التعامل مع قواعد البيانات ADO.Net

مقدمة

تعتبر قواعد البيانات أساسية في معظم المشاريع . فتعلمها أمرا ملحا لتنفيذ المشاريع التجارية و الإدارية ... التي تتيح للمستخدم استرجاع البيانات و تعديلها و حفظها و حذفها و البحث عنها .

تعريف قواعد البيانات :

 هي ملف تستخدم لتخزين و حفظ البيانات في جداول منظمة و مترابطة يسهل التعامل معها وهذا باستعمال برامج معينة مما ييسر لك البحث و التعديل و الحفظ و الإضافة و حذف البيانات بكل سهولة و بدون أخطاء . هذه البرامج تعرف بأنظمة إدارة قواعد البيانات Data base Mangement System (DMS)

وهناك أنواع كثيرة من قواعد البيانات منها :

* قواعد البيانات ACCESS
* قواعد البيانات SQL Server
* قواعد البيانات Oracle
* قواعد البيانات MYSQL

ومن المؤكد أن كل هذه البرامج تستعمل لغة فريدة عالمة مشتركة للتعامل مع البيانات تعرف بلغة الاستعلامات المهيكلة

Structure Query Langage (SQL)

و تحتوي قواعد البيانات على عناصر أساسية منها :

* الجداول Tables : يتم تخزين وحفظ البيانات فيها و تتكون من أعمدة Columns أو ما يعرف بالحقول Fields و الصفوف Rows أو ما يسمى بالسجلات Records
* العلاقات Relation تربط بين الجداول لتفادي التكرار و تنظيم البيانات
* القيود Constraint و هي شروط تخص الحقول
* العروض View وهي مثل الجداول تقوم بعرض البيانات اعتماد على جداول موجودة

التعامل مع قواعد البيانات ADO.Net

قواعد البيانات ADO.Net هي وسيلة المثلى لربط مصادر البيانات ببرنامج Vb.Net و ذلك لإنجاز المشاريع الخاصة بقواعد البيانات

مكونات ADO.Net

يتكون من عدة فئات و من أهمها :

1. فئة الاتصال Connection : تمكنك من الاتصال بمصدر البيانات
2. فئة الأوامر Command : تمكنك من تنفيذ الأوامر الناتجة عن جملة استعلام أو إجراء مخزن
3. فئة قارئ البيانات DataReader : تمكنك من قراءة السجلات الناتجة من تنفيذ كائن الأوامر
4. فئة موصل البيانات DataAdapter : تقوم بحلقة وصل بين مصدر البيانات و مجموعة البيانات dataSet
5. مجموعة البيانات Data Set : عبارة عن نسخة مصغرة يتم فيها تخزين البيانات في الذاكرة المؤقتة للجهاز

وبذلك يمكنك التعامل معها دون تأثير على مصادر البيانات الأصلية

الوضع المتصل و الوضع المنفصل

الوضع المتصل **Connected Mode :** و يتم فيه الاتصال المباشرة بين مصادر البيانات و الكائن ADO فأي تحديث أو تغيير للبيانات من خلال التعليمات البرمجية يتم مباشرة و آنيا و يستعمل الكائنات التالية :

1. كائن الاتصال Connection للاتصال بمصادر البيانات
2. كائن الأوامر Command لتنفيذ الأوامر
3. كائن قارئ البيانات DataReader لقراءة البيانات إلى الأمام فقط

الوضع المنفصل Disconnected Mode: فيمكنك الاتصال بقاعدة البيانات وتخزينها في الذاكرة كقاعدة بيانات مصغرة في ما يعرف بمجموعة البيانات Data Set و القيام بجميع العمليات كالتحديث و الإضافة و الحذف و التعديل و البحث و إنشاء جداول و الربط بينها ....

و تنتهي علاقتك بمصدر البيانات الأصلي و تتعامل مع البيانات دون تأثير على البيانات الأصلية . و هذا تخفيفا على ما يعرف ب Server و يتم هذا الوضع على النحو التالي :

1. الاتصال بقاعدة البيانات عن طريق الكائن Connection
2. نأخذ نسخة مصغرة من قاعدة البيانات الأصلية و خزنها في مجموعة البيانات Data Set
3. و يتم هذا عن طريق محول البيانات Data Adopter حيث يمثل وسيط بين قاعدة البيانات و مجموعة البيانات Data Set حيث يقوم بفتح الاتصال و تنفيذ الأوامر و تعبئة Data set فتصبح مستقلة عن مصدر البيانات وهذا الانفصال هو سر تسمية هذا الوضع بالوضع المنفصل حيث يمكنك القيام بجميع عمليات التحديد و الإنشاء دون تأثير على البيانات في القاعدة الأصلية .

و هذا الشكل يمثل مكونات كل من الوضع المتصل و المنفصل



مزودات البيانات NET Data Providers

ما هي إلا حلقة وصل بين قاعدة البيانات و Ado.Net . و هي أنواع حسب نوع قاعدة البيانات التي تتعامل معها

أنواع مزودات البيانات :

توجد أربعة أنواع مدعومة في إطار عمل .NET Framework حسب نوع قاعدة البيانات التي تتعامل معها و هي :

1. OLE DB .NET Data Provider : O bject L inking and E mbeding D ata B ase

 مزود خاص الاتصال بصورة عامة مع جميع أنواع قواعد البيانات MS Access و مصادر البيانات المبنية على تقنية COM المنجزة قبل تقنية Microsoft .NET

و يندرج تحت فضاء الأسماء system Data.OLEDB

2- SQL Server .NET Data Provider

 مزود خاص بقواعد البيانات SQL Server المرفقة بالإصدار الحالي

 و يندرج تحت فضاء الأسماء system Data .SQLClien أو system Data.SQL

1. ODBC .NET Data Provider التواصل المفتوح مع قواعد البيانات

 O pen D ata B ase C onnectivity ( Odbc ) Data Provider

 و هي مزودات تعتمد تقنية قديمة تعتمد على استخدام مشغلات ODBC Drivers

 و يندرج تحت فضاء الأسماء System Data . ODBC

1. Oracle.Net Data Provider



كائن الاتصال Connection Object

 يتح لك هذا الكائن الاتصال بقاعدة أو مصدر البيانات و هذا يرتبط ارتباطا وثيقا مع نوع مزود البيانات المستعمل . و كما يتطلب معلومات كافية عن الاتصال كمسار قاعدة البيانات و كلمة مرور و اسم المستخدم ... و هذا ما يعرف بنص الاتصال Connection String .

إنشاء كائن الاتصال :

لا تنسى استراد مجال الأسماء الخاص بالمزود الذي تريد التعامل معه أعلى فئة النموذج أو الموديل

Imports System.Data.OleDb ' مزود Access

Imports System.Data.SqlClient ' مزود SQL Server

الصيغة 1 : بدون معاملات

Dim ConAccess As New OleDbConnection 'كائن الاتصال ب ACCESS

Dim ConSql As New SqlConnection 'كائن الاتصال ب SQL Server

الصيغة2 : بإضافة نص يمثل نص الاتصال Connection String

Dim ConAccess As New OleDbConnection("Connection String")

Dim ConSql As New SqlConnection("Connection String")

نص الاتصال Connection String

هو عبارة عن عوامل نصية تمكنك بالاتصال بقاعدة البيانات و تتمثل في مسار و اسم قاعدة البيانات و المزود و كلمة المرور و اسم المستخدم ...يفصل بين كل عامل بفاصلة منقوطة .

و الجدول التالي يوضح لنا عوامل نص الاتصال مع OLEDB

|  |  |
| --- | --- |
| العامل | دوره |
| Provider | اسم مزود البيانات الخاص ب OLEDB"Provider =Microsoft.Jet.Oledb.4.0;" ' ACCESS 2003"Provider = Microsoft.ACE.Oledb.12.0;" Access 2007 |
| Data Source | مسار قاعدة البيانات كاملا |
| **PersistSecurityInfo** | ابقاء المعلومات سرية مع قيمتها الافتراضية False و يستوجب عليك تحديد كلمة المرور و اسم المستخدم إما إذا كانت True فيمكنك تحديد كلمة المرور و المستخدم عند فتح الاتصال و يبقى محافظا عليها  |
| passWord | كلمة المرور |
| User Id  | اسم المستخدم |
| Connection Timeout | مدة الانتظار عند فتح الاتصال |

أما عوامل نص الاتصال مع SQL Server يوضحها الجدول التالي

|  |  |
| --- | --- |
| العامل | دوره |
| Data sourceServer | اسم مزود البيانات مع SQL Server "Data source =.\SQLEXPRESS " ' مزود SQL server 2005  |
| **Initial Catalog**dataBase | اسم قاعدة البيانات المتصلة بالخادم دون ذكر المسار و الامتداد |
| AttachDbFilenameinitial File Name | مسار قاعدة البيانات الغير متصلة بالخادم كاملا |
| Integrated Security | حماية متكاملة و قيمتها الافتراضية False وهنا يتوجب عليك ذكر كلمة المرور و اسم المستخدم  |
| passWord | كلمة المرور |
| User Id  | اسم المستخدم |
| Connection Timeout | مدة الانتظار عند فتح الاتصال |

ربط كائن الاتصال بنص الاتصال

هناك طريقتان : الطريقة الأولى اسناد نص الاتصال مباشرة عند إنشاء كائن الاتصال

Dim ConAccess As New OleDbConnection("......................")

الطريقة 2

Dim ConAccess As New OleDbConnection()

ConAccess.ConnectionString = ".................."

و بنفس الكيفية مع SQL Server فقط نستبدل OLEDB ب SQL

الاتصال بقاعدة البيانات من نوع OLEDB

الاتصال بقاعدة البياناتا Access 2003 غير محمية

Dim ConAccess As New OleDbConnection("Provider = Microsoft.Jet.Oledb.4.0;" & \_

 "Data source =E:/ MyDataBase.mdb;" & \_

 "Persist Security Info=False;")

الاتصال بقاعدة بيانات Access 2003 محمية بكلمة مرور

Dim ConAccess As New OleDbConnection("Provider =Microsoft.jet.oledb.4.0;" & \_

 "Data source = E:/ MyDataBase.mdb;" & \_

 "Persist Security Info=True;" & \_

 "Jet Oledb:Database Password=PWord; User Id =UserName;")

الاتصال بقاعدة البيانات Access 2007 محمية بكلمة مرور

Dim ConAccess As New OleDbConnection("Provider =Microsoft.ACE.oledb.12.0;" & \_

 "Data source = E:/ MyDataBase.mdb;" & \_

 "Persist Security Info=True;" & \_

 "Jet Oledb:Database Password=PWord; User Id =UserName;")

الاتصال بقاعدة البيانات بواسطة SQL Server

الاتصال بقاعدة البيانات SQL Server 2005 متصلة بالخادم و غير محمية

Dim StrSQlConn As String = "Data source =.\SQLEXPRESS;" & \_

 "Database = MyDatabase;" & \_

 "Connect Timeout = 10;" & \_

 "Integrated Security = True;")

و الان الاتصال بقاعدة بيانات Sql Serverو لكنها محمية بكلمة مرور

Dim StrSQlConn As String = "Data source =.\SQLEXPRESS;" & \_

 **"Database = MyDatabase;" & \_**

 "Connect Timeout = 10;" & \_

 "Integrated Security = False;" & \_

 "PassWord =MyPassword;" & \_

 "User Id =MyUser;")

الاتصال بقاعدة البيانات SQL Server غير متصلة بالخادم و غير محمية بكلمة مرور

نص الاتصال ب SQL Server 2005 بقاعدة بيانات غير متصلة بالخادم

Dim StrSQlConn As String = "Data Source=.\SQLEXPRESS;" & \_

 "AttachDbFilename=E:\MyDataBase.mdf;" & \_

 "Integrated Security=True;" & \_

 "Connect Timeout = 10;")

و إذا كانت محمية بكلمة مرور نكتب

Dim StrSQlConn As String = "Data Source=.\SQLEXPRESS;" & \_

 "AttachDbFilename=E:\MyDataBase.mdf;" & \_

 "Integrated Security=False;" & \_

 "Connect Timeout = 10;" & \_

 "PassWord =MyPassword;" & \_

 "User Id =MyUser;")

الاتصال بقاعدة بيانات ODBC

الاتصال بقاعدة بيانات Access

Dim ConAccess As New OdbcConnection("Driver={Microsoft Access Driver (\*.mdb)};" & \_

 "DBQ=E:/MyDataBase.mdb;")

الاتصال بقاعدة بيانات Excel

Dim ConAccess As New OdbcConnection("Driver={Microsoft Excel Driver (\*.xls)};" & \_

 "DBQ=C:\Book1.xls;")

لا شك و أنك تجد صعوبة في كتابة نص الاتصال ناهيك عن الأخطاء الإملائية. فما هو الحل ؟

الحل هو كائن باني نص الاتصال **DbConnectionStringBuilder**

فئة باني نص الاتصال : DbConnectionStringBuilder

تقوم ببناء و تكوين صيغة و عوامل نص الاتصال الخاص بمزودات OLEDB

خصائصها :

|  |  |
| --- | --- |
| الخاصية | دورها |
| **ConnectionString** | تقرأ أو تغير نص الاتصال التابع لهذه الفئة |
| **DataSource** | تقرأ أو تغيير مسار قاعدة البيانات التي تم الاتصال بها |
| **Provider** | تقرأ و تغير نص يعبر عن مزود البيانات Oledb |
| **PersistSecurityInfo** | ابقاء المعلومات سرية وقيمتها في الوضع الافتراضي False مما يستوجب عليك تحديد كلمة المرور و اسم المستخدم كلما فتحت الاتصال إما إذا كانت قيمتها True فيمكنك إرسال هذه المعلومات عند فتح الاتصال لأول مرة و سيتم الاحتفاظ بها ليتم فتح الاتصال بعدها |

مثال : يوضح ما قلناه

Imports System.Data.OleDb

Public Class Form1

 Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

 Dim BCS As New OleDbConnectionStringBuilder()

 BCS.Provider = "Microsoft.jet.oledb.4.0;"

 BCS.DataSource = "E:/MyDataBase.mdb"

 BCS.PersistSecurityInfo = True

 Console.WriteLine("Provider = " & BCS.Provider)

 Console.WriteLine("Data source = " & BCS.DataSource)

 Console.WriteLine("Connection String = " & BCS.ConnectionString)

Dim ConAccess as New OleDbConnection(BCS.ConnectionString)

 End Sub

End Class

فئة باني نص الاتصال SqlConnectionStringBuilder

تقوم ببناء و تكوين صيغة و عوامل نص الاتصال الخاص بمزودات SQL server

أهم الخصائص

|  |  |
| --- | --- |
| الخاصية | دورها |
| **DataSource** | تقرأ و تغير اسم مزود البيانات SQL Serverو هي تماثل Data Source أو Server في نص الاتصال |
| **IntegratedSecurity** | حماية متكاملة Integrated Security في نص الاتصال و قيمتها الافتراضية False في هذه الحالة يتوجب عليك تحديد كلمة المرور و المستخدم أما إذا كانت true فلا يلزم ذلك |
| **InitialCatalog** | تقرأ و تغير اسم قاعدة البيانات المتصلة بالخادم دون ذكر المسار و هي تماثل Initial Catalog أو Database في نص الاتصال |
| **AttachDBFilename** | تقرأ و تغير مسار قاعدة البيانات المتصل بها و تماثل AttachDBFileName أو initial File Name في نص الاتصال |
| **ConnectTimeout** | تقرأ و تغير مدة انتظار الاتصال بالخادم  |
| **Password** | تقرأ و تغير كلمة المرور في حالة عدم استخدام الحماية المتكاملة |
| **UserID** | تقرأ و تغير اسم المستخدم في حالة عدم استخدام الحماية المتكاملة و هي تماثل User Id في نص الاتصال |
| **ConnectionString** | تقرأ و تغير نص الاتصال التابع لهذه الفئة  |

مثال: يوضح ما قلناه في حالة الاتصال بالخادم

Imports System.Data.SqlClient

Public Class Form1

 Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

 Dim build As New SqlConnectionStringBuilder

 build.DataSource = ".\SQLEXPRESS"

 **build.InitialCatalog = "MyDataBase"**

 build.IntegratedSecurity = True

 build.Password = ""

 build.UserID = ""

 build.ConnectTimeout = 15

 Console.WriteLine("Server =" & build.DataSource)

 Console.WriteLine("Database = " & build.InitialCatalog)

 Console.WriteLine("Time wait = " & build.ConnectTimeout)

 Console.WriteLine("Connection String = " & build.ConnectionString)

 Dim ConSQL as New SQLConnection(build.ConnectionString)

 End Sub

End Class

أما في حالة الاتصال بقاعدة البيانات SQL Server غير متصلة بالخادم

نغير السطر العريض في المثال السابق ب

build.AttachDBFilename = "E:/MyDataBase.mdf"

كما يمنحك ADO.Net مرونة و سهولة أكثر في بناء نص الاتصال و هذا عن طريق إعدادات المشروع Setting

بناء نص الاتصال عن طريق إعدادات المشروع Settings

اتبع الخطوات التالية :

1. من نافذة مستكشف الحل اضغط مرتين على الكائن My Project تظهر نافذة خصائص المشروع
2. اختر العنصر Setting من الهامش الأيسر
3. يظهر جدولا في خانة Name اكتب اسم نص الاتصال و ليكن StrConn
4. من العمود Type اختر Connection String
5. من خانة Value انقر زر الاستعراض ليظهر مربع حوار تتبع الخطوات كما تعلمنا سابقا
6. و بذلك يتم توليد نص الاتصال



1. ثم نكتب

Dim ConAccess As New OleDbConnection(My.Settings.StrConn)

خواص كائن الاتصال :

|  |  |
| --- | --- |
| الخاصية | دورها |
| ConnectionString | تقرأ و تغير نص الاتصال  |
| ConnectionTimeout | تعيد مدة الانتظار عند الاتصال بقاعدة البيانات |
| Database | تعيد اسم قاعدة البيانات التي يتم التعامل معها حاليا وهي للقراءة |
| DataSource | تعيد مسار قاعدة البيانات و هي للقراءة |
| Provider | تعيد اسم مزود المستعمل بالكامل و هي للقراءة |
| State | تعيد حالة الاتصال ) مفتوح – مغلق – في حالة الفتح ....( |

و هذا المثال يوضح لنا هذه الخصائص و كيفية الاستعلام عنها

Imports System.Data.SqlClient

Public Class Form1

 Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

 Dim SQLConn As New SqlConnection("Data source =.\SQLEXPRESS;" & \_

 "Database = MyDataBase;" & \_

 "Integrated Security=True;" & \_

 "Connect Timeout = 10;")

 Console.WriteLine("Server = " & SQLConn.DataSource) ' Server =.\SQLEXPRESS

 Console.WriteLine("DataBase = " & SQLConn.Database) ' database = MyDataBase

 Console.WriteLine("wait = " & SQLConn.ConnectionTimeout) ' Wait = 10

 End Sub

End Class

الخاصية State :

 تعيد قيمة من قيم المرقم Connection State و التي تعبر عن حالة الاتصال حسب الجدول التالي

|  |  |
| --- | --- |
| الحالة |  |
| Closed | تم غلق الاتصال |
| Open | الاتصال مفتوح |
| Connecting | يتم الاتصال حاليا |
| Executing | يتم تنفيذ أحد الأوامر حاليا |
| Fetching | يتم إحضار البيانات من قاعدة البيانات حاليا |
| Broken | تم الفتح لكن حدث خطأ أدى إلى غلق الاتصال |

وهذا مثال يوضح ما قلناه

If con.State = ConnectionState.Open Then

 MsgBox("تم فتح القاعدة بنجاح")

 ElseIf con.State = ConnectionState.Closed Then

 MsgBox("تم غلق القاعدة بنجاح")

 ElseIf con.State = ConnectionState.Connecting Then

 MsgBox("جاري الاتصال")

 ElseIf con.State = ConnectionState.Executing Then

 MsgBox("يتم تنفيذ الأمر")

 ElseIf con.State = ConnectionState.Fetching Then

 'اي شئ تريد تنفيذه هنا

 End If

وسائل كائن الاتصال

|  |  |
| --- | --- |
| الوسيلة | دورها |
| BeginTransaction | بداية العملية الواحدة |
| ChangeDatabase | تغيير قاعدة البيانات |
| CreateCommand | إنشاء أمر |
| Close | لغلق الاتصال بقاعدة البيانات |
| Dispose | محو كائن الاتصال من الذاكرة |
| Open | فتح الاتصال |

فتح الاتصال و غلقه

لفتح الاتصال بقاعدة البيانات استعمل الوسيلة Open و لغلقه استعمل الوسيلة Close

Dim Con As New OleDbConnection()

 Con.ConnectionString = "Provider =Microsoft.jet.oledb.4.0;" & \_

 "data source =E:/MyDataBase.mdb"

 Con.Open()

 Console.WriteLine(Con.State.ToString) 'Open

 Con.Close()

 Console.WriteLine(Con.State.ToString) 'Closed

اتقاء شر الاستثناءات عند الاتصال بقاعدة البيانات

استعمل طريقة صيد الاستثناءات try …..Cach….. و لها الصيغة التالية :

Try

{ يحتمل الخطأ مقطع من الكود }

Catch اسم\_الاستثناء As نوع الاستثناء

{ مقطع معالجة هذا النوع من الاستثناءات }

Finally

{ مقطع إلزامي، ينفّذ في كلّ الأحوال}

End Try

Sub TryCatchConnection()

 'نص الاتصال

 Dim ConString As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/MyDataBase.mdb"

 Dim Con As New OleDbConnection(ConString) ' إنشاء كائن الاتصال

 Try

 Con.Open() ' فتح الاتصال الذي يحتمل الخطأ

 Catch ex As Exception 'لصيد الأخطاء عامة

 MsgBox(ex.Message) ' صيد الخطأ

 ' عرض رسالة تبين الخطأ أو الاستثناء

 Finally

 Con.Close() ' ضرورة غلق الاتصال

 End Try

 End Sub

و يستحسن استعمال كائن صيد الأخطاء الخاص بقواعد البيانات OleDbException أو SqlException

 Try

 …

 …

 Catch ex As OleDbException ' كائن صيد الاستثناء الخاص بقواعد البيانات OLEDB

 …

 …

 Catch ex As Exception

 …

 …

 End Try

فتح قاعدة البيانات عن طريق مربع الحوار

Sub OpenConnection()

 ' إنشاء كائن الاتصال

 Dim Con As New OleDbConnection

 With OpenFileDialog1 'مربع حوار فتح

 .Title = "فتح"

 .Filter = "All datbase (\*.mdb)|\*.mdb" ' تصفية الملفات الخاصة بقواعد البيانات

 If .ShowDialog = Windows.Forms.DialogResult.OK Then

 Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & \_

 "Data Source = " & OpenFileDialog1.FileName ' اختر القاعدة من مربع الحوار

 Con.Open() ' افتح الاتصال

 If Con.State Then MsgBox("تم فتح الاتصال بنجاح")

 End If

 End With

End sub

فتح اتصال مشترك بين قاعدتين

Sub OpenPoolConnection()

Dim StrCon1 As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "Data source = E:/Db1.mdb"

 Dim StrCon2 As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "Data source = E:/Db2.mdb"

 Dim Con As OleDbConnection = New OleDbConnection

 Con.ConnectionString = StrCon1

 Con.Open() ' يتم فتح الاتصال1

 Console.WriteLine(Con.DataSource) ' قراءة مسار القاعدة1

 Con.Close() 'غلق الاتصال1

 Con.ConnectionString = StrCon2

 Con.Open() 'فتح الاتصال2

 Console.WriteLine(Con.DataSource) ' قراءة مسار القاعدة2

 Con.Close() ' غلق الاتصال2

 Con.ConnectionString = StrCon1

 Con.Open() ' يتم فتح الاتصال1

 Con.Close() 'غلق الاتصال1

 Console.WriteLine(Con.DataSource) ' العودة إلى مسار القاعدة1

 End Sub

تغيير اسم قاعدة البيانات Change DataBase

يمكنك تغيير اسم قاعدة البيانات مع SQL server بقاعدة أخرى متصلة بالخادم عند فتحه و هذا يغنيك عن إعادة كتابة نص الاتصال و كذا غلق و فتح الاتصال في كل مرة وهذا عندما تكون القاعدان غير محميتين بكلمة مرور

Sub ChangeDataBase()

 Dim ConStr As String = "Data source =.\SQLEXPRESS;" & \_

 "DataBase =Db1;" & \_

 "Integrated Security = True;"

 Dim SQLCon As New SqlConnection(ConStr)

 SQLCon.Open()

 SQLCon.ChangeDatabase("dbStud")

 Console.WriteLine(SQLCon.Database) 'Db2

 End Sub

كائن الأوامر Command Object

يستقبل جملة استعلامات مهيكلة SQL أو إجراء مخزنا StoredProcedure ثم يقوم بتنفيذ الأوامر على مصدر البيانات عبر الاتصال المباشر استنادا إلى كائن الاتصال Connection و يرجع كائن قارئ البيانات DataReader

أهم خصائص كائن الأوامر

|  |  |
| --- | --- |
| الخاصية | عملها |
| CommandType | و تحتوي على ثلاثة خيارات تحدد نوع الأمر المراد تنفيذه1. Text : لتنفيذ أوامر جملة الاستعلامات SQL
2. StoredProcedure : لتنفيذ إجراء موجود في قاعدة البيانات
3. TableDirect : للتعامل مع الجدول مباشرة
 |
| CommandText | تستقبل عبارة نصية تمثل أحد الخيارات السابقة للخاصية CommandType |
| Connection | يستقبل اسم كائن الاتصال المستخدم |
| Parameters | تمكنك من إضافة المعاملات |
| Transaction | تنفيذ العملية الواحدة |
| UpdatedRowSource | حفظ بيانات صف المصدر |
| CommandTimeout | تستقبل مدة الانتظار عند التنفيذ |

إنشاء كائن الأوامر Create Command

و لإنشائه نستعمل الكلمة المحجوزة Newهناك أربع صيغ

Dim Cmd As New OleDbCommand ' بدون معاملات

الصيغة الثانية تستقبل معاملا واحدا يمثل نص جملة الاستعلام التي توضع في الخاصية CommandText

Dim Cmd As New OleDbCommand("Commandtext")

الصيغة الثالثة تزيد بمعامل ثان يمثل كائن الاتصال الذي يتم بواسطته تنفيذ الأمر

Dim Cmd As New OleDbCommand("Commandtext", Con)

وهذا مثال يوضح ما قلناه

Imports System.Data.OleDb

Public Class Form1

 Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim Con As New OleDbConnection("Provider =Microsoft.Jet.Oledb.4.0;" & \_

 "Data source = E:/DbEcole.mdb;")

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand

 Cmd.Connection = Con

 Cmd.CommandType = CommandType.Text

 Cmd.CommandText = "Select \* From Students"

 MsgBox("Connection Open And Command Selected .")

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

End Class

خاصية المعاملات Parametres

تضيف معلملات لجملة الاستعلام أو الإجراء المخزن لإدخال قيم متنوعة حسب المراد. و سنتعرف عليها بالتفصيل مع كائن العاملات .

Imports System.Data.OleDb

Public Class Form1

 Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

 Dim Con As New OleDbConnection("Provider =Microsoft.Jet.Oledb.4.0;" & \_

 "Data source = E:/DbEcole.mdb;")

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand

 Cmd.Connection = Con

 Cmd.CommandType = CommandType.Text

 Cmd.CommandText = "Select \* From Students Where Classe =@Classe"

 Cmd.Parameters.AddWithValue("@Classe ", TxtClasse.Text)

 Cmd.ExecuteNonQuery()

 MsgBox("Connection Open And Command Prameters Selected .")

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

وسائل و طرق كائن الاتصال

|  |  |
| --- | --- |
| الوسيلة | عملها |
| ExecuteScalar | تنفيذ الأوامر التي ترجع قيمة و حيدة كعدد الصفوف |
| ExecuteReader | تنفيذ الأوامر التي ترجع مجموعة من الصفوف و قراءتها بواسطة قارئ البيانات |
| ExecuteNonQuery | تنفيذ أوامر التحديث كالإضافة و التعديل و الحذف و التي لا ترجع أي قيمة |
| Cancle | إلغاء تنفيذ الأوامر |
| Create Parameter | إنشاء معامل  |

تنفيذ الأمر ExecuteScalar

هذه الطريقة ترجع قيمة وحيدة تمثل قيمة السجل الأول من العمود الأول من الاستعلام . و تستعمل أيضا لمعرفة عدد السجلات المتأثرة باستعمال دوال التجميع Aggregate Function ، كحساب عدد كل السجلات ، و الوسط الحسابي و أدنى أو أعلى قيمة للسجلات انظر المثال

Private Sub OpenExecuteScalar()

 'نص الاتصال

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim con As OleDbConnection = New OleDbConnection(strConn)

 con.Open() ' فتح الاتصال

 'حساب عدد السجلات

 Dim strSQL As String = "SELECT COUNT(\*) FROM Students"

 ' إنشاء كائن الأوامر

 Dim cmdScalar As OleDbCommand = New OleDbCommand(strSQL, con)

 cmdScalar.CommandType = CommandType.Text 'تحديد نوع الأوامر

 'تنفيذ الأمر الذي يرجع قيمة واحدة

 Dim intCount As Integer = CInt(cmdScalar.ExecuteScalar)

 Label1.Text = intCount

 'Close the OledbConnection

 con.Close()

 End Sub

تنفيذ الأمر الذي يعيد مجموعة من الصفوف ExecuteReader

يمكنك بهذه الطريقة إرجاع مجموعة من السجلات و لقراءتها نستعمل كائن قارئ البيانات

Private Sub OpenExecuteReader()

 'نص الاتصال

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim con As OleDbConnection = New OleDbConnection(strConn)

 con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر

 Dim Cmd As New OleDbCommand("Select \* From Students", con)

 'تنفيذ الأمر

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 **Dr.Read() ' اقر السطر**

 **' عرض رقم و اسم الطالب من الصف الأول**

 **Console.WriteLine(Dr.Item("Id") & " - " & Dr.GetString(1))**

Dr.Close

 Con.Close

 End Sub

تنفيذ الأمر الذي لا يرجع أي قيمة ExecuteNonQuery

تستعمل هذه الطريقة في تنفيذ أوامر التحديث كالتعديل و الحفظ و الحذف و الإضافة و ترجع عددا صحيحا يمثل السجلات المتأثرة عند تنفيذ الأمر. وكذا الأوامر الخاصة بإنشاء الجداول و تعديل بنيتها و إضافة العلاقات بين الجداول ... و لكن لا ترجع أي سجل فقط ترجع 1 دلالة على نجاح العملية و 0 عند فشل العملية و إذا أردت التأكد قم بفتح قاعدة البيانات و لاحظ التغيير. فهذه الحالة نستعمل الطريقة ExecuteNonQuery

1. تنفيذ أوامر حفظ السجلات Update Records

يمكنك حفظ و تعديل البيانات باستعمال جملة الاستعلامات Update و هي على الشكل التالي

Update *table* Set *ColumsList* = *NewValues* Where *Condition*

مثال :

Sub UpdateRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As New OleDbConnection(S)

 ' جملة الاستعلام التي تقوم بتعديل البيانات حسب ما ادخل في صناديق النصوص

 Dim SQL As String = "Update Students Set Prenom ='" & txtName.Text & "' , Date\_N = '" & TxtDbirth.Text & "'" & \_

 "Where Id =" & TxtId.Text ' يتم التعديل وفق رقم الطالب

 Con.Open()

 ' إنشاء كائن الأوامر و ربطه بجملة الاستعلام و بكائن الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con)

 Label1.Text = Cmd.ExecuteNonQuery() ' تنفيذ الأمر

 MsgBox("تم التعديل بالنجاح")

 End Sub

1. تنفيذ أمر إدراج السجلات Insert Records

يمكنك إدراج سجلات عن طريق جملة SQL كما يلي :

Insert Into *table* (Columns\_List) Values (Values\_List)

مثال :

Sub CmdInsertIntoTable()

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As OleDbConnection = New OleDbConnection(strConn)

 ' جملة إدراج صف باستعمل مربعات النصوص

 Dim SQL As String = "Insert Into Students (Id , Prenom , [Date\_N]) " & \_

 "Values (" & TxtId.Text & ",'" & TxtName.Text & "','" & TxtDBirth.Text & "')"

 Try

 Con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر و ربطه بجملة الاستعلامات و كائن الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con)

 'تنفيذ الأمر

 Label1.Text = Cmd.ExecuteNonQuery ' 1 في حالة نجاح العملية

 Catch ex As Exception

 MsgBox(ex.Message)

 Finally

 Con.Close()

 End Try

 End Sub

1. تنفيذ أمر حذف السجلات Delete Records

لحذف سجلات نستعمل جملة الاستعلام المهيكلة Delete و صيغتها بالشكل :

 Delete From *Table* Where *Condition*

مثال :

Sub DeleteRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As New OleDbConnection(S)

 ' جملة حذف الصف حسب القيمة المدخلة في صندوق النص

 Dim SQL As String = "Delete From Students Where Id =" & TxtId.Text

 Con.Open()

 Dim Cmd As New OleDbCommand(SQL, Con)

 Label1.Text = Cmd.ExecuteNonQuery() '

 MsgBox("تم تعديل السجل")

 End Sub

1. التعامل مع جمل تعريف البيانات DDL **Data Definition Language**

و تتمثل في جمل إنشاء الجداول Creat Tables و تعديل بنيتها Alter Table و حذف الجدول نهائيا Drop

مثال : إنشاء جدول كلمة المرور

Sub CreateTable()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 ' جملة إنشاء جدول

 Dim SQL As String = "Create Table Users " & \_

 "(UserName nVarchar(20) ,Pword nVarchar(20));"

 Con.Open() ' فتح الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con) ' ربط كائن الأوامر بجملة الاستعلام و الاتصال

 ' تنفيذ الأمر

 Me.Text = Cmd.ExecuteNonQuery

 End Sub

كائن المعاملات Parameters Object

المعاملات تشبه المتغيرات توضع في جملة الاستعلام المهيكلة SQL للدلالة على قيمة متغيرة لحقل من حقول الجدول

و تختلف صيغة المعاملات حسب نوع مزود البيانات :

1. SQL Server يستعمل الرمز @ في بداية متبوعا باسم المتغير بدون ترك فواصل
2. OLEDB يستعمل الرمز ?

إنشاء المعاملات Creating Parameters

الصيغة العامة لإنشاء المعاملات :

Dim OledbParameter As New OledbParameter(@ParamName, OleDbType, intSize, Expression)

Dim SqlParameter As New SqlParameter(@ParamName, SqlType, intSize, Expression)

ParamName : الاسم البرمجي للمعامل

OleDbType /SqlType: نوع بيانات المعامل intSize : طول أو حجم المعامل / اختياري

Expression : عنوان الحقل يفهمه المستخدم / اختياري

إضافة المعاملات إلى كائن الأوامر بصورة عامة

Dim SQl As String = "Select \* From TableName Where Column =@NameParam"

 Dim Cmd As New OleDbCommand(SQl, Con)

 **Dim Param As New OleDbParameter** **("@NameParam", OleDbType.BSTR, Size)**

 **Param.Value = "…………نص"**

 **Cmd.Parameters.Add(Param)**

و يمكن إنشاء معاملا لكائن الأوامر في سطر واحد

Dim SQl As String = "Select \* From TableName Where Column =@NameParam"

 Dim Cmd As New OleDbCommand(SQl, Con)

cmd.Parameters.Add("@NameParam",SqlType.Varchar,75).Value = "Value"

أو اختصارا :

Cmd.Paramters .AddWithValue(""@NameParam","Value")

و يمكنك التفصيل أكثر كما يلي :

Dim paramUserName As New SqlParameter

Dim paramPassWord As New SqlParameter

paramUserName.Name= "@Param1"

paramUserName.Value = txtUserName.Text

paramPassWord.ParameterName = "@Param2"

paramPassWord.Value = txtPassWord.Text

sqlCommand.Parameters.Add(paramUserName)

sqlCommand.Parameters.Add(paramPassWord)

استعمال المعاملات في تعديل و حفظ البيانات

Sub CmdUpDateTable()

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As OleDbConnection = New OleDbConnection(strConn)

 'جملة تعديل البيانات باستعمال المعاملات

 Dim SQL As String = "Update Students Set Prenom =? ,Date\_N =? " & \_

 "Where Id =?"

 Try

 Con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر و ربطه بجملة الاستعلامات و كائن الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con)

 'إضافة المعاملات و ترتب حسب جملة الاستعلام

 Cmd.Parameters.Add("@Prenom", OleDbType.BSTR).Value = TextBox2.Text

 Dim param As New OleDbParameter

 param.ParameterName = "@Date\_N"

 param.OleDbType = OleDbType.Date

 param.Value = TextBox3.Text

 Cmd.Parameters.Add(param)

 Cmd.Parameters.AddWithValue("@id", TextBox1.Text)

 MsgBox("تمت عملية التعديل بنجاح")

 Label1.Text = Cmd.ExecuteNonQuery ' 1 في حالة نجاح العملية

 Catch ex As Exception

 MsgBox(ex.Message)

 Finally

 Con.Close()

 End Try

 End Sub

حفظ و تعديل البيانات بإدخال قيم مباشرة

Sub UpdateRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 ' تعديل البيانات بإدخال قيم مباشرة

 Dim SQL As String = "Update Students Set Prenom ='وردة' , Date\_N ='11/11/2011'" & \_

 "Where Id =4"

 Con.Open()

 Dim Cmd As New OleDbCommand(SQL, Con)

 Label1.Text = Cmd.ExecuteNonQuery() '

 MsgBox("تم تعديل السجل")

 End Sub

استعمال المعاملات في إدراج السجلات الجديدة

Sub CmdInsertIntoTable()

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As OleDbConnection = New OleDbConnection(strConn)

 ' جملة إدراج صف باستعمل مربعات النصوص

 Dim SQL As String = "Insert Into Students (Id , Prenom , [Date\_N]) " & \_

 "Values (?,?,?)"

 Try

 Con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر و ربطه بجملة الاستعلامات و كائن الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con)

 'تنفيذ الأمر

 Cmd.Parameters.AddWithValue("@id", TextBox1.Text)

 Cmd.Parameters.Add("@Prenom", OleDbType.BSTR).Value = TextBox2.Text

 Dim param As New OleDbParameter

 param.ParameterName = "@Date\_N"

 param.OleDbType = OleDbType.Date

 param.Value = TextBox3.Text

 Cmd.Parameters.Add(param)

 Label1.Text = Cmd.ExecuteNonQuery ' 1 في حالة نجاح العملية

 Catch ex As Exception

 MsgBox(ex.Message)

 Finally

 Con.Close()

 End Try

 End Sub

إدخال البيانات مباشرة

 Sub CmdInsertIntoTable()

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As OleDbConnection = New OleDbConnection(strConn)

 'تحديد جملة إنشاء جدول

 Dim SQL As String = "Insert Into Students (Id , Prenom , [Date\_N]) " & \_

 "Values (33,'مراد','15-01-2001')"

 Try

 Con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر و ربطه بجملة الاستعلامات و كائن الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con)

 'تنفيذ الأمر

 Label1.Text = Cmd.ExecuteNonQuery ' 1 في حالة نجاح العملية

 Catch ex As Exception

 MsgBox(ex.Message)

 Finally

 Con.Close()

 End Try

 End Sub

استعمال المعاملات في حذف السجلات

Sub DeleteRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim Con As New OleDbConnection(S)

 Dim SQL As String = "Delete From Students Where Id =?"

 Con.Open()

 Dim Cmd As New OleDbCommand(SQL, Con)

 Cmd.Parameters.Add("@Id", OleDbType.BigInt).Value = TxtId.Text

 Label1.Text = Cmd.ExecuteNonQuery() '

 MsgBox("تم حذف السجل")

 End Sub

البحث و تصفية السجلات باستعمال المعاملات

 Sub FindRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 Dim XPrenom As String = InputBox("ادخل اسم الطالب")

 ' جملة استعلام البحث تحتوى على معامل1

 Dim SQL As String = "Select \* From Students Where Prenom =?"

 Con.Open()' فتح الاتصال

 Dim Cmd As New OleDbCommand(SQL, Con) ' إنشاء كائن الأوامر

 ' إضافة معامل خاص بالبحث عن طريق الاسم

 Cmd.Parameters.Add("@Prenom", OleDbType.BSTR).Value = Xprenom

 ' تنفيذ الأمر

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 With Dr

 Do While .Read ' قراءة الصفوف في حلقة

 ' عرض النتائج في مربع الصناديق

 TextBox1.Text = .GetInt16(0)

 TextBox2.Text = .GetString(1)

 Loop ' الخروج من الحلقة

 End With

 Dr.Close() ' غلق قارئ البيانات

 Con.Close()' غلق الاتصال

 End Sub

كائن قارئ البيانات Data Reader Object

هو ناتج تنفيذ كائن الأمر لجملة استعلام مهيكلة عند الاتصال بقاعدة البيانات . فهو خاص بالقراءة إلى الأمام فقط و لا يمكنك تعديل أو تحديث البيانات و نستعمل الوسيط Read لقراءة السجلات واحدا بعد الآخر.

أهم خصائص قارئ البيانات

|  |  |
| --- | --- |
| الخاصية | الوظيفة |
| FieldCount | تعيد عدد الحقول المكونة لهذا الجدول |
| HasRows | تعيد True في حالة ما إذا كان يحتوي الحقل على الصفوف |
| Item | تستقبل عنوان الحقل المراد عرض صفوفه |
| RecordsAffected | تعيد عدد السجلات المتأثرة |

أهم الوسائل

|  |  |
| --- | --- |
| الوسيلة | الوظيفها |
| Close | غلق قارئ البيانات |
| GetFieldType | تقرأ نوع بيانات الحقل |
| GetName | تقرأ اسم الحقل |
| GetValue() | تستقبل رقم أو اسم الحقل لعرض سجلاته |
| IsDBNull | إن الحقل فارغ من السجلات وترجع قيمة منطقية |
| NextResult | اقرأ النتيجة التالية و هذا لعرض الصفوف أيضا في الحلقات التكرارية |
| Read | اقرأ الصف التالية و تعيد True في حالة و جود الصف التالي |

و يمنحك كائن قارئ البيانات وسائل للقراءة لقراءة سجلات الحقول و هذا حسب نوع بياناتها و إليك الجدول التالي يبين أهم هذه الوسائل

|  |  |
| --- | --- |
| الوسيلة | الدور |
| GetBoolean | لقراءة القيم المنطقية |
| GetByte | لقراءة البيانات الثنائية |
| GetDateTime | لقراءة بيانات التاريخ و الوقت |
| GetDecimal | لقراءة الأعداد العشرية |
| GetDouble | لقراءة الأعداد المزدوجة |
| GetInt16 | لقراءة الأعداد الصحيحة المفردة |
| GetInt32 | لقراءة الأعداد الصحيحة  |
| etInt64 | لقراءة الأعداد الصحيحة الطويلة |
| GetString | لقراءة البيانات النصية |

إنشاء و تنفيذ كائن قارئ البيانات

 اتبع المراحل التالية :

1. إنشاء كائن الاتصال Connection ثم اربطه بنص الاتصال String Connection
2. إنشاء كائن الأوامر مرفقا بجملة الاستعلام SQL و ربطه بكائن الاتصال
3. استعمل الطريقة ExecuteReader التابعة لكائن الأوامر و التي ترجع كائن قارئ البيانات
4. استعمل الطريقة Read لقراءة صف أو في حلقة لقراءة مجموعة من الصفوف
5. قم بغلق جميع الكائنات )كائن قارئ البيانات و الأوامر و الاتصال(

مثال : قراءة صف واحد فقط

Private Sub OpenExecuteReader()

 'نص الاتصال

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim con As OleDbConnection = New OleDbConnection(strConn)

 con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر

 Dim Cmd As New OleDbCommand("Select \* From Students", con)

 'تنفيذ الأمر

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 **Dr.Read() ' اقر السطر**

 **' عرض رقم و اسم الطالب الأول**

 **Console.WriteLine(Dr.Item("Id") & " - " & Dr.GetString(1))**

Dr.Close

 Con.Close

 End Sub

ولقراءة الصف الموالي نستعمل الخاصية Read وهكذا . في كل مرة تعيد True في حالة و جود الصف

و بكيفية أخرى نستعمل الوسيلة CommandBehavior.SingleRow التابعة لأمر تنفيذ Execute Reader الخاصة بكائن الأوامر

Private Sub OpenExecuteReader()

 'نص الاتصال

 Dim strConn As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/dbEcole.mdb"

 Dim con As OleDbConnection = New OleDbConnection(strConn)

 con.Open() ' فتح الاتصال

 'إنشاء كائن الأوامر

 Dim Cmd As New OleDbCommand("Select \* From Students", con)

 'تنفيذ الأمر

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader(CommandBehavior.SingleRow) ' قراءة الصف الأول فقط

 If Dr.HasRows Then

 While Dr.Read

 Console.WriteLine(Dr(0) & " " & Dr(1))

 End While

 End If

 End Sub

قراءة مجموعة من السجلات

1. في قائمة ListBox أو ComboBox

Private Sub ReadRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 Try

 Con.Open() ' فتح الاتصال

 ' إنشاء كائن الأمر

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader ' تنفيذ الأمر

 Do While Dr.Read ' المرور في حلقة على الصفوف بطريقة

 ListBox1.Items.Add(Dr("id") & " - " & Dr.GetString(1))

 Loop

 Dr.Close() ' غلق قارئ البيانات

 Con.Close() ' غلق الاتصال

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

1. في جدول عرض البيانات Data Grid View

Private Sub LoadDataGridView()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader 'تنفيذ الأمر

 Dim Dt As New DataTable ' إنشاء جدول

 Dt.Load(Dr) ' تحميل الجدول

 DGV1.DataSource = Dt ' عرض بيانات الجدول

 Dr.Close() ' غلق قارئ البيانات

 Con.Close() ' غلق الاتصال

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

1. في قائمة العرض ListView

Sub FillListViewDataReader()

 Dim Con As New OleDbConnection("Provider = Microsoft.jet.oledb.4.0;" & \_

 "Data source = E:/DbEcole.mdb")

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 LView.View = View.Details

 For i As Integer = 0 To Dr.FieldCount - 1

 LView.Columns.Add("Column " & i + 1, 100, HorizontalAlignment.Right)

 Next

 While Dr.Read

 Dim NewItem As New ListViewItem()

 For i = 0 To Dr.FieldCount - 1

 NewItem.SubItems.Add(Dr(i).ToString())

 Next i

 LView.Items.Add(NewItem)

 End While

 Catch ex As Exception

 End Try

 End Sub

التعامل مع القيم الفارغة

نستعمل الوسيلة ISDBNUL فهي تعيد True في حالة ما إذا كان الحقل فارغا لا يحوي سجلات ولو حاولت قراءة محتوى هذا الحقل ستحصل على استثناء و تجنبا لذلك نتأكد أولا هل الخانة فارغة أم لا

 While Dr.Read

 If Not Dr.IsDBNull(0) Then

 Console.WriteLine(Dr(0))

 End If

 End While

البحث عن سجل بواسطة قارئ البيانات

Private Sub FindRecords()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 Try

 Con.Open()'فتح الاتصال

 Dim XName As String = InputBox("ادخل الاسم")

 If XName = "" Then Exit Sub

 ' جملة الاستعلام الخاصة بالبحث

 Dim Cmd As New OleDbCommand("Select \* From Students Where Prenom ='" & XName & "'", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader ' تنفيذ الأمر

 ' الطريقة الأولى

 'With Dr

 ' If .HasRows Then ' في حالة وجود الصفوف

 ' .Read()' اقرأها

 ' Dim N As Integer = Dr.GetInt16(0) 'قراءة الصف 1 من العمود1

 ' Dim P As String = Dr.GetString(1) ' قراءة الصف1 من العمود2

 ' End If

 'End With

' Or

 Do While Dr.Read ' المرور عبر الصفوف في حلقة

 TextBox1.Text = Dr(0) ' عرض الصف من العمود1

 TextBox2.Text = Dr("Prenom") ' عرض الصف من العمود 2

 Loop ' الخروج من الحلقة

 Dr.Close()' غلق قارئ البيانات

 Con.Close()' غلق الاتصال

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

استقراء بيانات الجدول Discabe Table

يمكنك التعرف على أسماء الحقول و نوع بياناتها عن طريق الخاصيتين التابعتين لكائن قارئ البيانات GetName التي ترجع اسم الحقل و GetFieldType التي ترجع نوع بيانات الحقل

Private Sub DiscribeTable()

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S)

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

For I as integer = 0 to dr.filecount -1

ListBox1.Items.Add(Dr.GetName(i).ToString & " | " & Dr.GetFieldType(i).ToString)

 Next

 Dr.Close()

 Con.Close()

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

استعمال الوسيلة NextResult

تستعمل في حالة احتواء جملة الاستعلام على أكثر من جملة و يفصل بينها بفاصلة منقوطة فعند تنفيذ الجملة الأولى يمكنك الانتقال لقراءة و تنفيذ الجملة الثانية باستعمال الوسيلة Next Result وهذا مع مزود SQL Server

Private Sub NextResultDataReader()

 'Define and open the SqlConnection object

 Dim strConn As String = "Data Source=.\SQLEXPRESS;Initial Catalog=DB3;Integrated Security=True;Pooling=False"

 Dim cnn As SqlConnection = New SqlConnection(strConn)

 cnn.Open()

 'Define the SqlCommand to return two resultsets

 Dim strSQL As String = "SELECT ID ,MyName FROM Students"

 strSQL += ";SELECT Arabe ,Math ,France FROM Score"

 Dim cmdReader As SqlCommand = New SqlCommand(strSQL, cnn)

 cmdReader.CommandType = CommandType.Text

 'Define, create, and traverse the SqlDataReader

 Dim sdrReader As SqlDataReader = cmdReader.ExecuteReader

 With sdrReader

 If .HasRows Then

 While .Read

 'Fill a Shippers list box

 ListBox1.Items.Add(.GetValue(0).ToString + " - " + .Item(1).ToString)

 End While

 .NextResult()

 'Process additional resultset(s)

 While .Read

 'Fill an Employees list box

 ListBox2.Items.Add(.Item(0).ToString + " - " + \_

 .Item(1).ToString + " " + .Item(2).ToString)

 End While

 End If

 'Close the SqlDataReader and SqlConnection

 .Close()

 End With

 End Sub

قراءة سجل واحد يمثل الصف الأول

Dim Dr As OleDbDataReader = Cmd.ExecuteReader(CommandBehavior.SingleRow)

قراءة جدول المخطط GetSchemaTable

تعيد كائن الجدول Data Table فارغا تستطيع أن تقر أ بنيته كأسماء الأعمدة و أنواع بياناتها و أحجامها وعرضها في جدول عرض البيانات Data Grid View

Sub GetSchematable()

 Dim Con As New OleDbConnection("Provider = Microsoft.jet.oledb.4.0;" & \_

 "Data source = E:/DbEcole.mdb")

 Try

 Con.Open()

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 Dim schemaTable As DataTable = Dr.GetSchemaTable()

 Dim row As DataRow

 Dim column As DataColumn

 For Each row In schemaTable.Rows

 For Each column In schemaTable.Columns

 Console.WriteLine(String.Format("{0} = {1}", column.ColumnName, row(column)))

 Next

 Next

 Catch ex As Exception

 End Try

 End Sub

**ثانيا : الوضع المنفصل Disconnected Mode**

يمتاز هذا الوضع بكفاءة عالية عند الحديث عن العميل Client و الموزع Server إذا يخفف الازدحام على مصدر البيانات و هذا بتخزين البيانات في الذاكرة في مجموعة البيانات Data Set و لك الحرية في التعامل معها دون تأثير على البيانات المصدرية . إما إذا رغبت في تحديث البيانات فما عليك إلا الاتصال مرة أخرى للتمكن من حفظ التغيرات و هذا بواسطة جمل الاستعلامات المهيكلة أو كائن بناء و توليد الأوامر

سيناريو العمل مع مجموعة البيانات

1. الاتصال بقاعدة البيانات كما في الوضع المتصل
2. أرسل جملة الاستعلام المهيكلة بواسطة موصل البيانات DataAdpater و ربطه بكائن الاتصال هكذا

Dim Da as New OledbDataAdpater(SQL Statement , Con) ' مع مزود Access

Dim Da as New SQLDataAdpater (SQL Statement , Con) ' مع مزود SQL Server

1. تحميل مجموعة البيانات Data Set عن طريق موصل البيانات DataAdpater بواسطة الطريقة Fill
2. غلق الاتصال و التعامل مع البيانات دون تأثير على القاعدة الأصلية

 

مثال : إنشاء مجموعة البيانات

Public Class Form1

 Sub CreateDataSet()

 'نص الاتصال

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 Dim Con As New OleDbConnection(S) 'إنشاء كائن الاتصال

 Dim Ds As New DataSet("Ecole") ' إنشاء مجموعة البيانات

 Try

 Con.Open() ' فتح الاتصال

 Dim SQL As String = "Select \* from Students" ' جملة الاستعلام

 ' إنشاء موصل البيانات وربطه بجملة الاستعلام و كائن الاتصال

 Dim Da As New OleDbDataAdapter(SQL, Con) '

 Da.Fill(Ds, "Students") ' تحميل مجموعة البيانات

 Catch ex As OleDbException

 MsgBox(ex.Message)

 Finally

 Con.Close() 'غلق الاتصال

 End Try

 End Sub

مكونات مجموعة البيانات



أهم خواص مجموعة البيانات

|  |  |
| --- | --- |
| الخاصية | عملها |
| DataSetName | ترجع اسم مجموعة البيانات |
| Tables |  من الخواص التجميعية تمكنك من إضافة جدول أو حذف أو حساب عدد الجداول...  |
| Relation | لإنشاء علاقة |

أهم طرق و وسائل مجموعة البيانات

|  |  |
| --- | --- |
| **الوسيلة** | **دورها** |
| AcceptChanges | قبول التغيرات |
| Clear | مسح محتوى مجموعة البيانات من الجداول |
| Copy | نسخ مجموعة البيانات |
| CreateDataReader | إنشاء قارئ البيانات |
| Dispose | محو مجموعة البيانات من الذاكرة |
| EndInit |  |
| Load | تحميل مجموعة البيانات |
| Merge | دمج مجموعات البيانات مع بعضها |
| Reset | إعادة إلى الحالة الافتراضية |

إنشاء مجموعة بيانات من جديد Create New Data Set

تتكون مجموعة البيانات من مجموعة جداول و لإنشائها لا بد من التطرق إلى الخصائص التجميعية لكائن الجدول التابعة لمجموعة البيانات ) Ds.Tables. …….. (

|  |  |
| --- | --- |
| الخصائص و الوسائل | عملها |
| Add | إضافة جدول لمجموعة البيانات |
| AddRange | إضافة مجموعة جداول إلى مجموعة البيانات مرة واحدة |
| Clear | حذف كل الجداول من مجموعة البيانات |
| Contains | هل يحتوي على هذا الجدول إذا كان نعم فترجع True |
| Count | تعيد عدد الجداول الموجودة في مجموعة البيانات  |
| IndexOf | تبحث عن الجدول المحدد فإذا كان موجودا تعيد رقم ترتيبه في المجموعة |
| Item خاصية | تستقبل اسم أو رقم الجدول للتعبير عنه |
| Remove | تستقبل اسم أو رقم الجدول ثم تحذفه من مجموعة البيانات. |
| RemoveAt | تستقبل رقم الجدول فقط المراد حذفه |

إضافة جدول إلى مجموعة البيانات Add Table To Data Set

يمكنك إضافة جدول إلى مجموعة البيانات عن طريق الوسيلة Add أو مجموعة من الجداول عن طريق الوسيلة AddRange انظر المثال :

Sub AddTable()

 Dim Ds As New DataSet("Ecole") ' إنشاء مجموعة البيانات

 Dim Dt As New DataTable("Resulta") 'إنشاء جدول النتائج

 Dim Dt2 As New DataTable("Students") ' إنشاء جدول الطلاب

 Try

 'Ds.Tables.Add(Dt) ' إضافة جدول واحد

 Ds.Tables.AddRange(New DataTable() {Dt, Dt2}) ' إضافة الجدولين معا

 Catch ex As OleDbException

 MsgBox(ex.Message)

 End Try

 End Sub

التعرف على أسماء الجداول التابعة لمجموعة البيانات

يمكنك معرفة كل الجداول الموجودة في مجموعة البيانات دفعة واحدة فاستعمل الخاصية TableName انظر المثال

For Each tbl As DataTable In Ds.Tables ' مهما كان الجدول من مجموعة البيانات

 Console.WriteLine(tbl.TableName) ' قم بعرضه

 Next

و بكيفية أخرى

For i As Integer = 0 To Ds.Tables.Count – 1 ' المرور عبر كل الجداول

 Console.WriteLine(Ds.Tables(i).TableName) ' عرض الجداول

 Next

كما يمكنك الوصول إلى اسم الجدول باستعمال اسمه أو رقم فهرسته كما يلي :

MsgBox(Ds.Tables(0).ToString) ' Resulta

MsgBox(Ds.Tables("Students").ToString) ' Students

البحث عن جدول في مجموعة البيانات

و إذا أردت أن تعرف هل الجدول موجود فاستعمل الوسيلة Contains فكتب مثلا :

If Ds.Tables.Contains("Students") Then MsgBox("الجدول موجود") ' True

أو استعمل وسيلة البحث IndexOf فترجع لك رقم الجدول إن كان موجود

If Ds.Tables.IndexOf("Students") > 0 Then

 MsgBox("الجدول موجود و رقمه" & Ds.Tables.IndexOf("Students"))

 End If

و للبحث عن جدول استعمل الوسيلة IndexOf التي ترجع لنا الرقم الفهرسي للجدول ضمن مجموعة البيانات انظر المثال

 MsgBox(Ds.Tables.IndexOf("Students"))

حذف جدول من مجموعة البيانات

لحذف جدول نستعمل الوسيلتين Remove و التي تتطلب اسم الجدول البرمجي أو رقمه الفهرسي ضمن مجموعة البيانات . أو الوسيلة RemoveAt التي تستقبل الرقم الفهرسي للجدول

Dim Dt As New DataTable("MyTable")

 Ds.Tables.Remove(Dt) 'حذف الجدول باستعمال المرجع

 Ds.Tables.Remove(0) 'حذف الجدول باستعمال الرقم الفهرسي

 Ds.Tables .RemoveAt (0) ' يجب إدخال رقم الجدول الفهرسي

و إذا رغبت في مسح و حذف كل جداول المجموعة استعمل الوسيلة Clear

 Ds.Tables.Clear()' حذف كل الجداول

 MsgBox(Ds.Tables.Count) 'تأكد من عدد الجداول 0

أهم الوسائل التجميعية للعلاقات التابعة لمجموعة خواص مجموعة البيانات Ds.Relations. …….) (

فهي مماثلة للخصائص التجميعية الخاصة بمجموعة الجدول تمكنك من إضافة و حذف و البحث ... في العلاقات بين الجداول

إنشاء العلاقات و حذفها بين الجداول

 يمكنك إنشاء علاقة بين جدولين في مجموعة البيانات باستعمال الخاصية Relations وهي من خواص التجميع

وهذه الصيغة العامة لإنشاء العلاقات

Ds.Relations.Add("NameRelation",ParentTable.Columns("PrimaryKey") , ChildTable.Columns("ForeignKey"))

وهذا مثال :

Sub AddRelation()

 'نص الاتصال

 Dim Ds As New DataSet("Ecole")

 Dim Dt1 As New DataTable("Students") ' إنشاء جدول الطلاب

 Dim Dt2 As New DataTable("Resulta") 'إنشاء جدول النتائج

 Try

 ' إنشاء عمود الرقم

 Dim ColId As New DataColumn("Id", GetType(Integer))

 Dt1.Columns.Add(ColId) ' إضافة إلى الجدول1

 Dt1.PrimaryKey = New DataColumn() {ColId} ' جعله حقلا أساسيا

 ' إنشاء حقل آخر

 Dt2.Columns.Add("Num", GetType(Integer))

 Ds.Tables.AddRange(New DataTable() {Dt1, Dt2}) ' إضافته إلى الجدول2

 ' إنشاء علاقة بين الجدول 1 و الجدول2

 Ds.Relations.Add("R1", Dt1.Columns("Id"), Dt2.Columns("Num"))

 ' استعراض هذه العلاقة

 For Each rel As DataRelation In Ds.Relations

 Console.WriteLine(rel.RelationName)

 Next

 Catch ex As OleDbException

 MsgBox(ex.Message)

 End Try

 End Sub

حذف محتوى مجموعة البيانات

 لحذف محتوى مجموعة البيانات استعمل الوسيلة Clear انظر المثال : Ds.Clear()

 و لمحوها من الذاكرة استعمل الوسيلة Ds.Clear()

نسخ مجموعة البيانات

 Dim CopyDs As New DataSet 'إنشاء مجموعة بيانات جديدة

 CopyDs = Ds ' نسخ مجموعة البيانات القديمة في الجديدة

 MsgBox(CopyDs.Tables(0).ToString) ' تأكد بعرض الجدول 1

دمج مجموعتي بيانات

**التعامل مع كائن جدول البيانات Data Table**

يُتيح لك هذا الكائن التعامل مع أحد جداول مجموعة البيانات.. ويتمّ تعريفه كالتالي:

Dim DT As DataTable = New DataTable("TableName")

'Or

 Dim DT As New DataTable("TableName")

' Or

 Dim Ds As New DataSet

 Dim DT As New DataTable

 DT = Ds.Tables("TableName")

**أهم خصائص كائن الجدول**

|  |  |
| --- | --- |
| الخاصية | الوظيفة |
| ChildRelations | تعيد مجموعة العلاقات المترابطة مع هذا الجدول في حالة كونه جدول ثانوي  |
| Columns | تعيد مجموعة الأعمدة أو الحقول التي يتكون منها الجدول |
| Constraints | تعيد مجموعة القيود و الشروط الموجودة في هذا الجدول |
| DataSet | تعيد كائن مجموعة البيانات التي ينتمي إليها الجدول |
| DefaultView | تعيد هذه الخاصيّة كائن العرض View الذي يمثّل البيانات التي سيتمّ عرضها للمستخدم من هذا الجدول |
| DisplayExpression | تمثل عنوان الجدول الذي يتم عرضه في واجهة المستخدم  |
| HasErrors | تعيد True إذا كانت أخطاء في صفوف الجدول أو صفوف الجداول التي لها علاقة مع الجدول  |
| ParentRelations | تعيد مجموعة العلاقات المترابطة مع هذا الجدول في حالة كونه الجدول الرئيسي |
| PrimaryKey | تمثل مصفوفة الحقول التي ترغب في جعلها مفاتيح أساسية للجدول |
| Rows | تعيد مجموعة الصفوف (السجلات) الموجودة في هذا الجدول |
| TableName | تعيد اسم الجدول أو تغيير اسمه |

الخاصية TableName ) عرض أسماء الجداول في مجموعة البيانات (

Private Sub GetTableNames(ByVal dataSet As DataSet)

 ' عرض أسماء الجدول.

 Dim table As DataTable

 For Each table In dataSet.Tables

 Console.WriteLine(table.TableName)

 Next table

 End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

 Dim Ds As New DataSet

 Ds.Tables.Add("Table")

 Ds.Tables.Add("Table2")

 GetTableNames(Ds)

 End Sub

 الخاصية Columns ) عرض حقول أو أعمدة جدول (

Dim table As New DataTable

 For Each column As DataColumn In table.Columns

 Console.WriteLine(column.ColumnName)

Next

الخاصية Rows ) عرض صفوف أو سجلات حقل (

Dim table As New DataTable

 For Each Row As DataRow In table.Rows

 Console.WriteLine(Row(0)("ColumName"))

 Next

لقراءة صف نحتاج لرقم الصف و كذا اسم الحقل :

Row(رقم الصف ) (اسم الحقل )

الخاصية DefaultView ) لتصفية البيانات (

Dim table As New DataTable

 table.DefaultView.RowFilter = "Prenom ='وسيم'"

 DGV1.DataSource = table

و سنتطرق لهذه الخاصية بالتفصيل مع كائن العرض View

الخاصية PrimaryKey )إنشاء مفتاح أساسي(

Dim table As New DataTable

 'إنشاء حقل الرقم

 Dim Column As New DataColumn("Id", GetType(Integer))

 table.Columns.Add(Column) ' إضافته إلى الجدول

 'جعل الحقل مفتاحا أساسيا

 table.PrimaryKey = New DataColumn() {Column}

الخاصية HasError )استكشاف الخطأ(

Dim thisDataSet As DataSet = DataSet.GetChanges()

 ' Check each table's HasErrors property.

 Dim table As DataTable

 For Each table In thisDataSet.Tables

 ' If HasErrors is true, reconcile errors.

 If table.HasErrors Then

 ' Insert code to reconcile errors.

 End If

 Next table

أهم و سائل الجدول

|  |  |
| --- | --- |
| الوسيلة | الوظيفة |
| AcceptChanges() | قبول التغيرات في حالة العرض ليس في القاعدة الأصلية  |
| Clear | مسح كل بيانات الجدول |
| Clone() | نسخ بنية الجدول دون البيانات |
| Compute | تعيد عدد السجلات الموافق لشرط محدد |
| Copy | نسخ بنية و بيانات الجدول معا |
| Dispose | محو الجدول من الذاكرة |
| GetChanges | تعيد جدولا جديدا يمثل التغيرات المحدثة و يعرض الصفوف المتأثرة  |
| GetErrors | تعيد مصفوفة تمثل الصفوف التي حدث فيها الخطأ بعد الحفظ |
| ImportRow() | استخدم هذه الوسيلة لتنسخ إلى الجدول، السجلّ DaraRow المرسل لها كمعامل، بكلّ بياناته وخصائصه. |
| Load | لتحميل الجدول بواسطة الكائن قارئ البيانات Data Reader |
| LoadDataRow | أرسل لهذه الوسيلة مصفوفة تحتوي على قيم أحد السجلات.. ستبحث الوسيلة عن الحقل الذي يحتوي على القيمة المناظرة للمفتاح الأساسيّ في المصفوفة، فإذا وجدته، تقوم بنسخ باقي القيم من المصفوفة إلى باقي حقول السجلّ.. وإذا لم تجدْه، تقوم بإنشاء سجلّ جديد وتضع بحقوله قيم المصفوفة.ويوجد معامل ثانٍ لهذه المصفوفة، لو جعلته True فسيتمّ قبول التغييرات التي أدخلت على الحقل (بحيث لا يمكن التراجع عنها بعد ذلك).وتعيد هذه الوسيلة كائن صفّ DataRow يشير للصفّ الذي تمّ تعديله أو إنشاؤه. |
| Merge | تقوم بدمج الجدول المعطى مع الجدول الحالي |
| NewRow | تنشئ صفا جديدا له نفس مواصفات صفوف الجدول . لا تنس قم بإضافته إلى الجدول عن طريق الوسيلة Add التالبعة للجدول |
| RejectChanges | إلغاء التغيرات |
| Reset | إعادة الجدول إلى حالته الافتراضية قبل التعديلات |
| Select | تعيد مصفوفة من الصفوف التي توافق الشرط المحدد |

الوسيلة NewRow )إضافة صف جديد(

Dim table As New DataTable

 'إنشاء كائن الصف يستقبل صفا جديدا

 Dim dRow As DataRow = table.NewRow

 'بيانات الصف الجديد

 dRow(0) = 1

 dRow("Prenom") = "خالد"

 dRow("Sex") = "ذ"

 'إضافة الصف إلى الجدول

 table.Rows.Add(dRow)

الوسيلة Select ) تصفية البيانات(

تعيد مصفوفة تمثل صفوف الجدول الموافقة للشرط المحدد و لها عدة صيغ منها

Dim ArryRows() As DataRow = table.Select("Filter Expression")

--

Filter Expression : صيغة البحث

انظر المثال التالي :

 Dim RowsFilter() As DataRow = table.Select("Prenom ='وسيم'")

 For i As Integer = 0 To UBound(RowsFilter)

 Console.WriteLine(RowsFilter(i)("Id"))

 Next

و هناك صيغة أخرى تتيح لك اختيار نوع الترتيب (تصاعديّ ASC ـ وهو الافتراضيّ ـ أو تنازليّ DESC).

نفس المثال السابق نضيف نوع الترتيب الأبجدي تنازلي لحقل الرقم Id

Dim RowsFilter() As DataRow = table.Select("Prenom ='وسيم'", "Id DESC")

الوسيلة Compute )حساب عدد الصفوف وفق الشرط المحدد (

Dim Value As Object = table.Compute(expression, Filter)

Expression : تمثل عملية حسابية و تشمل دوال التجميع

و هي Count و Avg و Min و Max

Filter : صيغة البحث أو التصفية أو الشرط

فهو يماثل مقطع WHERE في جملة SQL

Dim Table As New DataTable

 'حساب عدد المتمدرسين في الصف الخامس

 Dim Values As Integer = Table.Compute("Count(Id)", "Classe=5")

 'أعلى معدل في قسم السنة الخامسة

 Dim I As Double = Table.Compute("Max(Moyen)", "Classe =5")

و لكن لا يمكنك إجراء عمليات حسابية على أكثر من حقل بهذه الوسيلة كأن تحسب مجموع مواد العربية و الرياضيات هذا غير ممكن :

 Dim I As Integer = Table.Compute("Arabe + Math", "")

فما هو الحل ؟

سنتعرف على الحل مع درس كائن العمود و الخاصية Expression .

الخاصية Load ) تحميل الجدول بواسطة قارئ البيانات(

Private Sub LoadDataGridView()

 'نص الاتصال

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source = E:/db1.mdb"

 'إنشاء كائن الاتصال

 Dim Con As New OleDbConnection(S)

 Try

 Con.Open()' فتح الاتصال

 'إنشاء كائن الأمر لتحديد حقول الجدول

 Dim Cmd As New OleDbCommand("Select \* From Students", Con)

 'تنفيذ الأمر عن طريق قارئ البيانات

 Dim Dr As OleDbDataReader = Cmd.ExecuteReader

 'إنشاء جدول

 Dim Dt As New DataTable

 'تحميل الجدول بعد تنفيذ الأمر بواسطة قارئ البيانات

 Dt.Load(Dr)

 'عرض البيانات في جدول

 dgv1.DataSource = Dt

 'غلق قارئ البيانات

 Dr.Close()

 'غلق الاتصال

 Con.Close()

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

الخاصية **Copy نسح جدول ببنيته و بياناته**

 Dim copyDataTable As DataTable

 copyDataTable = Table.Copy()

**التعامل مع كائن العمود Data Column**

يمكنك من تعريف عمود و إضافته إلى الجدول

إنشاء كائن العمود :

Dim Table As New DataTable 'إنشاء جدول

Dim Column As New DataColumn 'إنشاء كائن عمود جديد

و من المعلوم أن تعريف العمود يحتاج إلى اسم و نوع البيانات و مواصفات أخرى سنتعرف عليها لاحقا.

أهم خصائص كائن العمود

|  |  |
| --- | --- |
| الخاصية | الوظيفة |
| AllowDBNull | السماح بقيم فارغة أم لا |
| AutoIncrement | هل الحقل يقبل ترقيما تلقائيا أم لا |
| AutoIncrementSeed | تحديد قيمة بداية الترقيم |
| AutoIncrementStep | تحديد مقدار الزيادة في عمود الترقيم |
| Caption | العنوان الحقل الذي يظهر للمستخدم |
| ColumnName | اسم العمود أو الحقل |
| DataType | نوع بيانات العمود |
| DefaultValue | القيمة الافتراضية للحقل |
| Expression | الصيغة المرافقة للعمود |
| MaxLength | أقصى / طول للحقل النصي  |
| Ordinal | تحديد ترتيب العمود ضمن أعمدة الجدول |
| ReadOnly | جعل بيانات العمود للقراءة فقط فلا يمكن تغييرها |
| Table | تعيد الجدول الذي ينتمي إليه الحقل |
| Unique | لا يمكن تكرار البيانات / فريد  |

و الآن تعال نتناول أمثلة لتوظيف هذه الخصائص.

Dim Table As New DataTable("Students") 'إنشاء جدول

 Dim Column As New DataColumn ' إنشاء كائن العمود

 Column.ColumnName = "Id" ' اسم الحقل

 Column.DataType = GetType(Integer) ' نوع بيانات الحقل

 Column.AllowDBNull = False 'لا يقبل قيما فارغا

 Column.AutoIncrement = True ' حقل ترقيم تلقائي

 Column.AutoIncrementSeed = 1 ' قيمة البداية 1

 Column.Caption = "الرقم" ' يظهر للمستخدم

 Column.Unique = True 'لا يقبل التكرار

 Table.Columns.Add(Column) 'إضافة الحقل إلى الجدول

 '------------------------------------------------

 Dim ColSex As New DataColumn("Sex", GetType(String))

 ColSex.DefaultValue = "M" 'القيمة الافتراضية

 ColSex.MaxLength = 2 ' اقصى طول

 ColSex.AllowDBNull = False ' مطلوب لا يقبل قيما فارغا

 ColSex.Unique = True ' يقبل التكرار

 Table.Columns.Add(ColSex) ' إضافة الحقل إلى الجدول

إضافة حقلين معا إلى جدول AddRange

 Dim Dt As New DataTable("Students") ' إنشاء جدول

 Dim ColId As New DataColumn("Id", GetType(Integer)) 'إنشاء عمود الرقم

 Dim ColPrenom As New DataColumn("Prenom", GetType(String)) ' إنشاء عمود الاسم

 Dt.Columns.AddRange(New DataColumn() {ColId, ColPrenom})'إضافة الحقلين إلى الجدول

معرفة عدد الأعمدة

MsgBox(Dt.Columns.Count) **'** عدد الأعمدة

هل العمود موجود

MsgBox(Dt.Columns.Contains("Prenom")) 'True

البحث عن عمود IndexOf

تعيد رقم الفهرسي للعمود في الجدول . أما إذا كان غير موجود ترجع -1

MsgBox(Dt.Columns.IndexOf("Prenom")) '1

حذف عمود

يمكنك استعمال إحدى الطرق التالية

Dt.Columns.Remove("Prenom") ' حذف باستعمال اسم الحقل

 Dt.Columns.Remove(1) ' حذف باستعمال الرقم الفهرسي

 Dt.Columns.RemoveAt(1) ' الحذف باستعمال الرقم

و لحذف كل الأعمدة نستعمل الوسيلة Dt.Clear

**القيود و الشروط Constraint**

لإضافة قيد إلى جدول نستعمل الخاصية Add التابعة لخاصية Constrain الخاصة بجدول البيانات انظر المثال

إنشاء قيد لا يقبل التكرار Unique

 Dim Cunq As New UniqueConstraint(Dt.Columns("Id")) 'إنشاء قيد وتعيين العمود

 Dt.Constraints.Add(Cunq)

كما يمكنك إضافة القيد عند إنشاء العمود كما رأينا سابقا.

و بنفس الطريقة يمكنك إضافة باقي القيود كقيد المفتاح الثانوي و لا يقبل قيما فارغة ...

و لحذف القيد نكتب : Dt.Constraints.Remove(Cunq)

إنشاء قيد مفتاح أساسي

Dim Dt As New DataTable("Students")

 Dim ColId As New DataColumn("Id", GetType(Integer))

 Dim ColPrenom As New DataColumn("Prenom", GetType(String))

 Dt.Columns.AddRange(New DataColumn() {ColId, ColPrenom})

 Dt.PrimaryKey = New DataColumn() {ColId} ' جعل حقل الرقم مفتاحا أساسيا

عرض كل حقول الجدول

For Each col As DataColumn In Dt.Columns

 Console.WriteLine("{0} ,{1}", col.ColumnName, col.DataType)

 Next

For i As Integer = 0 To Dt.Columns.Count - 1

 Console.WriteLine("{0},{1}", Dt.Columns(i).ColumnName, Dt.Columns(i).DataType)

 Next

صيغة العمود Expression

تستعمل للقيام بالعمليات الحسابية بين الحقول . و كذا كشرط اختبار أو تصفية لعرض البيانات الموافقة للشرط كمعامل Where في جمل الاستعلامات.

1. استعماله في العمليات الحسابية

 Dim Table As New DataTable("Score") 'إنشاء جدول

 'إنشاء الحقول

 Dim ColId As New DataColumn("Id", GetType(Integer))

 Dim ColArabe As New DataColumn("Arabe", GetType(Double))

 Dim ColMath As New DataColumn("Math", GetType(Double))

 Dim ColFrance As New DataColumn("France", GetType(Double))

 Dim ColMoyen As New DataColumn("Moyen", GetType(Double))

 'إنشاء تعبير لحساب المعدل

 ColMoyen.Expression = "((Arabe + Math)\*2 + France)/5"

 'إضافة الحقول إلى الجدول

 Table.Columns.AddRange(New DataColumn() {ColId, ColArabe, ColMath, ColFrance, ColMoyen})

2 استعماله كشرط :

ColMoyen.Expression = "Moyen > 10"

ColPrenom .Expression = "Prenom = 'وسيم'"

و هو يستعمل كمعامل Where في جمل الاستعلامات المهيكلة و سنتطرق لهذه الصيغ لاحقا مع جملة Where.

**التعامل مع كائن الصف Data Row**

يمثل أحد صفوف الجدول

**أهم خصائص كائن الصف**

|  |  |
| --- | --- |
| الوظيفة | الخاصية |
| تستقبل اسم العمود الذي تريد قراءة صف أو صفوف منه | Item |
| تستقبل مصفوفة تمثل بيانات الصف المراد إضافته أو قراءته  | ItemArray |
| تعيد حالة الصف كونه مضافا أو محذوفا أو معدلا | RowState |
| تعيد كائن الجدول الذي ينتمي إليه الصف  | Table |

 الخاصية Item ) تحديد اسم العمود المراد عرض صف أو صفوف منه

وهذا بتحديد رقم الصف و اسم الحقل الذي تود عرض صف منه

 DTable.Rows(RowIndex).Item("ColumName")

 قراء صف واحد

TextBox1.Text = DTable(0).Item("Prenom")

و يمكن كتابة هذه التعليمات اختصارا بالاستغناء عن الخاصية Item

TextBox1.Text = DTable(0)("Prenom")

و تعني عرض الصف الأول ورقمه 0 من حقل الاسم

كما يمكنك كتابة رقم فهرسة العمود في الجدول هكذا

TextBox1.Text = DTable(0)(1)

و في كل الحلات نحصل على نفس النتيجة و هي الصف الأول من حقل الاسم

عرض مصفوفة من الصفوف أي محتوى كل العمود من الصفوف

أ – عن طريق كائن الجدول Data Table:

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source =E:/Db1.mdb"

 Dim Con As New OleDbConnection(S)

 Dim DTable As New DataTable

 Try

 Con.Open()

 Dim Da As New OleDbDataAdapter("Select \* From Students", Con)

 Da.Fill(DTable)

 For i As Integer = 0 To DTable.Rows.Count - 1

 LstName.Items.Add(DTable.Rows(i)("Prenom"))

 Next

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 و هناك طريقة أخرى و ذلك باستبدال الكود المظلل بالتعليمات التالية

For Each RowData As DataRow In DTable.Rows

 LstName.Items.Add(RowData("Prenom"))

 Next

إضافة صف جديد :

Dim DTable As DataTable 'إنشاء كائن الجدول

'تعريف كائن صف جديد

 Dim drow As DataRow = DTable.NewRow

'تحميل كائن الصف بالبيانات

 drow("Id") = txtId.text

 drow("Prenom") = txtPrenom.text

 drow(3) = 5

'إضافة كائن الصف الجديد إلى الجدول

 DTable.Rows.Add(drow)

 الخاصية ItemArray لعرض بيانات صف من الجدول

 Dim S As String = "Provider = Microsoft.jet.oledb.4.0;" & \_

 "data source =E:/Db1.mdb"

 Dim Con As New OleDbConnection(S)

 Dim DTable As New DataTable

 Try

 Con.Open()

 Dim Da As New OleDbDataAdapter("Select \* From Students", Con)

 Da.Fill(DTable)

 Dim Values() As Object = {111, "Ali", "12/10/2010"}

 Dim Drow As DataRow = DTable.NewRow

 Drow.ItemArray = Values

 DTable.Rows.Add(Drow)

 Dim CMB As New OleDbCommandBuilder(Da)

 Da.Update(DTable)

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

أهم وسائل كائن الصف

|  |  |
| --- | --- |
| **الوسيلة** | **الوظيفة** |
| Delete | حذف صف من الجدول |
| IsNull | تعيد True في حالة ما إذا كانت الخانة المحدد فارغة |

الخاصية Delete )حذف صف (

DTable.Rows(RowIndex).delete() **'**

**مثال :**

If MsgBox("هل ترغب حقا في حذف السجل المحدد", MsgBoxStyle.Question + MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then

 DTable.Rows(7).Delete()

End If

الخاصية IsNull

في حالة ما إذا كان الحقل غير فارغ أطبع محتوى الحقل

If Not row.IsNull(column) Then

 Console.WriteLine(row(column).ToString())

End If

مثال عن إنشاء مجموعة البيانات من جديد

Public Class Form1

 Dim Ds As New DataSet("Ecole") ' إنشاء مجموعة بيانات

 Dim DtStudName As New DataTable("Stud\_Name") ' إنشاء جدول أسماء الطلاب

 Dim DtStudResulta As New DataTable("Stud\_Resulta")' إنشاء جدول نتائج الطلاب

 Private Sub Form1\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

' إضافة الجدول مع إلى مجموعة البيانات

 Ds.Tables.AddRange(New DataTable() {DtStudName, DtStudResulta})

 ' عرض الجدولين

 For Each tbl As DataTable In Ds.Tables

 Console.WriteLine(tbl.TableName)

 Next

 'إنشاء عمود الرقم

 Dim ColId As New DataColumn

 ColId.ColumnName = "Id" ' اسمه الرقم

 ColId.DataType = GetType(Integer) ' نوع بياناته

 ColId.AutoIncrement = True ' هل هو ترقيم تلقائي

 ColId.AutoIncrementSeed = 1 ' يبدأ من

 ColId.AutoIncrementStep = 1 ' مقدار الزيادة

 ColId.AllowDBNull = False ' لا يقبل قيما فارغة

 ColId.Unique = False ' لا يقبل التكرار

 DtStudName.Columns.Add(ColId) ' إضافة حقل الرقم

 DtStudName.Columns.Add("Prenom", GetType(String)) 'إضافة حقل الاسم

 DtStudName.Columns.Add("DBirth", GetType(Date)) ' إضافة حقل التاريخ الميلاد

 ' عرض حقول جدول أسماء الطلاب و نوع بياناتها

 For Each col As DataColumn In DtStudName.Columns

 Console.WriteLine("{0} - {1}", col.ColumnName, col.DataType)

 Next

 '

 DtStudResulta.Columns.Add("Id", GetType(Integer)) ' إضافة حقل الرقم لجدول النتائج

 DtStudResulta.Columns.Add("Arabe", GetType(Double)) ' إضافة حقل العربية

 DtStudResulta.Columns.Add("Math", GetType(Double)) ' إضافة حقل الرياضيات

 DtStudResulta.Columns.Add("France", GetType(Double)) ' إضافة حقل الفرنسية

 DtStudResulta.Columns.Add("Moyen", GetType(Double), "((Arabe+Math)\*2 + France)/5") ' إضافة حقل مخصص يحسب معدل المواد

 ' عرض حقول جدول النتائج ببياناتها

 For Each col As DataColumn In DtStudResulta.Columns

 Console.WriteLine("{0} - {1}", col.ColumnName, col.DataType)

 Next

 ' جعل حقل الرقم في جدول أسماء الطلاب كحقل أساسي

 DtStudName.PrimaryKey = New DataColumn() {ColId}

 ' إضافة علاقة بين الجدولين

 Ds.Relations.Add("R1", DtStudName.Columns("id"), DtStudResulta.Columns("id"))

 ' عرض العلاقات بين الجداول

 For Each rel As DataRelation In Ds.Relations

 Console.WriteLine(rel.RelationName)

 Next

 'إضافة صف جديد لجدول الأسماء

 Dim Drow As DataRow = DtStudName.NewRow

 Drow(1) = "Salah"

 Drow(2) = "10-05-1970"

 DtStudName.Rows.Add(Drow)

 ' عرض صفوف جدول الأسماء

 For Each row As DataRow In DtStudName.Rows

 Console.WriteLine("{0} - {1} - {2}", row(0), row(1), row(2))

 Next

 ' إضافة صف جديد لجدول النتائج

 Dim Drow1 As DataRow = DtStudResulta.NewRow

 Drow1(0) = 1

 Drow1(1) = 14.5

 Drow1(2) = 12.5

 Drow1(3) = 10.5

 DtStudResulta.Rows.Add(Drow1) ' إضافة الصف إلى الجدول

 ' عرض صفوف جدول النتائج

 For Each row As DataRow In DtStudResulta.Rows

 Console.WriteLine("{0} - {1} - {2} - {3} , {4}", row(0), row(1), row(2), row(3), row("Moyen")) ' لا حظ أنه تم حساب المعدل تلقائيا

 Next

 End Sub

End Class