القسم الثاني– التقنيات من جهة المخدم Server Side

الفصل العاشر – التعامل مع قواعد البيانات برمجياً
Accessing Database with Code

التعامل مع قواعد البيانات

يُمكن في NET. التعامل مع قواعد البيانات باستخدام مجموعة من عناصر التحكم الجاهزة. كما يُمكن التعامل مع قواعد البيانات وتنفيذ العمليات الأساسية عليها برمجياً.

سنقوم خلال هذه الفصل بالتعرف على أهم التعليمات البرمجية للتعامل مع البيانات. وسنقوم في الفصل القادم ومن خلال تطبيق عملي من استعراض أهم عناصر التحكم للوصول للبيانات.

التعامل مع قواعد البيانات برمجيا

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

نعتمد 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. ومن أهم طرق هذا الغرض الطريقة ()Open التي تقوم بفتح اتصال مع قاعدة البيانات المحددة بسلسلة الاتصال، والطريقة ()Close التي تقوم بإغلاق الاتصال المفتوح.

```
String ConStr ;
ConStr="Data Source=(LocalDB)\MSSQLLocalDB; AttachDbFilename =
|DataDirectory|\Library.mdf;Integrated  Security=True";
SqlConnection sqlCON = new SqlConnection();
sqlCON.ConnectionString = ConStr;
sqlCON.Open();
….
….
```

```
….
…
sqlCON.Close();
```

وبما أنه سيتم استخدام conStr في أكثر من مكان لذلك يفضل تخزينه في الملف web.config وذلك بإضافة الأسطر التالية:

```
<connectionStrings>
        <add name="LibraryConnectionString1" connectionString="Data Source =
(LocalDB)\MSSQLLocalDB ;AttachDbFilename=|DataDirectory|\Library.mdf;Integrated
Security=True"
             providerName="System.Data.SqlClient" />
</connectionStrings>
```

ويعدل الكود المكتوب أعلاه إلى:

```
String ConStr ;
ConStr=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
SqlConnection sqlCON = new SqlConnection();
sqlCON.ConnectionString = ConStr;
sqlCON.Open();
….
….
….
…
sqlCON.Close();
```

الصف SqlCommand

يُستخدم غرض من هذا الصف لوصل غرض SqlConnection مع غرض SqlDataReader أو مع غرض SqlDataAdapter. يسمح الغرض SqlCommand بتنفيذ عبارات SQL أو إجرائيات 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 لاتُعيد مجموعة تسجيلات. من أهم طرق هذا الغرض:

- الطريقة ExecuteReader: تقوم بتنفيذ الأمر المعرف في الخاصية CommandText والذي يُعيد مجموعة من التسجيلات.
- الطريقة ExecuteNonQuery: تقوم هذه الطريقة بتنفيذ الأمر المُعرّف في الخاصية CommandText (استعلام لايُعيد صفوف مثل تعليمة Insert أو Update أو Delete) وتُعيد عدد التسجيلات التي تأثرت بالاستعلام.
- الطريقة ExecuteScalar: تقوم بتنفيذ الأمر المعرف في الخاصية CommandText والذي يُعيد قيمة وحيدة.

الصف SqlDataReader

يوفر غرض من هذا الصف طريقة سريعة وفعالة للوصول إلى مجموعة من التسجيلات بشكل أمامي وللقراءة فقط. من أكثر طرق هذا الغرض استخداماً:

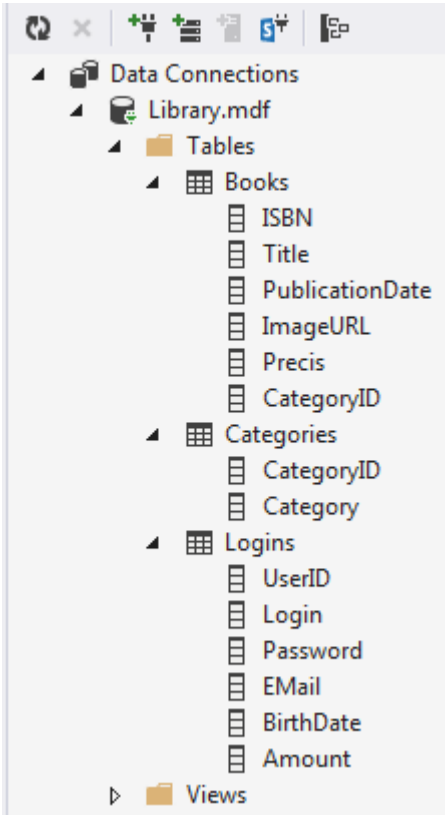
- الطريقة Read: تقوم بتحريك مؤشر التسجيلة إلى التسجيلة التالية. وتُعيد false عند الوصول لنهاية التسجيلات.
- الطريقة GetOrdinal("ColName"): تُعيد فهرس العمود الذي اسمه "ColName".
- الطريقة GetString (index): تُعيد قيمة العمود ذو الفهرس index كسلسلة نصية.

مثال تعليمي 1

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

يُبين الشكل التالي حقول هذه الجداول:



ولنفرض كمثال تعليمي أننا نُريد قراءة بيانات جدول الكتب واستعراض عناوينها في قائمة برمجياً.

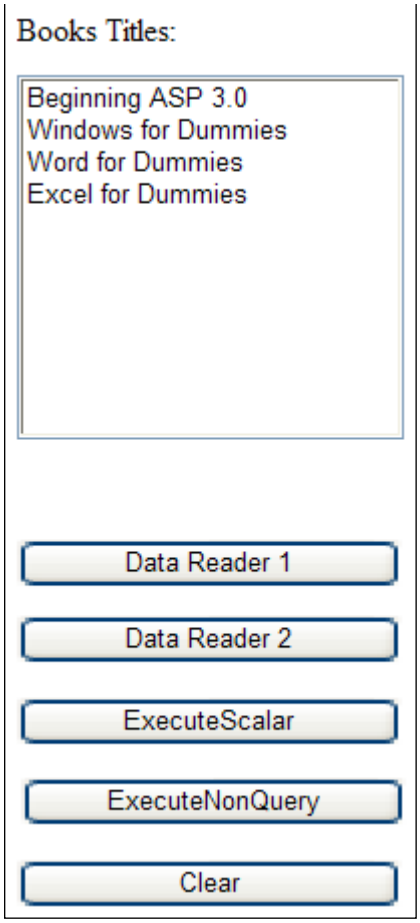
نقوم في الزر الأول (Data Reader 1) في الصفحة التالية بإنشاء اتصال مع قاعدة البيانات وتعريف أمر Select، ومن ثم تنفيذ هذا الأمر باستخدام الطريقة ExecuteQuery

وتضمين البيانات الناتجة في DataReader.

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=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();

            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();

        }
```

وتكون نتيجة تنفيذ هذه الإجرائية:

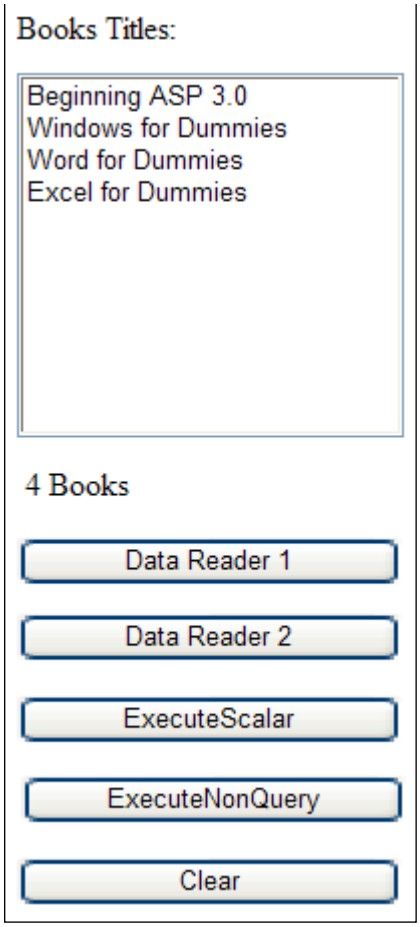


يُمكن ربط القائمة مباشرة مع القارئ وذلك بإسناد اسم القارئ إلى الخاصية DataSource للقائمة، وإسناد اسم الحقل المعني إلى الخاصية DataTextField، ومن ثم تنفيذ الطريقة DataBind() على القائمة. كما تُبين الإجرائية التالية للزر Data Reader 2:

```
    protected void Button1_Click(object sender, EventArgs e)
        {
            String ConStr ;

ConStr=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            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=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            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();
    }
```

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=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            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();

        }
```

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>
```

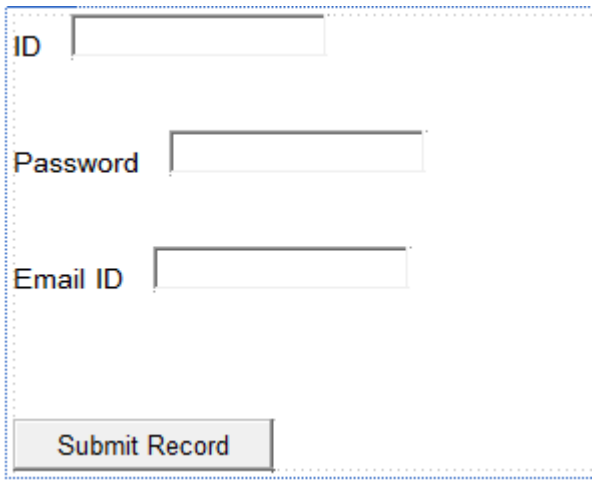
مثال تعليمي 2

سنتعلم في هذا المثال كيفية طلب إجرائية مخزنة Stored procedure .

ننشئ الإجرائية المخزنة التالية:

```
CREATE PROCEDURE dbo.submitrecord
            @UserId varchar(50),
            @Password varchar(50),
            @EmailID varchar(200)
AS
      Begin
            insert into logins (UserId,Password,EMail)
                        values (@UserId,@Password,@EmailID)
      End
```

ثم نصمم واجهة لإدخال القيم



ويكون الكود المقابل:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

    <head runat="server">
```

```
        <title>Store Procedure</title>
    </head>

    <body>
        <form id="form1" runat="server">
            <div>
                <asp:Label ID="Label1" runat="server" Text="ID"></asp:Label>
                  
                <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />
                <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label>
                  
                <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br /><br />
                <asp:Label ID="Label4" runat="server" Text="Email ID"></asp:Label>
                  
                <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox><br /><br /><br />
                <asp:Button ID="Button1" runat="server" Text="Submit Record"
                     OnClick="Button1_Click" /> </div>
        </form>
</body>
</html>
```

```
    protected void Button1_Click(object sender, EventArgs e)
    {
        string ConStr;

ConStr=ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
        SqlConnection sqlCON = new SqlConnection();
        sqlCON.ConnectionString = ConStr;
        cmd = new SqlCommand("submitrecord", sqlCON);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@ID", SqlDbType.VarChar).Value = TextBox1.Text;
        cmd.Parameters.Add("@Password", SqlDbType.VarChar).Value = TextBox2.Text;
        cmd.Parameters.Add("@EmailID", SqlDbType.VarChar).Value = TextBox4.Text;

        sqlCON.Open();
        cmd.ExecuteNonQuery();
        sqlCON.Close();

    }
```

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.

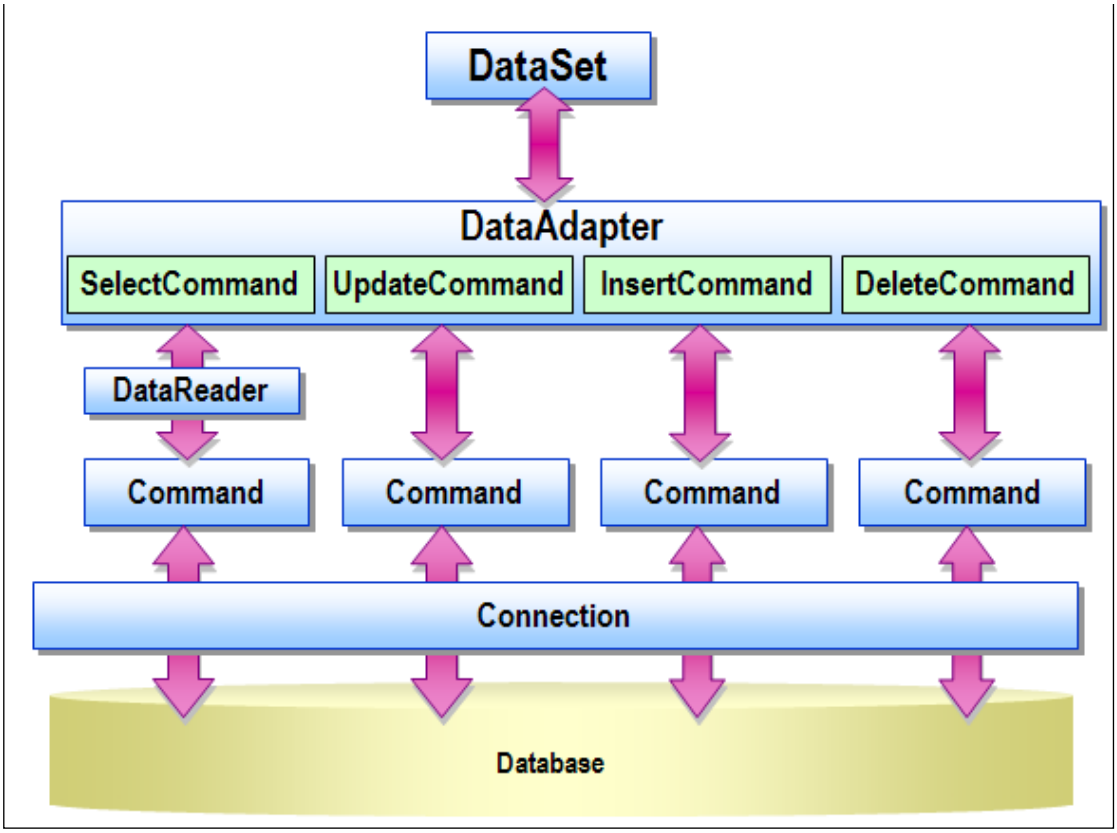
ومن أهم الطرق:

- الطريقة Fill: تقوم بتنفيذ الأمر SelectCommand لتأهيل غرض DataSet.
- الطريقة Update: تقوم باستدعاء الأوامر InsertCommand, DeleteCommand, UpdateCommand لكل تسجيلة تمّ إدراجها أو تعديلها أو حذفها ضمن مصدر البيانات DataSet الموافق.

الصف DataSet

يُقدم هذا الصف مجموعة من الطرق للتعامل مع قواعد البيانات العلاقاتية في بيئة غير متصلة. حيث نقوم أولاً بملء غرض الصف من قاعدة بيانات، ثم نتعامل مع هذا الغرض في بيئة غير متصلة، ثم نُعيد الاتصال مع قاعدة البيانات لعكس جميع التغييرات التي قُمنا بها على البيانات.

يُبين الشكل التالي مخطط التعامل مع قاعدة بيانات:

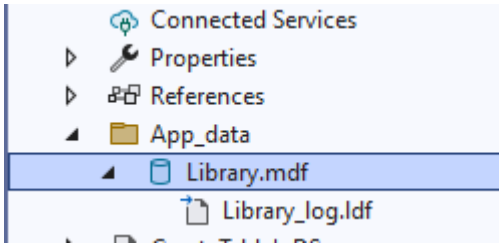


مثال: استخدام DataAdapter

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.

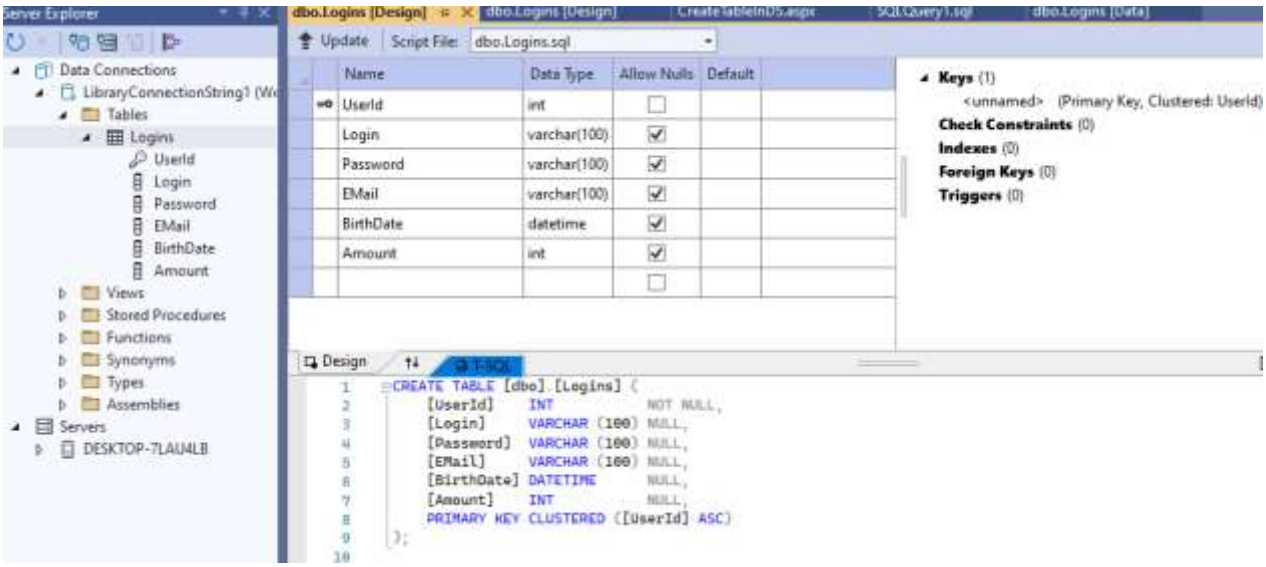
كما نربط عنصر تحكم من النوع GridView بمجموعة البيانات عن طريق الخاصية DataSource والطريقة DataBind().

أولاً– نقوم بإنشاء قاعدة بيانات Library

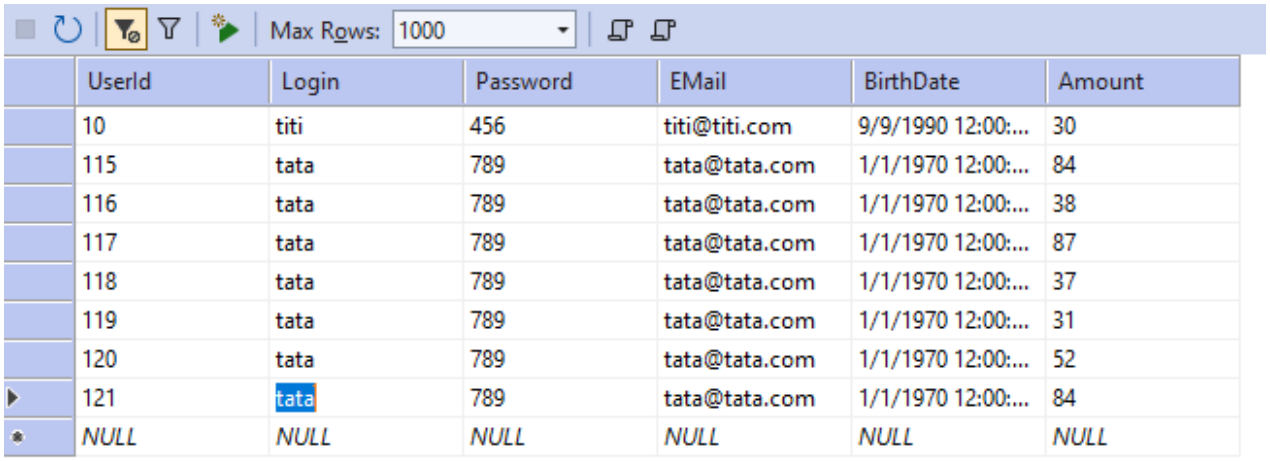


ثانياً- نقوم بإنشاء جدول Logins في قاعدة البيانات Library:

```
CREATE TABLE [dbo].[Logins] (
    [UserId]    INT           NOT NULL,
    [Login]     VARCHAR (100) NULL,
    [Password]  VARCHAR (100) NULL,
    [EMail]     VARCHAR (100) NULL,
    [BirthDate] DATETIME      NULL,
    [Amount]    INT           NULL,
    PRIMARY KEY CLUSTERED ([UserId] ASC)
);
```



نضيف بعض البيانات لهذا الجدول مثلاً:



Max Rows: 1000

UserId	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:...	30
115	tata	789	tata@tata.com	1/1/1970 12:00:...	84
116	tata	789	tata@tata.com	1/1/1970 12:00:...	38
117	tata	789	tata@tata.com	1/1/1970 12:00:...	87
118	tata	789	tata@tata.com	1/1/1970 12:00:...	37
119	tata	789	tata@tata.com	1/1/1970 12:00:...	31
120	tata	789	tata@tata.com	1/1/1970 12:00:...	52
121	tata	789	tata@tata.com	1/1/1970 12:00:...	84
NULL	NULL	NULL	NULL	NULL	NULL

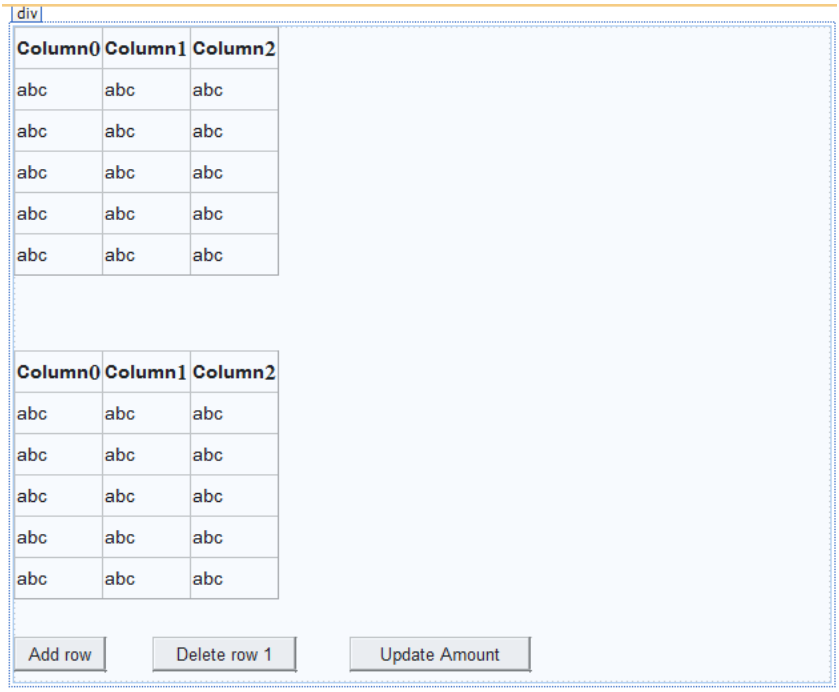
ثالثاً– نضيف معلومات الإتصال إلى ملف الإعدادات Web.config:

```
<connectionStrings>
    <add name="LibraryConnectionString1" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\
Library.mdf;Integrated Security=True"
      providerName="System.Data.SqlClient" />
</connectionStrings>
```

ثالثاً– أنشئ ملف الصفحة DataAdapter.aspx كمايلي:

```
<%@ 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></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server"> </asp:GridView>
            <br />
            <br />
            <asp:GridView ID="GridView2" runat="server"> </asp:GridView>
            <br />
            <asp:Button ID="AddRowBtn" runat="server"
OnClick="AddRowBtn_Click" Text="Add row" />
      
            <asp:Button ID="DelRowBtn" runat="server"
OnClick="DelRowBtn_Click" Text="Delete row 1" />
       
            <asp:Button ID="UpdateBtn" runat="server"
OnClick="UpdateBtn_Click" Text="Update Amount" />
        </div>
```

```
    </form>
</body>
</html>
```



معالجة الأحداث:

1-في معالج الحدث Page_Load نضع الكود التالي:

```
protected void Page_Load(object sender, EventArgs e)
        {
            string ConStr;
            ConStr =
ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            SqlConnection sqlCON = new SqlConnection();
            sqlCON.ConnectionString = ConStr;

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

            SqlCommand SelectCom = new SqlCommand();
            SelectCom.CommandText = "Select * from Logins";

            SelectCom.Connection = sqlCON;
            DBAdapter.SelectCommand = SelectCom;
            DBAdapter.Fill(ResultsDataSet,"fullTable");
            this.GridView1.DataSource = ResultsDataSet.Tables[0];
            // //Tables[0]="fullTable"
            this.GridView1.DataBind();

            SqlCommand SelectCom1 = new SqlCommand();
            SelectCom1.CommandText = "Select userid,login from Logins";
            SelectCom1.Connection = sqlCON;
```

```
            DBAdapter.SelectCommand = SelectCom1;
            DBAdapter.Fill(ResultsDataSet, "partiallyTable");
            this.GridView2.DataSource =
ResultsDataSet.Tables["partiallyTable"];
            this.GridView2.DataBind();

            ResultsDataSet.Dispose();
            DBAdapter.Dispose();
        }
```

عند تنفيذ هذه الصفحة تكون النتيجة على الشكل التالي:

UserId	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	30
115	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	84
116	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	38
117	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	87
118	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	37
119	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	31
120	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	52
121	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	84

userid	login
10	titi
115	tata
116	tata
117	tata
118	tata
119	tata
120	tata
121	tata

Add row Delete row 1 Update Amount

نلاحظ أنه تم إنشـــاء جدولين في الغرض ResultsDataSet أحدهما يتضـــمن جميع الأعمدة الموجودة في الجدول الأساسي Logins والآخر تضمن فقط عمودين.

كما نلاحظ أنه يمكن الوصول إلى الجداول بإحدى طريقتين:

- عن طريق المصفوفة ResultsDataSet.Tables[?]
- عن طريق الاسم في حال تم إعطاء اسم للجدول أثناء تنفيذ الأمر Fill.

2-في معالج الحدث للزر Delete Row 1 نضع الكود التالي:

```
        protected void DelRowBtn_Click(object sender, EventArgs e)
        {
            string ConStr;
            ConStr =
ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            SqlConnection sqlCON = new SqlConnection();
            sqlCON.ConnectionString = ConStr;

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

            SqlCommand SelectCom = new SqlCommand();
            SelectCom.CommandText = "Select * from Logins";

            SelectCom.Connection = sqlCON;
            DBAdapter.SelectCommand = SelectCom;
            DBAdapter.Fill(ResultsDataSet, "fullTable");
            this.GridView1.DataSource = ResultsDataSet.Tables["fullTable"];
            this.GridView1.DataBind();

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

            DBAdapter.DeleteCommand = DelCom;
            ResultsDataSet.Tables["fullTable"].Rows[1].Delete();
            //    ResultsDataSet.Tables["fullTable"].Rows.RemoveAt(1);

            DBAdapter.Update(ResultsDataSet, "fullTable");
            this.GridView1.DataSource = ResultsDataSet.Tables[0];
            this.GridView1.DataBind();

        }
```

يقوم هذا الكود بحذف السطر الثاني أي ذو الترتيب (1) من الـــ Collection الخاصة بالأسطر.

عند التنفيذ والضغط مثلا أربعة مرات على هذا الزر نجد النتيجة التالية:

UserId	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	30
119	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	31
120	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	52
121	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	84

userid	login
10	titi
119	tata
120	tata
121	tata

Add row Delete row 1 Update Amount

3-في معالج الحدث للزر Add Row نضع الكود التالي:

```
        protected void AddRowBtn_Click(object sender, EventArgs e)
         {
             string ConStr;
             ConStr =
ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
             SqlConnection sqlCON = new SqlConnection();
             sqlCON.ConnectionString = ConStr;

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

             SqlCommand SelectCom = new SqlCommand();
             SelectCom.CommandText = "Select * from Logins order by UserId";

             SelectCom.Connection = sqlCON;
             DBAdapter.SelectCommand = SelectCom;
             DBAdapter.Fill(ResultsDataSet, "fullTable");



             SqlCommand InsertCom = new SqlCommand();

             InsertCom.Connection = sqlCON;


             InsertCom.CommandText = "INSERT INTO [Logins] ([UserId],[Login],
[Password], [BirthDate], [Amount], [EMail]) VALUES (@UserId,@Login, @Password,
@BirthDate, @Amount, @EMail)";
```

```
            InsertCom.Parameters.Add("@UserId", SqlDbType.Int, 5, "UserId");
            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");
            DBAdapter.InsertCommand = InsertCom;

            int lastUserId =
(int)ResultsDataSet.Tables[0].Rows[ResultsDataSet.Tables[0].Rows.Count -
1]["USerId"];


            ResultsDataSet.Tables[0].Rows.Add(lastUserId + 1, "tata", "789",
"tata@tata.com", "1/1/1970", 50);

            DBAdapter.Update(ResultsDataSet, "fullTable");

            this.GridView1.DataSource = ResultsDataSet.Tables["fullTable"];
            this.GridView1.DataBind();


             }
```

عند الضغط على هذا الزر مرتين يضاف سطرين إلى نهاية الجدول بحيث يكون UserId زائداً (1) عن آخر سطر.

UserId	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	30
119	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	31
120	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	52
121	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	84
122	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	50
123	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	50

userid	login
10	titi
119	tata
120	tata
121	tata
122	tata
123	tata

Add row Delete row 1 Update Amount

4-أخيراً في معالج الحدث للزر Update Amount نضع الكود التالي:

```
        protected void UpdateBtn_Click(object sender, EventArgs e)
        {
            string ConStr;
            ConStr =
ConfigurationManager.ConnectionStrings["LibraryConnectionString1"].ToString();
            SqlConnection sqlCON = new SqlConnection();
            sqlCON.ConnectionString = ConStr;

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

            SqlCommand SelectCom = new SqlCommand();
            SelectCom.CommandText = "Select * from Logins";

            SelectCom.Connection = sqlCON;
            DBAdapter.SelectCommand = SelectCom;
            DBAdapter.Fill(ResultsDataSet, "fullTable");
            this.GridView1.DataSource = ResultsDataSet.Tables["fullTable"];
            this.GridView1.DataBind();

            SqlCommand UpCom = new SqlCommand();
            UpCom.Connection = sqlCON;
            DBAdapter.UpdateCommand = UpCom;
            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");

            SqlParameter parameter =
DBAdapter.UpdateCommand.Parameters.Add("@UserId", SqlDbType.Int);
            parameter.SourceColumn = "UserId";


            DBAdapter.UpdateCommand = UpCom;

            Random rnd = new Random();

            for (int i = 0; i <= ResultsDataSet.Tables[0].Rows.Count - 1; i++)
                ResultsDataSet.Tables[0].Rows[i]["Amount"] = rnd.Next(100); ;

            DBAdapter.Update(ResultsDataSet, "fullTable");
 }
```

عند الضـــــغط على الزر Update Amount نلاحظ تغير القيم في العمود Amount

(ليس من الضرورة ظهور نفس الأرقام لدى الطالب نظراً لاستخدام Random)

UserId	Login	Password	EMail	BirthDate	Amount
10	titi	456	titi@titi.com	9/9/1990 12:00:00 AM	52
119	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	9
120	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	49
121	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	10
122	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	61
123	tata	789	tata@tata.com	1/1/1970 12:00:00 AM	1

userid	login
10	titi
119	tata
120	tata
121	tata
122	tata
123	tata

Add row Delete row 1 Update Amount

*****