have no previous experience*



#### *Artificial Intelligence A Modern Approach*

ISBN: 0764543636

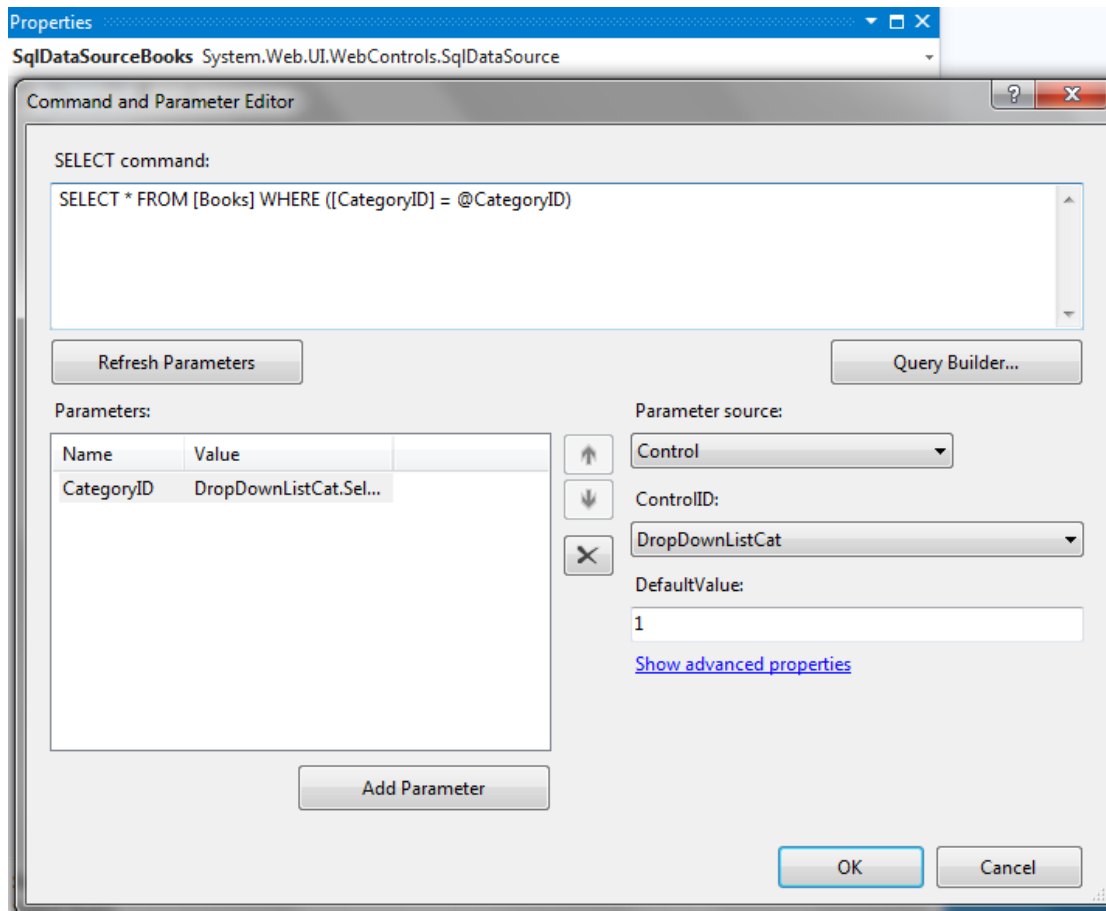
Published: July 2006

*This is a great book Most widely used in U.S. universities*

---

*For more information visit <http://www.svuonline.org>*

ترتبط قائمة الفئات DropDownListCat في المثال السابق بمصدر بيانات مرتبط مع جدول الفئات. أما العنصر Repeater فيرتبط مع مصدر بيانات مرتبط مع جدول الكتب إنما مع شرط Where وبحيث أن معرف فئة الكتاب تساوي إلى فئة الكتاب المحددة من قائمة فئات الكتب:



يتم تحديد قوالب العنصر Repeater في الملف .aspx. وأهمها:

- قالب ترويسة العنصر HeaderTemplate
- قالب تذييل العنصر FooterTemplate
- قالب العنصر ItemTemplate
- قالب الفاصل بين العناصر SeparatorTemplate

لاحظ أننا نربط صورة غلاف الكتاب بصفحة أخرى BooksForm.aspx تحوي على عنصر FormView مرتبط مع جدول الكتب. نُمرر لهذه الصفحة معامل ISBN في العنوان كي تقوم بإظهار الكتاب المعني.

```
<div>
  <strong><span style="font-size: 14pt; color: #0000cc">
    Search By Category:</span></strong>
  <asp:DropDownList ID="DropDownListCat" runat="server"
DataSourceID="SqlDataSourceCat"
  DataTextField="Category" DataValueField="CategoryID"
Style="position: relative" AutoPostBack="True" Font-Bold="True" Font-
Names="Tw Cen MT Condensed" Font-Size="Larger" ForeColor="DarkBlue">
  </asp:DropDownList><asp:SqlDataSource ID="SqlDataSourceCat"
runat="server" ConnectionString="<%$
ConnectionStrings:LibraryConnectionString %>"
  SelectCommand="SELECT * FROM [Categories]"></asp:SqlDataSource>
<br />
</div>
```



```

<div>

<asp:Repeater ID="Repeater1" runat="server"
DataSourceID="SqlDataSourceBooks">

<HeaderTemplate>
<div class="rHead">
Search Results: <br />

</div>
</HeaderTemplate>

<ItemTemplate>
<div class="rItem">
<a href="BooksForm.aspx?ISBN=<%# DataBinder.Eval(Container.DataItem,
"ISBN")%>" >
    <input type="image"
        src="images/<%# DataBinder.Eval(Container.DataItem, "ImageUrl")%>"
    />
</a>

<b><%# DataBinder.Eval(Container.DataItem, "Title")%></b><br/>
ISBN: <%# DataBinder.Eval(Container.DataItem, "ISBN")%> <br/>
Published: <%# DataBinder.Eval(Container.DataItem, "PublicationDate")%><br
/>
<%# DataBinder.Eval(Container.DataItem, "Precis")%>
</div><br/>
</ItemTemplate>

<SeparatorTemplate>

</SeparatorTemplate>

<FooterTemplate>

<div class="rFoot">
For more information visit
    <a href="http://www.svuonline.org">http://www.svuonline.org</a>
</div>
</FooterTemplate>

</asp:Repeater>

</div>
<asp:SqlDataSource ID="SqlDataSourceBooks" runat="server"
ConnectionString="<%$ ConnectionStrings:LibraryConnectionString %>"
SelectCommand="SELECT * FROM [Books] WHERE ([CategoryID] = @CategoryID)">
    <SelectParameters>
        <asp:ControlParameter ControlID="DropDownListCat" DefaultValue="1"
Name="CategoryID"
        PropertyName="SelectedValue" Type="Int32" />
    </SelectParameters>
</asp:SqlDataSource>
    &nbsp;
</form>
</body>
</html>

```

## مجموعات البيانات Datasets

### الأهداف التعليمية

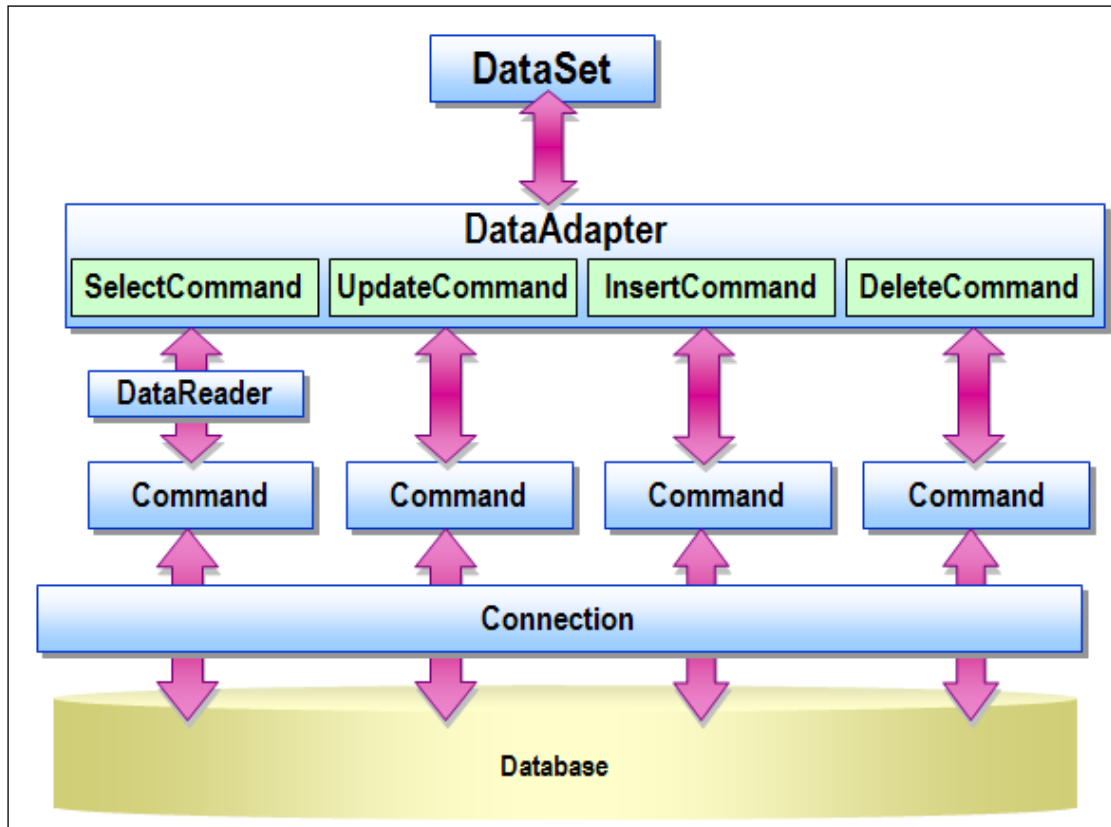
- مجموعات البيانات Datasets.

## الصف SqlDataAdapter

- يقوم الغرض من هذا الصف بوصل غرض من الصف sqlCommand أو أكثر بغرض DataSet.
- يملك أربعة خصائص أساسية لتحديد الأوامر الأساسية: SelectCommand, InsertCommand, UpdateCommand, DeleteCommand.
- من أهم الطرق:
  - الطريقة Fill: تقوم بتنفيذ الأمر SelectCommand لتأهيل غرض DataSet.
  - الطريقة Update: تقوم باستدعاء الأوامر InsertCommand, DeleteCommand, UpdateCommand لكل تسجيلية تم إدراجها أو تعديلها أو حذفها ضمن مصدر البيانات DataSet الموافق.

## الصف DataSet

- يُقدم هذا الصف مجموعة من الطرق للتعامل مع قواعد البيانات العلاقتية في بيئة غير متصلة.
  - نقوم أولاً بملء غرض الصف من قاعدة بيانات.
  - نتعامل مع هذا الغرض في بيئة غير متصلة.
  - نُعيد الاتصال مع قاعدة البيانات لعكس جميع التغييرات التي قُمنّا بها على البيانات.
- يُبين الشكل التالي مخطط التعامل مع قاعدة بيانات:



مثال:

نقوم في هذا المثال التعليمي بربط عنصر DataAdapter مع أربعة أوامر Select, Insert, Update, Delete. ومن ثم استخدام الطريقة Fill() لملء مجموعة بيانات DataSet. نقوم بعد ذلك بتعديل بيانات هذه المجموعة (إضافة، تعديل، حذف) من خلال الكود. ومن ثم تنفيذ هذه التعديلات على قاعدة البيانات باستخدام الطريقة Update.

كما نربط عنصر تحكم من النوع GridView بمجموعة البيانات عن طريق الخاصية DataSource والطريقة .DataBind()

مثال:

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

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="Button5" runat="server" Height="48px"
OnClick="Button5_Click" Style="z-index: 100;
left: 168px; position: absolute; top: 240px" Text="DataAdapter"
Width="177px" />
            <asp:GridView ID="GridView1" runat="server" Style="z-index: 102;
left: 32px; position: absolute;
top: 96px" Width="504px">
                </asp:GridView>
        </div>
    </form>
</body>
</html>
```

```
protected void Button5_Click(object sender, EventArgs e)
{
    String ConStr ;
    ConStr = "Data
Source=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\Library.mdf;Integrated
Security=True";

    SqlConnection sqlCON = new SqlConnection();
    sqlCON.ConnectionString = ConStr;

    SqlCommand SelectCom = new SqlCommand();
    SqlCommand UpCom = new SqlCommand();
    SqlCommand InsertCom = new SqlCommand();
    SqlCommand DelCom = new SqlCommand();

    SelectCom.Connection = sqlCON;
    UpCom.Connection = sqlCON;
    InsertCom.Connection = sqlCON;
    DelCom.Connection = sqlCON;
```

```

SelectCom.CommandText = "Select * from Logins";

UpCom.CommandText = "UPDATE [Logins] SET [Login] = @Login, [Password] =
@Password, [BirthDate] = @BirthDate, [Amount] = @Amount, [EMail] = @EMail
WHERE [UserID] = @UserID";
UpCom.Parameters.Add("@Login", SqlDbType.VarChar, 5, "Login");
UpCom.Parameters.Add("@Password", SqlDbType.VarChar, 100, "Password");
UpCom.Parameters.Add("@BirthDate", SqlDbType.DateTime, 100, "BirthDate");
UpCom.Parameters.Add("@Amount", SqlDbType.Int, 100, "Amount");
UpCom.Parameters.Add("@EMail", SqlDbType.VarChar, 100, "EMail");
UpCom.Parameters.Add("@UserId", SqlDbType.Int, 100, "UserId");

InsertCom.CommandText = "INSERT INTO [Logins] ([Login], [Password],
[BirthDate], [Amount], [EMail]) VALUES (@Login, @Password, @BirthDate,
@Amount, @EMail)";
InsertCom.Parameters.Add("@Login", SqlDbType.VarChar, 5, "Login");
InsertCom.Parameters.Add("@Password", SqlDbType.VarChar, 100, "Password");
InsertCom.Parameters.Add("@BirthDate", SqlDbType.DateTime, 100,
"BirthDate");
InsertCom.Parameters.Add("@Amount", SqlDbType.Int, 100, "Amount");
InsertCom.Parameters.Add("@EMail", SqlDbType.VarChar, 100, "EMail");

DelCom.CommandText = "Delete from [Logins] WHERE [UserID] = @UserID";
DelCom.Parameters.Add("@UserId", SqlDbType.Int, 100, "UserId");

SqlDataAdapter DBAdapter= new SqlDataAdapter() ;
DataSet ResultsDataSet = new DataSet();

DBAdapter.SelectCommand = SelectCom;
DBAdapter.UpdateCommand = UpCom;
DBAdapter.InsertCommand = InsertCom;
DBAdapter.DeleteCommand = DelCom;

DBAdapter.Fill(ResultsDataSet);

for (int i = 0 ; i<= ResultsDataSet.Tables[0].Rows.Count - 1;i++)
    ResultsDataSet.Tables[0].Rows[i].ItemArray[5] = i * 1000;

ResultsDataSet.Tables[0].Rows.Add(2, "tata", "789");
ResultsDataSet.Tables[0].Rows[0].Delete();

DBAdapter.Update(ResultsDataSet);

sqlCON.Close();

this.GridView1.DataSource = ResultsDataSet;
this.GridView1.DataBind();
}

```

UserID	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	2000.0000
2	tata	789			

DataAdapter

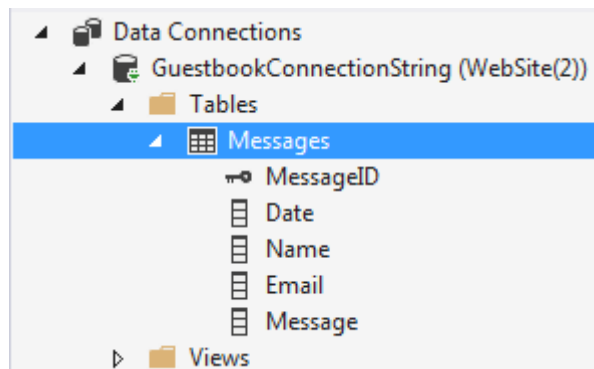
## مثال تطبيقي

### الأهداف التعليمية

- مثال تطبيقي.

ليكن المطلوب إنشاء صفحة في موقع يستطيع من خلالها زوار الموقع تسجيل ملاحظاتهم حول هذا الموقع. لتكن قاعدة البيانات Guestbook.mdf. تحوي هذه القاعدة جدول وحيد لتخزين تعليقات زوار الموقع Messages.

يُبين الشكل التالي حقول الجدول Messages:



لتكن الصفحة Guestbook.aspx التالية:

**Please leave a message in our guestbook:**

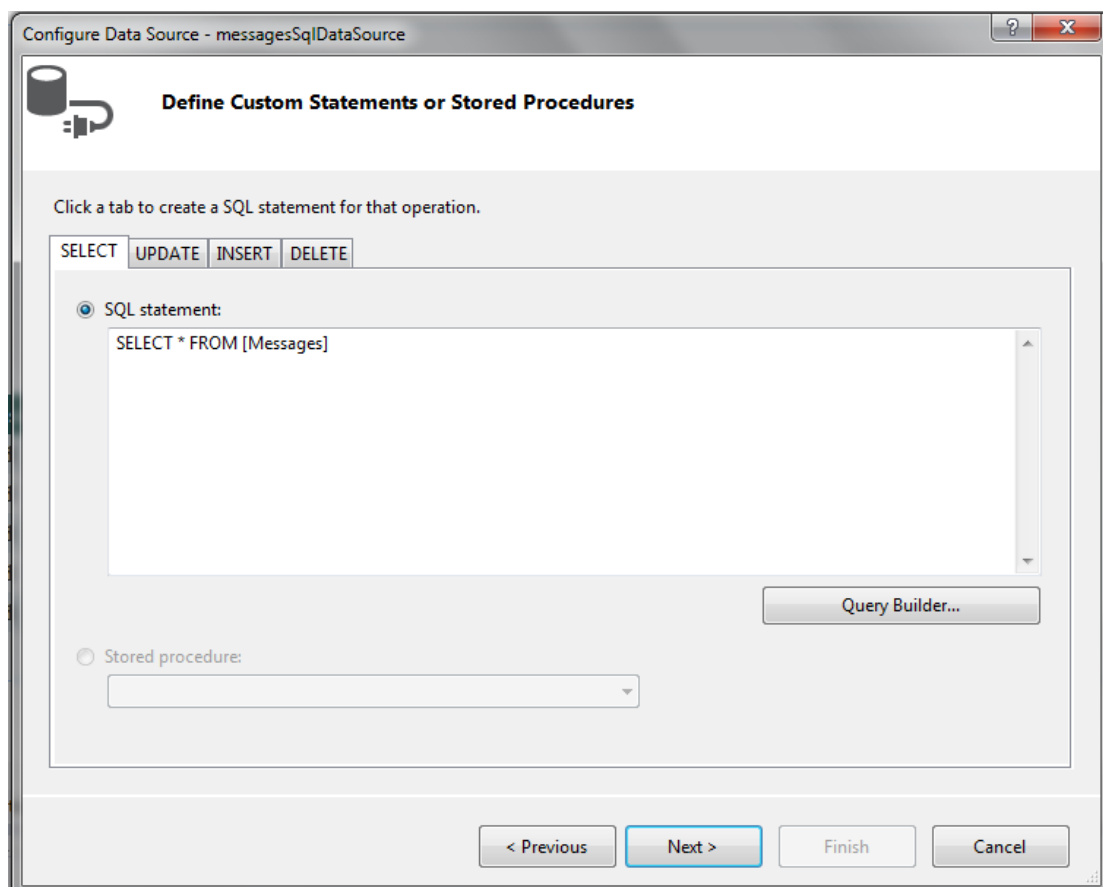
Your name:	<input type="text"/>
Your e-mail address:	<input type="text"/>
Tell the world:	<input type="text"/>
<input type="button" value="Submit"/> <input type="button" value="Clear"/>	

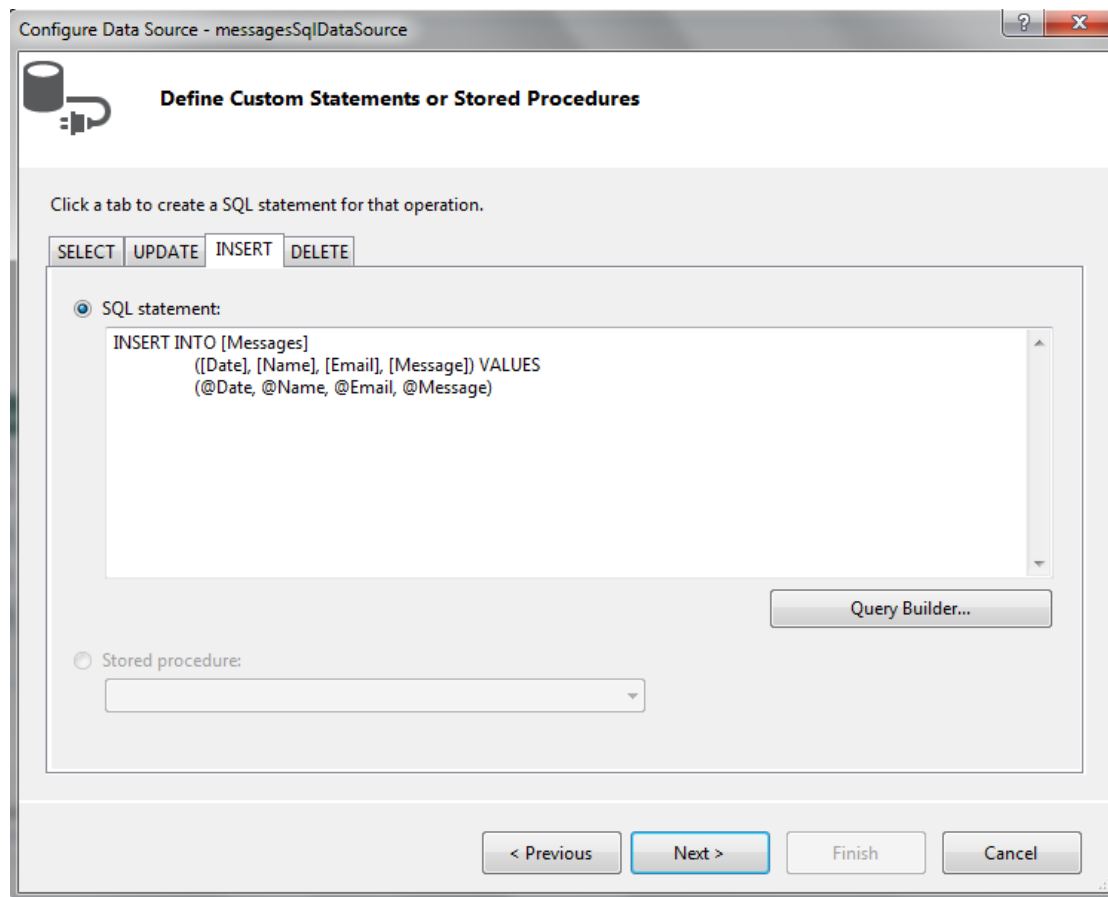
Date	Name	Email	Message
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc
abc	abc	abc	abc

SqlDataSource - messagesSqlDataSource

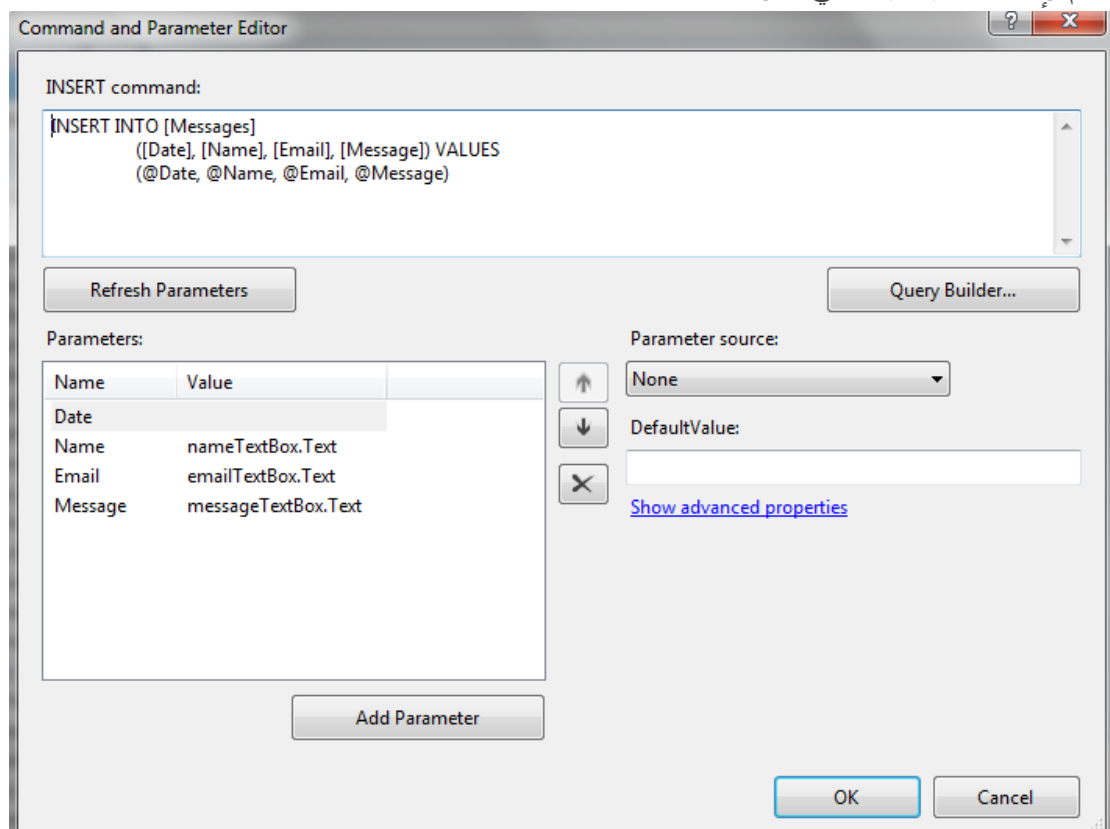


يرتبط العنصر MessagesSqlDataSource من النوع sqlDataSource مع جدول الرسائل وذلك باستخدام المعالج Configure Data Source (بالنقر بالزر الأيمن على العنصر).





ترتبط معاملات العبارة السابقة بعناصر التحكم وفق ما يلي مع ملاحظة أن المعامل @Date لا يرتبط بعنصر تحكم وإنما سنعطيه قيمته في الكود:



```

<%-- Guestbook.aspx --%>
<%-- Guestbook Web application with a form for users to submit --%>
<%-- guestbook entires and a GridView to view existing entries. --%>
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Guestbook.aspx.cs" Inherits="Guestbook" %>

<!DOCTYPE html >
<html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Guestbook</title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <h2><span style="color: navy">
                    Please leave a message in our guestbook:</span></h2>

                <table>
                    <tr>
                        <td style="width: 130px; height: 21px;"
                            valign="top">
                            Your name:<br />
                        </td>
                        <td style="width: 300px; height: 21px;"
                            valign="top">
                            <asp:TextBox ID="nameTextBox" runat="server"
                                Width="300px"></asp:TextBox>
                        </td>
                    </tr>
                    <tr>
                        <td style="width: 130px;" valign="top">
                            Your e-mail address:<br />
                        </td>
                        <td style="width: 300px;" valign="top">
                            <asp:TextBox ID="emailTextBox" runat="server"
                                Width="300px"></asp:TextBox></td>
                    </tr>
                    <tr>
                        <td style="width: 130px;" valign="top">
                            Tell the world:<br />
                        </td>
                        <td style="width: 300px;" valign="top">
                            <asp:TextBox ID="messageTextBox" runat="server"
                                Height="100px" Rows="8" Width="300px">
                            </asp:TextBox>
                        </td>
                    </tr>
                    <tr>
                        <td style="width: 130px;" valign="top">
                        </td>
                        <td style="width: 300px;" valign="top">
                            <asp:Button ID="submitButton" runat="server"
                                Text="Submit" OnClick="submitButton_Click" />
                            <asp:Button ID="clearButton" runat="server"
                                Text="Clear" OnClick="clearButton_Click" />
                        </td>
                    </tr>
                </table>
            </div>
        </form>
    </body>
</html>

```

```

</table>
<br />

<asp:GridView ID="messagesGridView" runat="server"
    AutoGenerateColumns="False" CellPadding="4"
    ForeColor="#333333" GridLines="None"
    DataSourceID="messagesSQLDataSource" Width="600px"
    DataKeyNames="MessageID">
    <FooterStyle BackColor="#1C5E55" Font-Bold="True"
        ForeColor="White" />
    <RowStyle BackColor="#E3EAEB" />
    <PagerStyle BackColor="#666666" ForeColor="White"
        HorizontalAlign="Center" />
    <SelectedRowStyle BackColor="#C5BBAF" Font-Bold="True"
        ForeColor="#333333" />
    <HeaderStyle BackColor="#1C5E55" Font-Bold="True"
        ForeColor="White" />
    <EditRowStyle BackColor="#7C6F57" />
    <AlternatingRowStyle BackColor="White" />
    <Columns>
        <asp:BoundField DataField="Date" HeaderText="Date"
            SortExpression="Date" />
        <asp:BoundField DataField="Name" HeaderText="Name"
            SortExpression="Name" />
        <asp:BoundField DataField="Email"
            HeaderText="Email"
            SortExpression="Email" />
        <asp:BoundField DataField="Message"
            HeaderText="Message"
            SortExpression="Message" />
    </Columns>
</asp:GridView>

<asp:SqlDataSource ID="messagesSqlDataSource"
    runat="server"
    ConnectionString=
        "<%= $ConnectionString:GuestbookConnectionString %>"
    SelectCommand="SELECT * FROM [Messages]"
    DeleteCommand="DELETE FROM [Messages] WHERE
        [MessageID] = @original_MessageID"
    InsertCommand="INSERT INTO [Messages]
        ([Date], [Name], [Email], [Message]) VALUES
        (@Date, @Name, @Email, @Message)"
    UpdateCommand="UPDATE [Messages] SET [Date] = @Date,
        [Name] = @Name, [Email] = @Email, [Message] = @Message
        WHERE [MessageID] = @original_MessageID">
    <DeleteParameters>
        <asp:Parameter Name="original_MessageID"
            Type="Int32" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="Date" Type="String" />
        <asp:Parameter Name="Name" Type="String" />
        <asp:Parameter Name="Email" Type="String" />
        <asp:Parameter Name="Message" Type="String" />
        <asp:Parameter Name="original_MessageID"
            Type="Int32" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="Date" Type="String" />
        <asp:ControlParameter ControlID="nameTextBox"

```

```

        Name="Name" PropertyName="Text" Type="String" />
        <asp:ControlParameter ControlID="emailTextBox"
        Name="Email" PropertyName="Text" Type="String" />
        <asp:ControlParameter ControlID="messageTextBox"
        Name="Message" PropertyName="Text" Type="String" />
    </InsertParameters>
</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

أما ملف الكود الخلفي:

```

// Guestbook.aspx.cs
// Code-behind file that defines event handlers for the guestbook.
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Guestbook : System.Web.UI.Page
{
    // Submit Button adds a new guestbook entry to the database,
    // clears the form and displays the updated list of guestbook entries
    protected void submitButton_Click( object sender, EventArgs e )
    {
        // create a date parameter to store the current date

        System.Web.UI.WebControls.Parameter date =
            new System.Web.UI.WebControls.Parameter(
                "Date", TypeCode.String, DateTime.Now.ToShortDateString() );

        // set the @Date parameter to the date parameter

        messagesSqlDataSource.InsertParameters.RemoveAt( 0 );
        messagesSqlDataSource.InsertParameters.Add( date );

        // execute an INSERT SQL statement to add a new row to the
        // Messages table in the Guestbook database that contains the
        // current date and the user's name, e-mail address and message
        messagesSqlDataSource.Insert();

        // clear the TextBoxes
        nameTextBox.Text = "";
        emailTextBox.Text = "";
        messageTextBox.Text = "";

        // update the GridView with the new database table contents
        messagesGridView.DataBind();
    } // end method submitButton_Click

    // Clear Button clears the Web Form's TextBoxes

```

```
protected void clearButton_Click( object sender, EventArgs e )
{
    nameTextBox.Text = "";
    emailTextBox.Text = "";
    messageTextBox.Text = "";
} // end method clearButton_Click
} // end class Guestbook
```

ويكون التنفيذ:

**Please leave a message in our guestbook:**

Your name:

Your e-mail address:

Tell the world:

Date	Name	Email	Message
5/17/2007	Bassel ALKHATIB	t_balkhatib@svuonline.org	Great site!
5/17/2007	Sue Black	sblack@svuonline.org	I love the site! Keep up the good work!
5/18/2007	Liz White	lwhite@svuonline.org	Very useful information. Will visit again soon.