



نظم ادارة قواعد بيانات

ADO.NET

الفصل الرابع حاسوب

منهج مادة نظم ادارة قواعد بيانات يهتم بربط VB.NET مع قاعدة بيانات SQL Server عن طريق تعيين

ADO.Net

أنظمة قواعد بيانات تقنية

ADO.NET

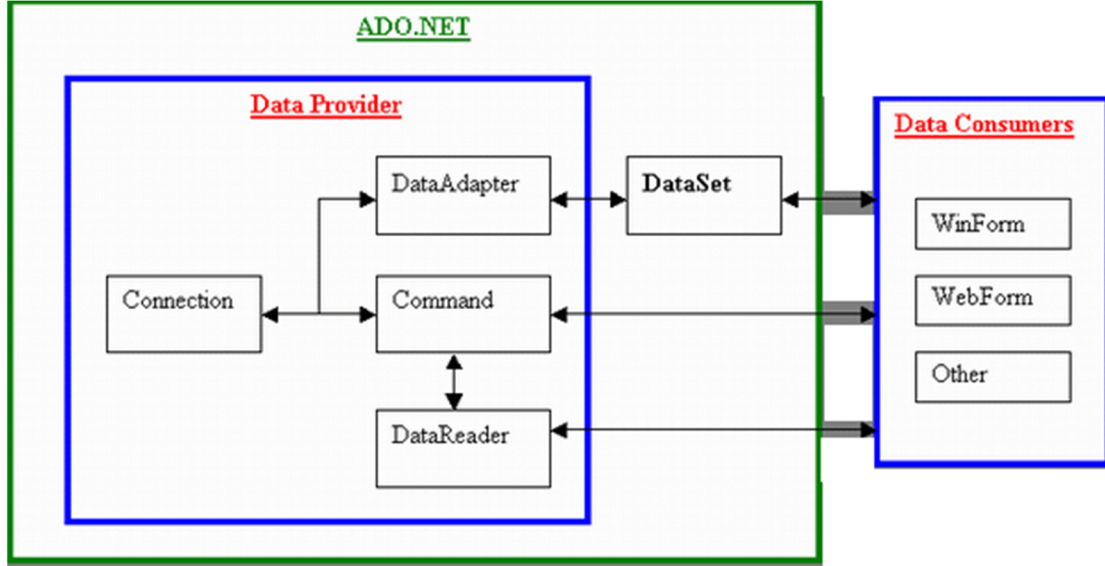
نبذة عن ADO.NET

هي مجموعة من الفئات مشمولة في مجال الأسماء **System.Data** غرضها الوصول إلى مصادر البيانات **Data Sources** والتي تمثل بيانات محفوظة تحت أنظمة قواعد بيانات متعددة الأنواع مما يعني قدرتك على الوصول إلى أي قاعدة بيانات مهما كانت الشركة المنتجة لها .

مميزات ADO.NET

- 1-مصممة من الأساس للعمل في بيئة غير متصلة .
- 2-يستخدم الكائن **DataSet** للاحتفاظ بعدة مجموعات من البيانات .
- 3-لا تستخدم المؤشرات لأنها تعمل في بيئة غير متصلة.
- 4-تخزن البيانات في هيئة **XML** العالمية . وهذه الهيئة مصممة لكي ترسل عبر جدران الحماية وعبر الشبكات دون مشاكل كما يمكن لأي تطبيق قراءة البيانات بهيئة **XML** بسهولة
- 5-تعمل كنظام بيانات منفصل عن قاعدة البيانات فهي لا تتصل بقاعدة البيانات إلا عند الضرورة وبالتالي لن تستهلك مورد النظام إلا عند الضرورة .

والمقصود هنا الخصائص المزودة والمساعدة في عملية الاتصال لقراءة وتكليف البيانات وهي



[1]- كائن مجموعة البيانات DataSet

وهو الكائن وبه الكثير من المزايا والتحسينات حيث يستطيع تخزين أكثر من جدول أو نتيجة استعلام في نفس الوقت حيث يمثل كل واحد من هذه الجداول كائناً منفصلاً عن الآخر.

[2]- كائن محول البيانات DataAdapter

يمثل الجسر الذي يربط بين DataSet وقاعدة البيانات ويدعم أوامر - Update - Select - Delete - Insert وبالتالي بإمكانه القيام بعمليات مختلفة على البيانات كما أنه المسؤول عن تحميل كائن DataSet بالبيانات.

[3]- كائن قارئ البيانات DataReader

يستخدم هذا الكائن لقراءة البيانات فقط ويمكنه قراءة كميات ضخمة منها تلك التي لا يمكن تخزينها في الذاكرة مؤقتاً .

[4] - كائن علاقات البيانات DataRelation

يستخدم هذا الكائن لتمثيل العلاقات بين الجداول في قاعدة البيانات JOIN .

[5] - كائن الاتصال Connection

يعمل هذا الكائن بصورة مشابهة للكائن **Connection** في **ADO** وهو يمكننا من إنشاء اتصال مع قاعدة البيانات .

[6]-كائن الامر **Command**

يسمح هذا الكائن لكائن **DataAdapter** بتطبيق الأوامر على قاعدة البيانات ويمكنه أن يتضمن اربعة من هذه الأوامر.

أهدفنا في التعامل مع قواعد البيانات:

- 1-الاتصال بمصدر قاعدة البيانات.
- 2-قراءة البيانات التي بها وعرضها بطرق شتى من خلال الأدوات المتوفرة في بيئة .NET Microsoft.
- 3-معالجة البيانات وتحديثها من خلال تعديل قيم هذه البيانات أو إضافة بيانات جديدة أو حذف بيانات غير مرغوب فيها.
- 4-حفظ البيانات المحدثة.
- 5-الإبحار في البيانات من خلال الانتقال للسجل التالي أو العودة للسجل السابق أو الانتقال للسجل الأول أو الانتقال للسجل الأخير.
- 6-البحث عن البيانات بسهولة ويسر من خلال وضع فلاتر للترشيح.
- 7-ربط البيانات ذات العلاقة في أكثر من جدول وعرضها ومعالجتها.
- 8-طباعة التقارير الملخصة للبيانات.

الاتصال بقواعد البيانات :

اولا الوضع المتصل connection mode

1. عند بدأ مشروع جديد للاتصال بقاعدة البيانات يجب قبل كل شيء أن نستدعي فضاء الأسماء المناسب لمزود البيانات (Provider) الذي نتعامل معه . والذي تدرج تحته كائناته الخاصة به للاتصال بقاعدة البيانات و معالجتها . ومن مميزات استدعاء فضاء الأسماء أنه يغنينا عن كتابة مسار الكائنات التي سوف نستخدمها . فقط نكتب اسم الكائن.

وعند التعامل مع قواعد بيانات SQL Server مثلا . نستدعي فضاء الأسماء System.Data.SqlClient و ذلك بكتابة الكود التالي في أعلى صفحة الكود:

```
Imports System.Data.SqlClient
```

وتدرج تحته مجموعة من الكائنات ولكن ما يهمنا منها ما يلي:

SqlCommand , SqlConnection , SqlDataAdapter , SqlDataReader ,
SqlParameter , SqlCommandBuilder , SqlTransaction. SqlException .

2. انشاء كائن الاتصال Connection

حيث SqlConnection مخصصة للاتصال بقواعد البيانات من نوع .NET SQL Server Data Provider فسيكون تعريف الاتصال بهذا الشكل:

```
Dim Con As New SqlConnection()
```

3. نص الاتصال

بعد انشاء كائن الاتصال يبقى عليك ارفاق نص الاتصال (Connection String) به . وهو قيمه نصيه تحوي كل شيء يتعلق بعملية الاتصال ابتداءً بمزود البيانات واسم الخادم وقاعدة البيانات و كلمة المرور . ونص اتصال بقواعد بيانات من نوع SQL Server بهذا الشكل:

```
"server= اسم الخادم;Database=اسم القاعدة;Integrated Security=الحماية"
```

ويسند نص الاتصال الي كائن الاتصال كلاتي :

```
Dim Con As New SqlConnection ("server=.;Database=st_4;Integrated Security=True")
```

4. فتح و اغلاق الاتصال :

بعد اسناد نص الاتصال الي كائن الاتصال يمكن فتح الاتصال عن طريق جملة الأتية

con.open

فتح الاتصال يمكن المستخدم من تنفيذ جمل SQL علي جداول وبيانات القاعدة لان المستخدم بهذه الطريقة فتح قناة تبادل بيانات بين بيئة تطوير VB ونظام ادارة قواعد البيانات SQL Server , و من الضروري جدا غلق الاتصال بالقاعدة اذا لم تكن بحاجة للاتصال بها حتى لانشغل حيز كبير من الذاكره العشوائيه , و ذلك بكتابة الجملة

con.close

تستطيع معرفة حالة الاتصال هل هي مفتوحة مغلقة ... عن طريق الخاصية state متلا تريد ان تعرف هل حالة الاتصال متصلة ام مغلقة و لفعل هذا اكتب

```
If con.State = ConnectionState.Open Then
```

```
    MsgBox ("الاتصال متصل")
```

```
Else
```

MsgBox ("الاتصال مغلق")

End If

5. كائن الامر Command

كائن الأوامر SqlCommand في اعتقادي أنه يمثل القلب للاتصال المتصل . وهو من أجل تنفيذ عملية قراءة البيانات المخزنة في قاعدة البيانات . أو تحديثها (من إضافة وتعديل وحذف) . ويعتمد في تكوينه على وجود اتصال مفتوح مع قاعدة البيانات . وعلى جملة الاستعلام البنيوية SQL وهي لغة قياسية لبناء ومعالجة قواعد البيانات ومحتوياتها . وتتضمن هذه اللغة تعليمات من أجل إضافة . تعديل . حذف . ترتيب . و اختيار سجلات) .

Dim Cmd As New SqlCommand

6. انواع كائن الامر

وله ثلاثة انواع

- 1) Text وهو النوع الافتراضي لتنفيذ جملة استعلام بنيوية SQL .
- 2) StoredProcedure لتنفيذ إجراء مخزن في قاعدة البيانات .
- 3) TableDairct للتعامل مع جدول واحد.

يتم تحديد النوع عن طريق الجملة الآتية

Cmd.CommandType = CommandType.Text

7. تنفيذ كائن الامر

بعد تجهيز نوع كائن الأمر ونريد تنفيذ الأعمال المطلوبة منه عند حصوله على تيار من البيانات . لدينا ثلاثة طرق يتم بها العمل:

الطريقة الأولى (ExecuteReader) وهي أحد خصائص كائن الأمر وتقوم بتنفيذ استعلام بنيوي يرجع لنا مجموعة من الصفوف الموجودة في قاعدة البيانات وهو ما يتطلب استرجاع هذه النتيجة بواسطة الكائن. DataReader.

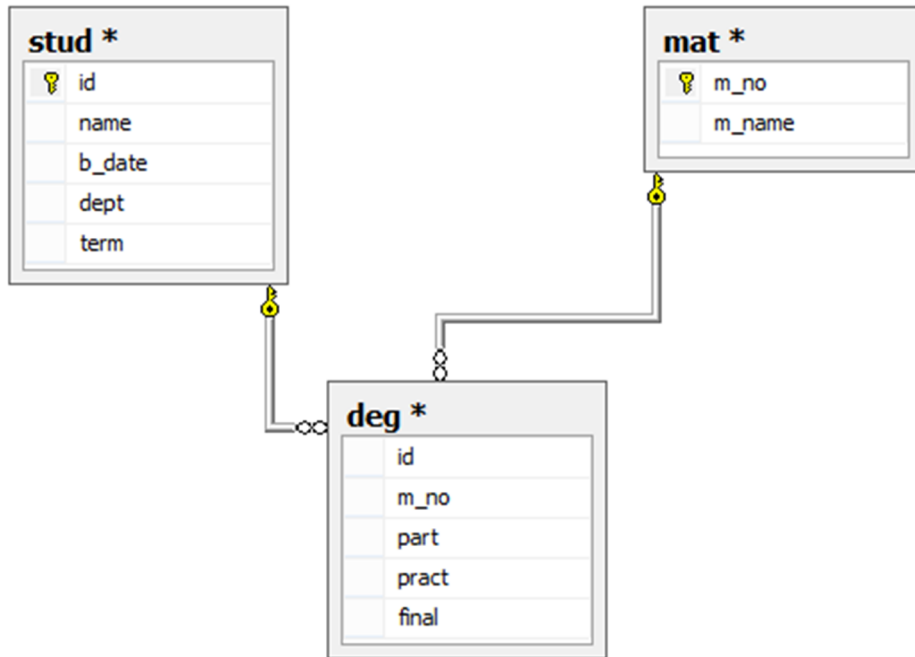
الطريقة الثانية (ExecuteNonQuery) وهي أحد خصائص كائن الأمر وتقوم بتنفيذ استعلام بنيوي يقوم بتحديث قاعدة البيانات (من إضافة . تعديل . و حذف) أو تكوين

قاعدة بيانات والنتيجة التي ترجع هي عدد الصفوف المتأثرة بالأمر.

الطريقة الثالثة (ExecuteScalar) وهي أحد خصائص كائن الأمر وتقوم بتنفيذ استعلام بنيوي يرجع لنا قيمة واحدة فقط من قاعدة البيانات.

مشروع تطبيقي عن الوضع المتصل

1. نكون قاعد بيانات باسم ST_4 علي نظام ادارة قواعد بيانات MS SQL Server بها الجداول الآتية :



2. نكون نموذج علي VB.net الغرض منه ربط النموذج بجدول في قاعدة البيانات واجراء عمليات تحديث قواعد البيانات من خلاله . النموذج كلاتي :

المواد

رقم المادة 111 بحث

اسم المادة برمجه 2

اضافه تعديل حذف

عدد المواد بالمعهد | 2 ماده عرض

3. تحديث قاعدة البيانات (ادخال , حذف , تعديل , قراءة)

أ- ادخال البيانات الي جدول المواد |

في حدث عند النقر على زر امر " اضافة " نكتب

```
Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
Try
con.Open()
Dim cmd as New SqlCommand(" insert into mat values (
'" & TextBox1.Text & "',''" & TextBox2.Text & "' ) ",
con)
cmd.ExecuteNonQuery()
MsgBox("...! تمت الإضافة بنجاح ")
con.Close()
Catch
MsgBox(" انتبه... المدخلات!")
End Try
```

ب- حذف البيانات من جدول المواد |

في حدث عند النقر على زر امر " حذف " نكتب

```
Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
Try
con.Open()
Dim cmd as New SqlCommand(" delete from mat where
m_no =" & TextBox1.Text, con)
cmd.ExecuteNonQuery()
MsgBox("...! تم الحذف ")
con.Close()
Catch
MsgBox(" انتبه... المدخلات!")
End Try
```

ت- تعديل البيانات من جدول المواد |

في حدث عند النقر على زر امر " تعديل " نكتب


```

Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
Try
con.Open()
Dim cmd as New SqlCommand(" update mat set m_name =
'" & TextBox2.Text & "' where m_no =" &
TextBox1.Text, con)
cmd.ExecuteNonQuery()
MsgBox("تم التعديل...!")
con.Close()
Catch
MsgBox("المدخلات...! انتبه")
End Try

```

ث- البحث عن البيانات في جدول المواد |

في حدث عند النقر على زر امر " بحث " نكتب

```

Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
Dim dr As SqlDataReader
con.Open()
Dim cmd as New SqlCommand("select * from mat where
m_no=" & TextBox1.Text, con)
dr = cmd.ExecuteReader
If dr.Read Then
TextBox2.Text = dr!m_name
End If
con.Close()

```

ج- استرجاع قيمه واحده في جدول المواد |

في حدث عند تحميل النموذج نكتب

```

Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
con.Open()
Dim cmd As New SqlCommand("select count(*) from mat",
con)
Label3.Text = " | بالمعهد المواد عدد" & cmd.ExecuteScalar
& " ماده"
con.Close()

```

ح- استعراض سجلات جدول المواد |

نضيف panel1 الي النموذج السابق بها DG لعرض جميع السجلات بالجدول :

رقم المادة	اسم المادة
111	برمجة 4
112	برمجة 2

في حدث عند النقر على الرابط "عرض" نكتب

```

dg.Rows.Clear()
Dim i As Byte
Dim con As New SqlConnection("server=. ;
database=st_4 ; integrated security=true")
Dim dr As SqlDataReader
con.Open()
Dim cmd As New SqlCommand("select * from mat ", con)
dr = cmd.ExecuteReader
While dr.Read
    dg.Rows.Add()
    dg.Item(0, i).Value = dr!m_no
    dg.Item(1, i).Value = dr!m_name
    i += 1
End While
con.Close()

```

Try...Catch statement

يوفر وسيلة للتعامل مع بعض أو كافة الأخطاء المحتملة التي قد تحدث في كتلة معينة من التعليمات البرمجية أثناء عمل البرنامج . الصيغة البرمجية كالتالي :

```

Try
    [ tryStatements ]
Catch [ exception [ As type ] ] [ When expression ]
    [ catchStatements ]
Finally
    [ finallyStatements ]
End Try

```

إذا كنت تتوقع أن خطأ معين قد يحدث خلال مقطع (إجراء) معين من التعليمات البرمجية، ضع التعليمات البرمجية في كتلة try واستخدام كتلة catch للاحتفاظ بالسيطرة ومعالجة الأخطاء في حال حدوثها.

ثانياً الوضع المنفصل connectionless mode

وتقوم المكتبة من خلال أحد الكائنات التي تتكون منها وهو الكائن (Data Set) من إنشاء قاعدة بيانات مشابهة لقاعدة البيانات الموجودة على الـ (Server) أو الجهاز حسب نوع قاعدة البيانات ويتم هذا نسخها في الذاكرة الخاصة بالجهاز ثم إجراء جميع التعديلات في النسخة الوهمية التي بالذاكرة ثم عندما تصدر أمر التحديث يقوم الجهاز بعمل التحديثات التي تمت. كما ان هذا النظام يفضل عدم التعامل معه الا في حالة التقارير أو اي بيانات لا تريد إجراء التعديلات عليها وخاصة في حالة وجود عدة اجهزه تعمل على نفس قاعدة البيانات.

في هذا الجزء سنرى كيف يعمل للاتصال المنفصل وقد قسمته إلى قسمين ، DataSet رئيسيين الاول يعرض لك الفئة DataAdapter والثاني، لمعالجة البيانات التي تمثل الجسر الذي سترسل به بياناتك بعد معالجتها.

1. عند بدأ مشروع جديد للاتصال بقاعدة البيانات يجب قبل كل شيء أن نستدعي فضاء الأسماء المناسب لمزود البيانات من نوع SQL Server . يستدعي كلاتي:

Imports System.Data.SqlClient

2. ننشئ كائن الاتصال Connection و نسند له نص الاتصال كلاتي :

```
Dim Con As New SqlConnection ("server=.;Database=st_4;Integrated Security=True")
```

3. إنشاء محول البيانات Data Adapter

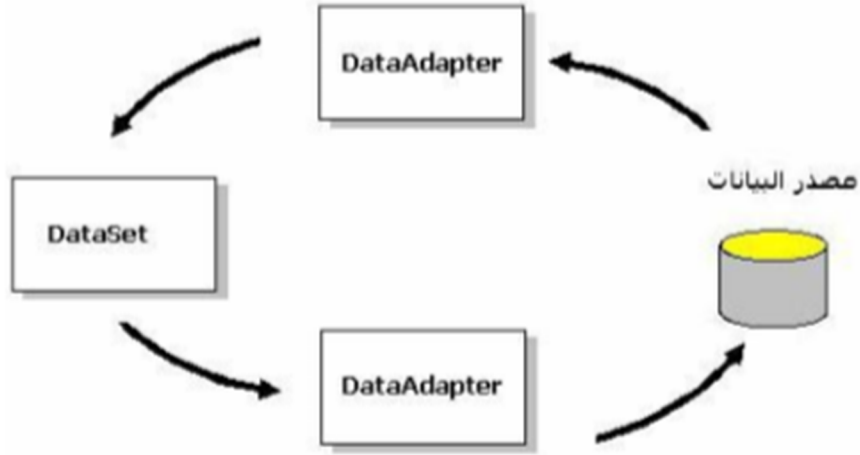
يقوم هذا الكائن باستقبال وارسال البيانات من وإلى مصدر البيانات وإرسالها إلى الكائن (Data Set) ويعتبر هذا الكائن حلقة وصل بين الكائن (Connection) وبين الكائن (Data Set) و بسبب وظيفته هذه يعتبر كائنا معقدا حيث يحتوي على أربع كائنات أخرى تساعد في أداء مهمته هي :

1- (Select Command) : للقراءة من قاعدة البيانات ووضع البيانات التي يتم قراءتها في الكائن (Data Set) .

2- (Update Command) : لإرسال التعديلات إلى مصدر البيانات .

3- (Insert Command) : لإضافة بيانات جديدة إلى مصدر البيانات .

4- (Delete Command) : لحذف بيانات قائمة في مصدر البيانات .



حيث محول البيانات Data Adapter المخصص لتنفيذ جمل SQL علي قواعد البيانات من نوع .NET SQL Server . فسيكون تعريفه بهذا الشكل:

Dim ad As New SqlDataAdapter(sql , con)

يشبه في عمله كائن الامر في الوضع المتصل الا انه يختلف عنه في انه لا يحتاج الي فتح الاتصال . و يمرر له قيمتان : جملة SQL وكائن اتصال .

4. كائن مجموعة البيانات Data Set

يعتبر هذا الكائن قوة وعصب مكتبة الـ (ADO.Net) حيث يمكن لهذا الكائن أخذ نسخة من قاعدة البيانات التي نريد الاتصال بها ثم إجراء جميع العمليات الخاصة بالبيانات على هذه النسخة ثم رفعها لحفظ التعديلات على مصدر البيانات الرئيسي .. وهذا كله يتم في أحد الموارد الرئيسية لنظام التشغيل ألا وهو (الذاكرة) .. ليس هذا فحسب وإنما يمكننا أخذ البيانات من أكثر من قاعدة بيانات للتعامل معها ثم حفظ جميع التعديلات على مصادر البيانات الرئيسية ... و من ذلك نجد أن الفكرة الرئيسية التي يقوم عليها هذا الكائن هي عدم الاتصال بقاعدة البيانات أثناء التعامل معها . و يتم التصريح عنها بهذا الشكل :

Dim ds As New DataSet

5. الكائن CommandBuilder يقوم ببناء الأوامر الخاصة بتحديث البيانات من تعديل وإضافة وحذف . والتي يحتاجها الكائن SqlDataAdapter في عملية التحديث.

Dim cm As New SqlCommandBuilder(ad)

وخلال ذلك التصريح نقوم بربط كائن CommandBuilder بالكائن DataAdapter . ثم نقوم بإيقاف التحديث للبيانات الموجودة في الـ DataSet وقبول التحديثات الحالية من أجل حفظها ونقلها من الـ DataSet إلى مصدر البيانات .

مشروع تطبيقي عن الوضع المنفصل

1. نعمل على جدول الطالب Stud في قاعد بيانات ST_4 .
2. نكون نموذج علي VB.net الغرض منه ربط النموذج بجدول في قاعدة البيانات واجراء عمليات تحديث قواعد البيانات من خلاله . النموذج كلاتي :

3. نعلن عن كائن الاتصال في منطقة الاعلانات العامة في النموذج حتي لا نضطر الي تكرار كتابته في كل اجراء في النموذج كلاتي

```
Public Class Form4
```

```
Dim con As New SqlConnection("server=. ; database=st_4 ;  
integrated security=true")
```

4. تحديث قاعدة البيانات (ادخال . حذف . تعديل . قراءة)

خ- ادخال البيانات الي جدول الطالب |

في حدث عند النقر على زر امر " جديد " نكتب

```
Try
```

```
Dim ad As New SqlDataAdapter("select* from stud",  
con)
```

```
Dim ds As New DataSet  
ad.Fill(ds, "stud")
```

```

Dim dr As DataRow = ds.Tables("stud").NewRow
    dr!id = TextBox1.Text
    dr!name = TextBox2.Text
    dr!b_date = datetimestr1.Text
    dr!dept = TextBox3.Text
    dr!term = ComboBox1.Text
    ds.Tables("stud").Rows.Add(dr)
Dim cm As New SqlCommandBuilder(ad)
ad.Update(ds, "stud")
MsgBox("بنجاح التسجيل تم")
Catch
    MsgBox("مدخلات .. خطأ")
End Try

```

د- حذف البيانات من جدول الطالب |

في حدث عند النقر على زر امر " حذف " نكتب

```

Try
Dim ad As New SqlDataAdapter("select * from stud
where id=" & TextBox1.Text, con)
Dim ds As New DataSet
    ad.Fill(ds, "stud")
    ds.Tables("stud").Rows(0).Delete()
Dim cm As New SqlCommandBuilder(ad)
ad.Update(ds, "stud")
MsgBox("بنجاح الطالب حذف تم")
Catch
    MsgBox("العملية في خطأ")
End Try

```

د- تعديل البيانات من جدول الطالب |

في حدث عند النقر على زر امر " تحديث " نكتب

```

Try
Dim ad As New SqlDataAdapter("select * from stud
where id=" & TextBox1.Text, con)
Dim ds As New DataSet
    ad.Fill(ds, "stud")
    ds.Tables("stud").Rows(0).Item("Name") =
        TextBox2.Text

```

```

ds.Tables("stud").Rows(0).Item("b_date") =
    datetimestricker1.Text
ds.Tables("stud").Rows(0).Item("dept") =
    TextBox3.Text
ds.Tables("stud").Rows(0).Item("term") =
    ComboBox1.Text
Dim cm As New SqlCommandBuilder(ad)
ad.Update(ds, "stud")
MsgBox("بنجاح التعديل تم")
Catch
    MsgBox("العملية في خطأ")
End Try

```

ر- البحث عن البيانات في جدول الطالب |
في حدث عند النقر على زر امر " بحث " نكتب

```

Try
Dim ad As New SqlDataAdapter("select * from stud
where id=" & TextBox1.Text, con)
Dim ds As New DataSet
ad.Fill(ds, "stud")
TextBox2.Text =
ds.Tables("stud").Rows(0).Item("Name")
datetimestricker1.Text =
ds.Tables("stud").Rows(0).Item("b_date")
TextBox3.Text =
ds.Tables("stud").Rows(0).Item("dept")
ComboBox1.Text =
ds.Tables("stud").Rows(0).Item("term")
Dim cm As New SqlCommandBuilder(ad)
ad.Update(ds, "stud")
Catch
    MsgBox("العملية في خطأ")
End Try

```

ز- استرجاع قيمه واحده في جدول الطالب |
في حدث عند تحميل النموذج نكتب

```

Dim ad As New SqlDataAdapter("select * from stud ",
con)
Dim ds As New DataSet
ad.Fill(ds, "stud")

```

Label6.Text = " | عدد الطلبة بالمعهد " &
ds.Tables("stud").Rows.Count & " طالب"

من - استعراض سجلات جدول الطالب |

نضيف نموذج جديد به DG لعرض جميع السجلات بالجدول :

id	name	b_date	dept
11	علي سليم	10/25/2000	كيمياء
1010	سعد محمد	12/6/1990	حاسوب
5656	سيب	11/1/2014	لايلى

في حدث عند تحميل النموذج الجديد نكتب :

```
Dim con As New SqlConnection("server=. ;  
database=st_4 ; integrated security=true")  
Dim ad As New SqlDataAdapter("select * from stud ",  
con)  
Dim ds As New DataSet  
ad.Fill(ds, "stud")  
Dg_s.DataSource = ds.Tables("stud")
```

□

تمت