

قواعد البيانات وعمليات Select, Insert, Update, Delete بسهولة في C#

بالنسبة للغة C# عملية الإتصال بقواعد البيانات تحتاج للكثير من العول والتعب فيجب أولاً انشاء كائن اتصال من نوع OleDbConnection وبعدها كائن استعلام من نوع OleDbCommand ثم يجب أن تشحن كائن الاتصال بمسار قاعدة البيانات وتعليقات الاتصال التي تختلف حسب نوع قاعدة البيانات مثلاً إذا كانت قاعدة البيانات من نوع Access فتكون تعليمات الإتصال كمايلي:

```
1 "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + fileName + ";;";
```

حيث أن fileName هو مسار قاعدة البيانات, وبعدها يجب أن تشحن كائن الاستعلام بقيمة الاستعلام الذي تريد تنفيذه على قاعدة البيانات من إضافة أو حذف أو تعديل وبعدها اسناد كائن الاتصال إلى كائن الاستعلام وبالنهاية تنفيذ الاستعلام وأخذ النتيجة سواء كانت رقم يمثل عدد السجلات المتأثرة أو جدول يمثل نتيجة استعلام select تلك الطريقة تبدو جيدة ولكن مع مرور الوقت ومع تكرارها مع كل مشاريعك التي تحتاج إلى قواعد بيانات سيصبح الأمر مهلاً جداً, وعندها فكرت بطريقة لتسهيل الموضوع وبحثت على شبكة الإنترنت ووجدت مقطع الفيديو التالي:

[بناء الـ Class الهمهي DB لتشغيل جهل Insert, Update, delete, Select](#)

ومن خلاله توصلت إلى طريقة جيدة لإنشاء صف DataBase يقوم بكل العمليات الروتينية السابقة فكان كالتالي

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Collections;
5 using System.Data;
6 using System.Windows.Forms;
7 using System.Data.OleDb;
8
9 namespace DataBase
10 {
11     class AccessDB
12     {
13         //Fields
14         string ConnectionString;
15         OleDbConnection conn;
16         OleDbCommand cmd;
17
18         //properties
19         public string Connection { set { ConnectionString = value; } get { return
20 ConnectionString; } }
21     }
22 }
23 }
```

بالبداية تعريف اسم الصف وكائن الاتصال وكائن الاستعلام وخاصية من أجل التعامل مع كائن الاتصال...

بعدها نحتاج إلى دالة الباني وهي كما يلي:

```
1 public AccessDB(string fileName)
2     {
3         ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source='" + fileName + "'";
4     }
```

هذا الباني يأخذ معامل واحد عبارة عن نص يمثل مسار قاعدة البيانات وبالتالي عند التعامل مع هذا الصف وإنشاء كائن منه فإن أول عملية هي تحرير مسار قاعدة البيانات وإنشاء كائن الإتصال...
يوجد بعض العمليات الروتينية الضرورية التي تحتاجها في تطبيقات قواعد البيانات مثل الاتصال بقاعدة البيانات وفتح الاتصال أو التأكد من صحة الاتصال وهذه العمليات تتكرر تقريباً بكل عمليات الاستعلامات لذلك سنضعها في دالة خاصة كما يلي:

```
1 private void Launch(CommandType CT, string Ctxt)
2     {
3         try
4         {
5             // اتصال بالقاعدة وإنشاء كائن الاتصال
6             conn = new OleDbConnection(ConnectionString);
7             conn.Open();
8             cmd = new OleDbCommand();
9             conn.Close();
10        }
11        catch (Exception x)
12        {
13            MessageBox.Show("خطأ بالقاعدة" + x.Message);
14            System.Environment.Exit(0);
15        }
16
17        cmd.Connection = conn;
18        cmd.CommandType = CT;
19        cmd.CommandText = Ctxt;
20        conn.Open();
21    }
```

بعد الاتصال بالقاعدة وتحديد جملة الاستعلام نحتاج أيضاً إلى عمليات أخرى روتينية مثل عملية تنفيذ الاستعلام وإغلاق الإتصال بقاعدة البيانات ومثل العمليات السابقة سنضعها في دالة خاصة:

```
1 private int MakeItWork()
2     {
3         try
4         {
5             int n = cmd.ExecuteNonQuery();
6             conn.Close();
7             return n;
8         }
9         catch
```

```

10     {
11         conn.Close();
12         return -1;
13     }
14 }

```

والآن نحتاج إلى الدوال التي تنفذ الإستعلامات, بها أن عملية select مختلفة عن عمليات insert,update,delete سنحدد دالة خاصة لها:

```

1 public DataTable RunQuery(string query)
2 {
3     Launch(CommandType.Text, query);
4     DataTable dt = new DataTable();
5     dt.Load(cmd.ExecuteReader());
6     conn.Close();
7     return dt;
8 }

```

هذه الدالة تأخذ نص عبارة عن الإستعلام المراد تنفيذه وتعيد جدول من نوع DataTable يحوي النتائج... ونحتاج بالنهاية إلى دالة لتنفيذ عمليات insert,update,delete إلى دالة خاصة بهم كما يلي:

```

1 public int RunUpdateQuery(string InsUpdDel)
2 {
3     Launch(CommandType.Text, InsUpdDel);
4     return MakeItWork();
5 }

```

وبهذا يكون الصف الخاص بالتعامل مع قواعد البيانات جاهزاً, وللتعامل مع قواعد البيانات باستخدام هذا الصف يمكنك تنفيذ مشابهة للمثال التالي:

```

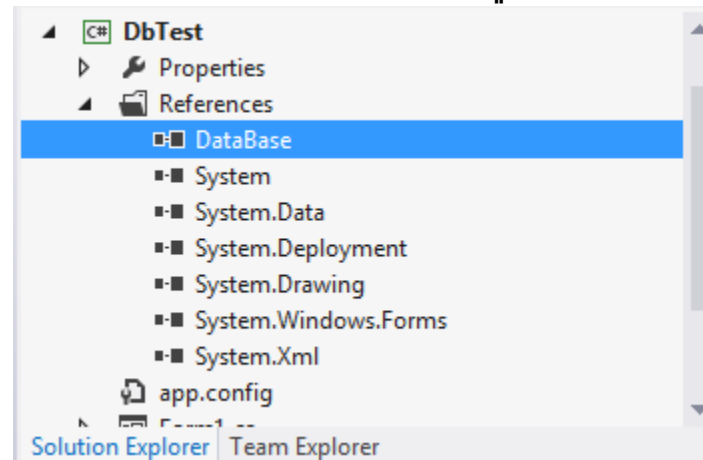
1 AccessDB newDb = new AccessDB("D://customers.mdb");
2
3 string insertQuery = "insert into Customers Values (1,'Fuad','Prince')";
4 newDb.RunUpdateQuery(insertQuery);

```

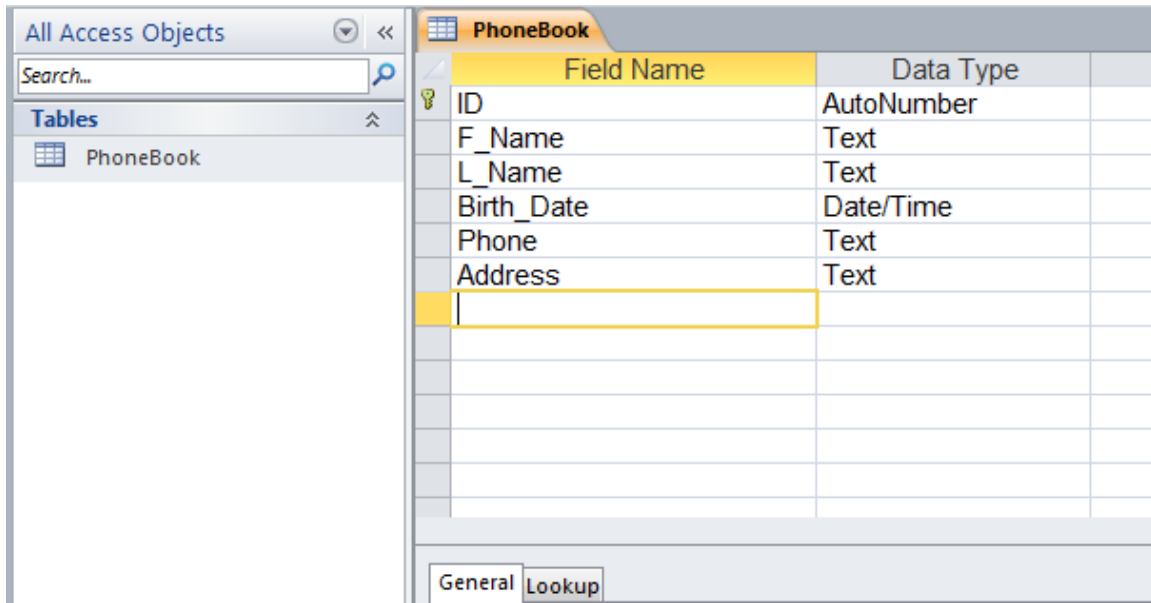
طبعاً هذا الصف خاص بقواعد البيانات من نوع MS Access أما إن كنت تستخدم قواعد بيانات من نوع آخر (مثل SQL SERVER) فيجب أن تقوم ببعض التعديلات على هذا الصف

لنقم بهثال تطبيقي لهذا الصف كالشكل التالي:

بعد انشاء مثل تلك الواجهة قم بإضافة المكتبة البرمجية للصف DataBase الذي أنشأناه في قائمة الـ References الخاصة بهذا المشروع كما يلي:



ثم أنشئ قاعدة بيانات ببرناج MS Access ولتكن كما يلي:



والآن بقي وضع التعليمات الخاصة باستخدام الصف السابق
أولاً استورد المكتبات الخاصة كما يلي:

```
using System.Drawing;
using System.Text;
using System.Windows.Forms;

using System.Data.OleDb;
using DataBase;
```

```
namespace DbTest
{
    public partial class Form1 : Form
    {
```

ثم في أي مكان في الصف Form1 قم بتعريف كائن من نوع AccessDB كما يلي:

```
using System.Data.OleDb;
using DataBase;
```

```
namespace DbTest
{
    public partial class Form1 : For
    {
        DataBase.AccessDB AcData;
        public Form1()
        {
            InitializeComponent();
```

وبعدها في دالة تحميل النافذة ضع الشيفرة التالية لإنشاء الكائن وشحنه بهسار قاعدة البيانات:

```
private void Form1_Load(object sender, EventArgs e)
{
    string path=System.IO.Directory.GetCurrentDirectory();
    path += @"phoneBook.mdb";

    AcData = new DataBase.AccessDB(path);
}
```

ولتطبيق عملية الإضافة ضع الشيفرة التالية في حدث النقر على زر "إضافة جديد":

```
private void button1_Click(object sender, EventArgs e)
{
    if (CheckData())
    {
        string query =
            String.Format("INSERT INTO PhoneBook(F_Name,L_Name,Birth_Date,Phone,Address) VALUES ('{0}','{1}','{2}','{3}'");
        int rst = AcData.RunUpdateQuery(query);
        if (rst >= 0)
            MessageBox.Show("تمت الإضافة بنجاح", "نجاح", MessageBoxButtons.OK, MessageBoxIcon.Information);
        else
            MessageBox.Show("حصل خطأ أثناء الإضافة يرجى التأكد من البيانات", "خطأ", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
        MessageBox.Show("يرجى ملء جميع الحقول");
}
```

طبعاً في البداية عليك أن تفحص المدخلات فإن كانت ناقصة عليك أن تنبه المستخدم لذلك, حيث أنني أنشأت دالة خاصة من أجل عملية فحص المدخلات وهي الدالة CheckData(). وإذا تمت العملية بنجاح فسيتم إضافة السجل الجديد.

والآن من أجل عملية البحث نضع الشيفرة التالية في الحدث النقر على زر "بحث بالإسم":

```
private void button2_Click(object sender, EventArgs e)
{
    if (fname.Text != "")
    {
        string query = String.Format("SELECT ID,F_Name,L_Name,Birth_Date,Phone,Address FROM PhoneBook WHERE F_Name = '{0}'");
        DataTable dT = AcData.RunQuery(query);
        if (dT.Rows.Count > 0)
        {
            MessageBox.Show("تم إيجاد " + dT.Rows.Count + " نتيجة");
            int id=0;
            for (int i = 0; i < dT.Rows.Count; i++)
            {
                id=Convert.ToInt32(dT.Rows[i][0].ToString());
                fname.Text = dT.Rows[i][1].ToString();
                lname.Text = dT.Rows[i][2].ToString();
                date.Text = dT.Rows[i][3].ToString();
                phone.Text = dT.Rows[i][4].ToString();
                address.Text = dT.Rows[i][5].ToString();
            }
        }
        else
            MessageBox.Show("لم يتم إيجاد أي نتيجة");
    }
    else
        MessageBox.Show("يرجى ادخال الإسم");
}
```

وإذا تمت العملية بنجاح سيعرض رسالة توضح عدد النتائج وستمتلئ الحقول ببيانات آخر نتيجة. وبنفس طريقة البحث يمكنك تطبيق الحذف والتعديل ولكن لأنني متكاسل حالياً سأترك هاتين العمليتين لك.

بالمرفقات توجد جميع الملفات التي عملنا عليها.

أتمنى أن تكون هذه التدوينة مفيدة