المركز العالي للمهن الشاملة درنة قسم الحاسوب

الفصل الثالث حاسوب

ملخصات في بيسك المرئي(2)

Visual Basic .NET 2003

إعداد: إحسان مزهر رشيد

فصل الربيع 2005

الشكل1 إطار بدء تحميل VS .NET (Visual Studio .NET)

مقدمة

تحتوي هذه الملخصات على تعليمات عن نظام البرمجة بيسك المرئي للشبكات Visual Basic .NET (VB .NET) الإصـــدارة 2003 ، والـــذي يعمل تحت نظام التشغيل Windows2000 أو ما يليه. وهي مخصصة لطلبة الفصل الثالث حاســـوب حيث سبق لهم أن درَسوا مبادئ بيسك المرئي في الفصل الثاني ويدرسوا مبادئ قواعد البيانات في هذا الفصل.

فتهـــتم هذه الملخصات بتخزين البيانات باستخدام VB .NET حيث تبدأ بمراجعة سريعة والتعرف على بيئة VS .NET بعض مشاريع التعامل مع المسجل Registry، ثم التعامل مع الملفات من خلال VB .NET، ثم تطبيقات قواعد البيانات.

مراجعة Review

إنّ لغـــة البـــرمجة هي مجموعة من الأوامر والقواعد تستخدم لكتابة برنامج (Program) ، والأوامر عبارة عن تعليمات محددة المعنى تقوم بعمليات معالجة البيانات بحيث لا تتغير أشكال هذه التعليمات في اللغة الواحدة. إذن فالبرنامج عـــبارة عن مجموعة من الأوامر مكتوبة بلغة محددة بحيث تكون مرتبة ترتيباً منطقياً وموجهة لحل مسألة معينة. فيعتبر نظـــام البـــرمجة vb تطويـــراً للغة البرمجة QB ، ففي سنة 1991 ظهر الإصدار الأول vb1 وفي سنة 1998 الإصدار vb6.0 . والإصـــدار الأحدث هو vb7 أو VB .NET ظهرت سنة 2001 ثم ظهرت الإصدارة التي نحن بصدد دراستها وهـــي VB .NET2003، وهـــي جيل جديد لهذه اللغة تختلف جذرياً عن VB6.0 ، حيث أصبحت لغة برمجة باستخدام الكائنات الموجهة Object-Oriented programming (OOP) .

لقد تم الأخذ بنظر الاعتبار أنّ القارئ قد درَس أساسيات البرمجة وأساسيات بيسك المرئي في فصل سابق ولديه فكرة عن تصميم قواعد البيانات، والتعامل مع تطبيقات وملفات Windows.

إنّ كلمة مشروع Project في vb لها نفس معنى كلمة برنامج Program لكنها تشير إلى أنّ أكثر من ملف تعمل معـــاً ، وفـــي بيـــئة VS .NET أُطلقت تسمية حل Solution لنفس الغرض. وكلمة تطبيق Application هي الأخرى تحمـــل نفـــس المعنـــى فبعد الانتهاء من العمل بالمشروع نعمل نسخة تنفيذية تسمى تطبيق وهي النسخة التي توزَّع على المســـتخدمين Users. والمراحل الأساسية التي يمر فيها البرنامج منذ بداية كتابة الكود حتى الوصول إلى مرحلة التطبيق هي كما يلي: 1. البرنامج المصدري 2. البرنامج الهدفي 3. البرنامج التنفيذي (التطبيق)

البرنامج المصدري (Source Program) هو البرنامج الذي يكتبه المُبرمج ومفهوم من قِبل الإنسان.

البرنامج الهدفي (Object Program) هو البرنامج المكتوب بلغة الآلة.

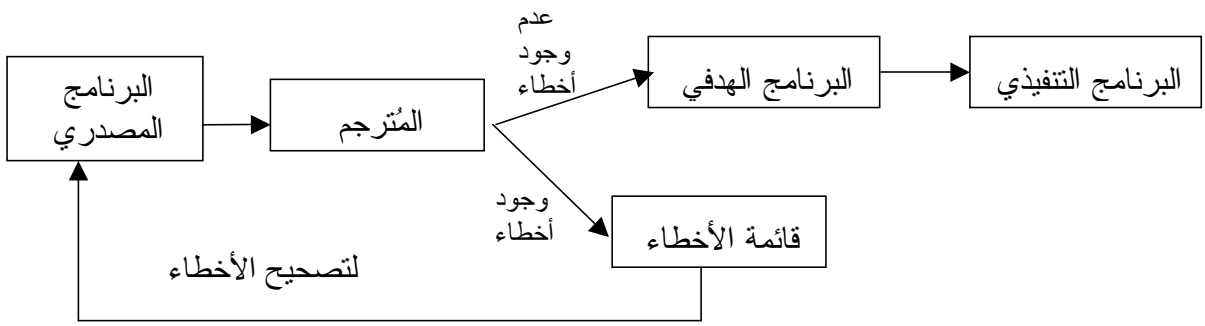
لغة الآلة (Machine Language) هي لغة البرمجة التي تُكتب تعليماتها بالشفرة الثنائية (Binary Code) 0 ، 1 .

المُتـــرجم (Compiler) هـــو البـــرنامج الذي يقوم بعملية تحويل البرنامج المصدري إلى برنامج هدفي، ويقوم المُترجم بالوظائف التالية 1. تحويل البرنامج المصدري الخالي من الأخطاء إلى برنامج هدفي

2. اكتشاف الأخطاء مثل أ. أخطاء إملائية ب. أخطاء قواعدية ت. أخطاء تنفيذية

3. ربط الجُمل الثنائية وبناء ما يُسمى بالبرنامج التنفيذي (Executable Program)

يبيّن الشكل التالي آلية عمل المُترجم



يمتاز المترجم في لغة VB بأنهُ مترجم فوري Interpreter أي بمجرد الانتهاء من كتابة سطر والانتقال إلى سطر آخر يقوم المترجم بترجمة الجملة. وهناك نوع آخر من المترجمات يقوم بالترجمة للبرنامج دفعة واحدة مثل مترجم لغة C.

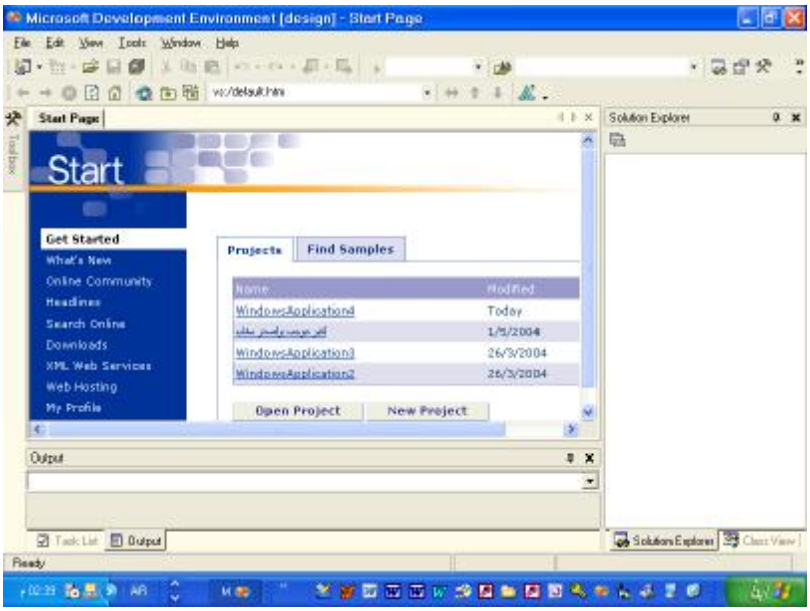
وفي VB .NET يقوم المترجم ببناء الملف التنفيذي داخل المجلّد bin في مجلّد الحل (المشروع).

تشغيل بيئة VS .NET لتشغيل بيئة VS .NET هناك عدة أساليب منها من القائمة ابدأ وكما يلي:

ابدأ> البرامج أو كافة البرامج > Microsoft Visual Studio .NET 2003 < Microsoft Visual Studio .NET 2003 .

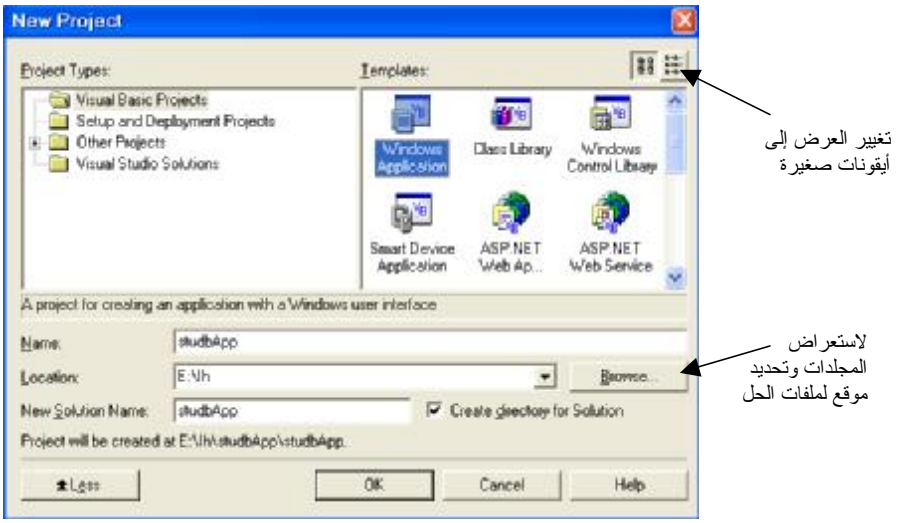
كمـــا يمكنـــنا إنشاء اختصار لـــ VS على سطح المكتب بالنقر بالأيمن على اسمهُ في القائمة ابدأ ثم نختار إرسال إلى > ســـطح المكـــتب كاختصـــار، وبالنقر المزدوج على أيقونته على سطح المكتب يتم تشغيلهُ، وأثناء تحميلهُ للذاكرة الرئيسية RAM يظهـــر إطـــار يحـــتوي رقم ونوع الإصدارة واسم المالك والمؤسسة كما في الشكل1 في الصفحة 0 ، ثم يختفي وتظهر بيئة VS .NET ، كما في الشكل 2 ، وفيها صفحة البداية Start Page.

للخروج من البيئة يكفي أن نغلق نافذتها الرئيسية أو من القائمة File > Exit أو كبس المفتاحين Alt+F4 معاً.



الشكل2 بيئة Visual Studio .NET صفحة البداية

لإنشـــاء مشـــروع جديـــد يمكنـــنا أن نختار من القائمة File>New> Project... Ctrl+Shift+N أو بالنقر على الزر New Project في صفحة البداية فيظهر الإطار New Project التالي

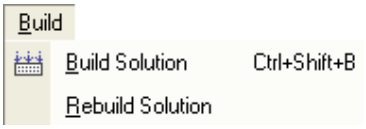


الشكل3 مربع حوار New Project في بيئة Visual Studio .NET

فنحدد نوع المشروع Visual Basic Projects و Windows Application ثم نكتب في الخانة Name: اسماً مناسباً للمشروع ونحدد موقعاً مناسبا في الخانة Location: ثم Ok فيقوم البرنامج بإنشاء مجلّد Folder بنفس اسم المشروع داخل الموقع المحـــدد لـــيحفظ فيه ملفات المشروع. في حالة كتابة اسم مشروع بنفس اسم وموقع مشروع سابق فستظهر الرسالة التالية في حالة تعطيل خانة التدقيق Create directory for Solution

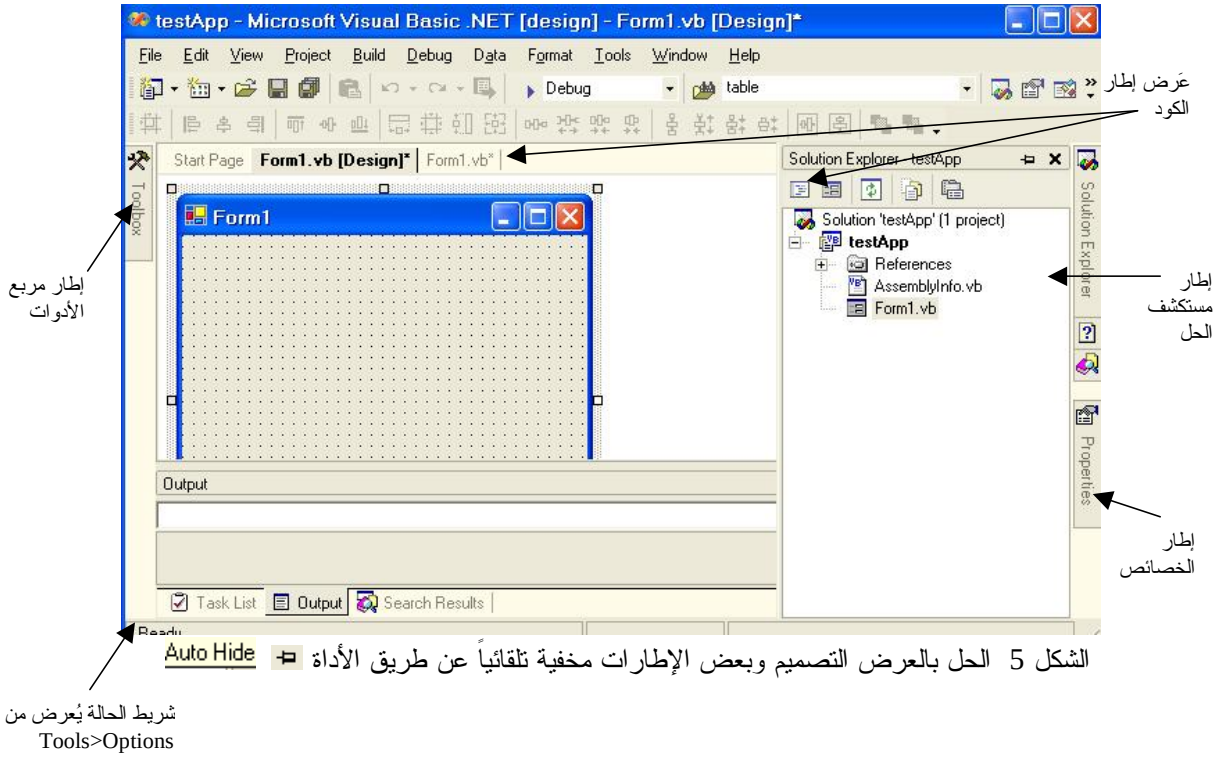
الشكل4 رسالة التحذير في حالة وجود الاسم

بعـد إنشـاء المشروع سيتم حفظ ملف الحل بالامتداد sln. وملف المشروع vbproj. وملفات كود النموذج بالامتداد vb. والملف المسـئول عن عَرض النموذج وقت التصميم بالامتداد resx. ، كما يحتفظ النظام بخيارات مستخدم الحل في ملف suo. وخيارات مستخدم المشروع في ملف امتدادهُ user. والنسخة التنفيذية داخل المجلّد bin بالامتداد exe. بعد التنفيذ أو من القائمة Build



حـيث يستخدم VB .NET نماذج ويندوز القياسية بمعنى أنّ النموذج يتم إنشاؤهُ وكذلك إنشاء عناصر تحكمهُ Controls مع ضبط خصائصها باستخدام الكود وعند التنفيذ.

الشـكل5 التالـي يُبين بيئة VB .NET والنموذج بالعرض التصميمي وإطار مستكشف الحلSolution Explorer والذي يحتوي على كائنات المشروع



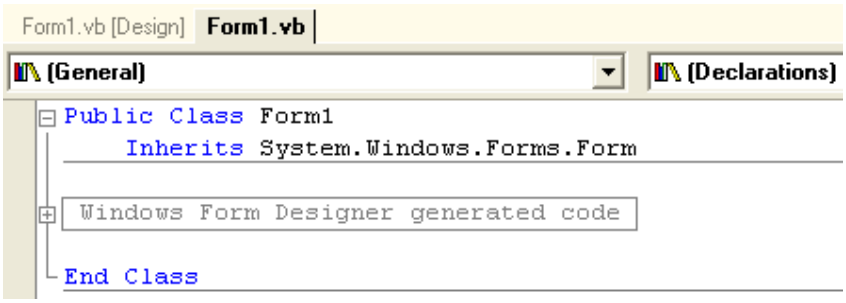
الشكل 5 الحل بالعرض التصميم وبعض الإطارات مخفية تلقائياً عن طريق الأداة

الاختـيار الافتراضـي للحفظ هو حفظ ملفات الحل كلّما تم التنفيذ أو بناء الحل ويمكن تغييرهُ عن طريق ..Tools>Options ثم الاختيارات تحت الاختيار Save all changes التابع للبند Projects and Solutions ، مثلاً Don't save changes to open documents

هـناك إطـارات غيـر ظاهرة في الشكل 5 أعلاه يُمكن إظهارها عن طريق القائمة View منها إطار مستعرض المُلقم Server Explorer وهـو لعـرض أسـماء المُلقمـات وقواعد بيانات SQL server والاتصالات التي تم إنشاؤها ، موقعهُ الافتراضي إلى جهة اليسار مع مربع الأدوات يُمكن تغيير موقعهُ بالسحب والإلقاء Drag and Drop لعلامة تبويبه أو لعنوانه.

إطار الكود

تعـتمد بيـئة VS .NET علـى نماذج Windows القياسية فبتكوين الحل الذي يحتوي على نموذج Windows فإنَّ النموذج سيُنشأ باستخدام الكود ومن الصفر حيث كود تصنيف (فصيلة) Class النموذج كما يلي:



نجد أنَّ اسم التصنيف (الفصيلة) Class باسم النموذج ومُعلن عنها في أول سطر، أما السطر الثاني فيَستخدم الوراثةInherits وهي مـــن مفاهـــيم البرمجة باستخدام الكائنات الموجهة OOP حيث يجعل التصنيف الجديد يرث جميع خصائص وأساليب التصنيف الأساسي الموجود في الجزء الثاني من السطر. بعد فتح منطقة الكود الأساسي بالنقر على علامة الجمع ستظهر مقاطعهُ.

الكود الأساسي لتكوين النموذج في تطبيق Windows

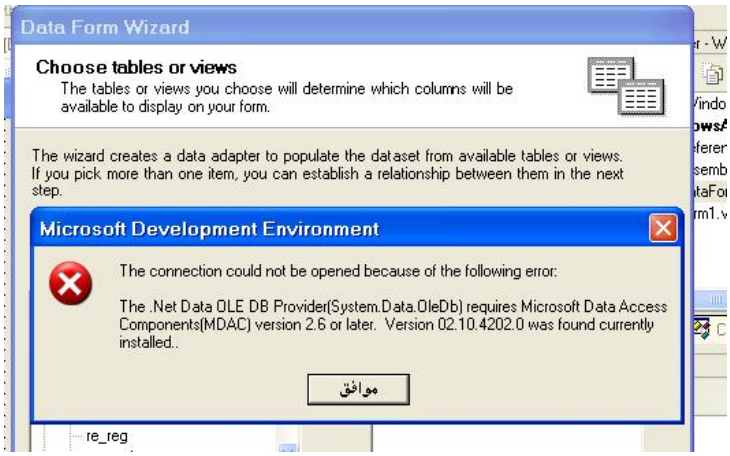
بعد إنشاء الحل سيتم تكوين النموذج من خلال الكود لا يحتوي على أي عنصر تحكم. والكود التالي يُبيّن كود تصنيف النموذج وكلّما أضفنا عنصر تحكم للنموذج أو ضبطنا خصائص كائن ما فإنّ VB .NET سيكتب ذلك في الكود

Form1 (Declarations)

```
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        components = New System.ComponentModel.Container()
        Me.Text = "Form1"
    End Sub

#End Region

End Class
```

بيســك المرئــي للشبكات أو VB .NET أو VB7.1 أحد أنظمة البرمجة في بيئة التطوير Visual Studio .NET والتي تحتوي أيضا على Visual C# (C Sharp) و Visual C++ و Visual J# ، ويُطلق على المشروع في هذه البيئة التسمية **حــل Solution** ، ومن المتطلبات والملحقات لبيئة التطوير VS .NET ما يلي:

1. يحتاج نظام تشغيل Windows2000 أو ما يليه مثل Windows XP أو Windows 2003 Server
2. يحتاج إطار عمل NET Framework. يأتي مع البرنامج بالإصدارة الأحدث 1.1 ، ويبقى التطبيق يحتاج إلى NET framework1.1 Redistributable. في جهاز المستخدم فيجب إرفاقهُ مع النسخ التوزيعية للتطبيق وهو بحجم 23.1MB
3. يحــتاج الوصــول إلى البيانات البرنامج Microsoft Data Access Components version 2.6 (MDAC) أو ما يليه والإصدارة الأحدث هي MDAC2.8 في 13/8/2003 ، وعند إعداد VS .NET2003 يتم إعداد هذا البرنامج بالإصدارة 2.7 تلقائياً، و تظهر الرسالة التالية في حالة عدم وجود إحدى الإصدارات المطلوبة عندما نرغب بإنشاء نموذج بيانات بمعالج النماذج ولا يتمكن المعالج (wizard) من إنشاء النموذج



الشكل 6 رسالة خطأ بعدم وجود MDAC بالإصدارة المطلوبة

ومن الملحقات بهذا البرنامج

Microsoft Data Access Components (MDAC) 2.8 Software Development Kit (SKD) 2004/3/17
Microsoft Data Access Components (MDAC) Security Patch MS04-003 (32-bit) 2004 /1/13

4. من الملحقات المهمة شبكة المطورين من مايكروسوفت Microsoft Developer Network (MSDN) وهي تعليمات لكيفية استخدام البرمجة في بيئة VS .NET وتأتي بثلاث أقراص مضغوطة (3CD) مع VS .NET 2003
5. من الملحقات المفيدة للمطورين في بيئة VS .NET وخصوصا باستخدام VB .NET هي Visual Basic NET Resource Kit وهي مجانية في موقع الشركة www.microsoft.com/downloads ويصل حجمها حتى 192MB .
6. في حالة بناء مشاريع للاتصال بمصدر بيانات مفتوح ODBC .NET. فيجب إعداد الملف الخاص بهذا الموفِّر وهو ODBC .NET Data Provider. وحجمهُ 848KB وغير مرفق مع VS .NET2003 .
7. عــند الحاجة إلى استخدام قواعد بيانات SQL server أو MSDE server وهي مجموعة جزئية من SQL server فيجب إنشــاء مُلقِّــم (server) خــاص باســتخدام MSDE (Microsoft Desktop Engine) وهــو متوفر مجاناً مع منتجات مايكروســوفت مــثل Office، فــيجعل من الممكن الوصول إلى قواعد بيانات مُلقم SQL فعندما يكون البرنامج يعمل تظهر أيقونتهُ بمنطقة System Tray (صينية النظام) ضمن شريط المهام .
8. البــرنامج VISIO وهــو برنامج رسومي لإنشاء المخططات، والبرنامج SourceSafe لتكوين قاعدة بيانات تخزّن الملفات المرافقة للمشروع لاستخدامها فيما بعد، ولتتبع الأخطاء التي تحصل بكود المشروع.

مشروع تجريبي

نرغب في هذا المشروع البسيط عرض الوقت والتاريخ في متسميتين Labels في النموذج فنضع في النموذج إضافة إلى المتســميتين موقّت Timer ونضبط الخاصية Enabled على True والخاصية Interval على 1000 مل ثانية ونكتب في المقطع Timer1_Tick السطر التالي:lblt.Text = TimeString لعرض الوقت، حيث Lblt هو اسم التسمية.

ولعرض التاريخ نستخدم في التسمية Lbld فنكتب السطر التالي في المقطع Form1_Activated مثلاً .

Lbld.Text = DateString

ملاحظة: يُمكن إيقاف عمل الموقّت باستخدام الأسلوب Timer1.Stop() ، وبدء عمل الموقّت بالأسلوبTimer1.Start()

مثال لعرض رسالة للمستخدم بعد مرور 10 ثواني فنكتب الكود التالي في موقّت

```
Static t As Short
t = t + 1
If t = 10 Then
    t = 0
    Timer1.Stop()
    MsgBox(" مرحبا بكم ؟ ", MsgBoxStyle.Information Or MsgBoxStyle.MsgBoxRight _
        Or MsgBoxStyle.MsgBoxRtlReading)
End If
```

تمــرين: طــوّر البــرنامج الأخير ليختبر التاريخ ففي حالة كون التاريخ أصغر من 2004/9/1 فتظهر رسالة نعم / لا بأنّ تاريخ الــنظام خطــأ هــل تريد تصحيحهُ، عند اختيار نعم يتم فتح نموذج آخر يحتوي على عناصر تحكم تُمكن المستخدم من تغيير التاريخ، مثل DateTimePicker أو ComboBox .

طرق تخزين البيانات

مــن طرق تخزين البيانات استخدام الكود في تخزين البيانات وكما سبق في المشروع السابق، استخدام المسجلات Registry وهــي قاعــدة بــيانات فــي نظام التشغيل لتخزين بيانات عن البرامج، استخدام الملفات مثل ملفات ini. (initialization) ملفات إعــدادات التكوين، وملفات xml. (extensible markup language) لغة الترميز القابلة للتوسيع، ... ، واستخدام قواعد البيانات Databases.

ومن الملفات التي تُستخدم لتخزين البيانات ملفات القيم المفصولة بفاصلة Comma-Separated Values File (CSV) .

مــثال: قــراءة محتويات ملف نوعهُ CSV وعرضها في مربع نص اسمه txtStu بمشروع نوعهُ Windows Application ، يجــب أن يكون الملف موجود في نفس مجلّد التطبيق، فيمكن تكوينهُ باستخدام المفكرة (Notepad) ونحفظهُ في المجلّد bin باسم "students.csv" وبترميز Unicode وليس ANSI ليُمكن قراءة النص العربي من خلال التطبيق.

كود التصنيف (الفصيلة) (class) الخاص بالنموذج كما يلي:

```
Imports System    'استيراد فضاء اسم namespace يحتوي على تصنيفات أساسية لم نستخدمها هنا
Imports System.IO 'فضاء اسم namespace يحتوي على تصنيفات تسمح بالقراءة والكتابة للملفات
Public Class Form1    ' تصنيف (فصيلة) تكوين النموذج
    Inherits System.Windows.Forms.Form  ' عبارة الوراثة وتشير إلى أنّ تصنيف النموذج يرث
    ' جميع خصائص وأساليب التصنيف الذي يمثلهُ الجزء الثاني من العبارة وهو التصنيف الأساسي
    Windows Form Designer generated code     ' منطقة كود تكوين النموذج
    Function funread(ByVal filna As String) As String
        '   إعلان عن متغير مثيل لتصنيف في فضاء الاسم
        Dim clsSr As New StreamReader(filna)
        'ReadToEnd إعلان عن متغير وتخصيص قيمة لهُ وهي محتويات الملف باستخدام الأسلوب
        Dim strcon As String = clsSr.ReadToEnd()
        clsSr.Close()  ' Instance إغلاق المثيل
        Return strcon   '  القيمة العائدة من الدالة
    End Function

'  مقطع (إجراء) خاص بالنموذج عند التحميل
Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
    Const deli = ","     '   ; ثابت يمثل الفاصل بين الحقول ويمكن أن يكون
    Dim strRes As String, file_contents As String, records() As String
    Dim field_names() As String, feilds() As String, record_number As Integer
    Dim Field_number As Integer
    strRes = ""
    'استدعاء الدالة وتمرير الوسيطة مسار التطبيق مع اسم الملف أو الاسم فقط
    file_contents = funread("students.csv")
    ' file_contents = fnread(Application.StartupPath & "\students.csv")
    records = Split(file_contents, vbCrLf) '  الفاصل بين السجلات هو بداية سطر
    '   السجل (الصف) الأول لأسماء الحقول والفاصلة تفصل بين أسماء الحقول
    field_names = Split(records(0), deli)
    For record_number = 1 To records.GetUpperBound(0)  ' دوران على جميع السجلات
        ' لتجاوز المسافات الفارغة في بداية ونهاية السجل ويكفي عمل ذلك للحقول
        records(record_number) = records(record_number).Trim
        If records(record_number).Length > 0 Then 'طول السجل موجب بمعنى يحتوي بيانات
            feilds = Split(records(record_number), deli) ' الفاصلة تفصل بين الحقول
            ' عرض رقم السجل وخط للفصل بين السجلات في العَرض
            strRes = strRes & record_number & "  ----------------------" & vbCrLf
            For Field_number = 0 To feilds.GetUpperBound(0) ' دوران على الحقول في السجل
                ' لتجاوز المسافات الفارغة في بداية ونهاية قيمة الحقل
                feilds(Field_number) = feilds(Field_number).Trim
                '   الاضافة للمتغير النصي
                strRes = strRes & field_names(Field_number) & ": " & feilds _
                   (Field_number) & vbNewLine
            Next
        End If
    Next
```

```
    '   عرض النتائج
    txtstu.Text = strRes 'أو.Substring(vbCrLf.Length)بلا السطر الفارغ في البداية إن وجد
    txtstu.Select(0, 0)  '  عدم اختيار (عدم تحديد) أي نص
End Sub
```

وكمثال على الملف ما يلي

وناتج القراءة من الملف والعرض في مربع النص كما يلي

تمـــرين: حوّر الكود في المشروع الأخير السابق بحيث يصبح عَرض البيانات في مربع النص كما في الشكل المجاور

حل التمرين: فيما يلي التحوير في الكود بعد الإعلان عن المتغيرات في المقطع Form_Load وتم تغيير أسماء الكائنات إلى اللغة العربية كتجربة لكتابة أسماء المتغيرات باللغة العربية.

```
'استدعاء الدالة وتمرير الوسيطة مسار التطبيق مع اسم الملف
محتويات_الملف = الحصول_على_محتويات_الملف(Application.StartupPath & "\students.csv")
السجلات = Split(محتويات_الملف, vbCrLf)  '  الفاصل بين السجلات هو بداية سطر
'   السجل (الصف) الأول لأسماء الحقول والفاصلة تفصل بين أسماء الحقول
اسماء_الحقول = Split(السجلات(0), الفاصل)
النتيجة = "رقم السجل"
For رقم_الحقل = 0 To اسماء_الحقول.GetUpperBound(0) ' دروان على أسماء الحقول في الصف الأول
    النتيجة = النتيجة & "     " & اسماء_الحقول(رقم_الحقل)
Next
النتيجة = النتيجة & vbCrLf & _
"    ------          --------          -------          -------" & vbCrLf
For رقم_السجل = 1 To السجلات.GetUpperBound(0)  '  دوران على جميع السجلات
    '  طول السجل موجب بمعنى يحتوي بيانات ولتجاوز الأسطر الفارغة في الملف
    If السجلات(رقم_السجل).Length > 0 Then
        الحقول = Split(السجلات(رقم_السجل), الفاصل)  '  الفاصلة تفصل بين الحقول
            النتيجة = النتيجة & "     " & رقم_السجل
            For رقم_الحقل = 0 To الحقول.GetUpperBound(0) ' دروان على الحقول في السجل
                ' لتجاوز المسافات الفارغة في بداية ونهاية قيمة الحقل '
                الحقول(رقم_الحقل) = الحقول(رقم_الحقل).Trim
                '   الاضافة للمتغير النصي
                النتيجة = النتيجة & "          " & الحقول(رقم_الحقل)
            Next
            النتيجة = النتيجة & vbCrLf
        End If
    Next
    '   عرض النتائج
      '   عرض النتائج
      txtstu.Text = النتيجة
      txtstu.Select(0, 0)  '  عدم اختيار (عدم تحديد) أي نص
End Sub
```

تخزين البيانات في المُسجِّل

مـــثال: (الـــتعامل مـــع المُسجِّل Registry) سنُصمم في هذا المثال مشروعاً يحتوي على ثلاث نماذج، وسنقوم بحماية التطبـــيق بكلمـــة مرور Password حيث أنّ الكلمة الافتراضية هي 0000 ويستطيع المستخدم تغييرها من خلال الـــنموذج الـــثالث Form3 والذي يظهر من القائمة أمان في النموذج الثاني. للوصول إلى المسجلات نستخدم الدالة GetSetting للقـــراءة من المسجِل، والدالة SaveSetting للكتابة (التخزين)إلى المسجِّل، والدالة DeleteSetting لحـــذف إعـــدادات تطبـــيق أو جزء منها أو مفتاح Key من المسجِّل، والمسجِّلات أو التسجيل registry هي قاعدة بـــيانات Database مركزية يستخدمها نظام التشغيل Windows لتخزين بيانات هامة تضم الإعدادات Settings والتكويـــنات Configurations الخاصـــة بـــنظام التشغيل نفسه والبرامج العاملة من خلاله والمعدات المتصلة به. ولإجـــراء تعـــديلات علـــى قاعدة البيانات السابقة الذِّكر يوفر Windows برنامج محرر المسجِل Rigistry Edit ولتشـــغيله نكتب الأمر regedit في خانة الأمر في إطار تشغيل... Run من القائمة ابدأ ثم موافق. وتركيب الدوال التي يوفرها VB .NET للتعامل مع المسجِل كما يلي:

للقـــراءة مـــن المسجِل في `Public Function GetSetting(AppName As String, Section As String, Key As String, [Default As String = ""]) As String`

حالة عدم وجود المفتاح Key المحدد فإنَّ الدالة GetSetting ستُعيد الوسيطة الرابعة في تركيبها إن وجِدت.

للكـــتابة إلـــى المسجِل `Public Sub SaveSetting(AppName As String, Section As String, Key As String, Setting As String)` إن لم يكن المفتاح موجوداً سيتم تكوينهُ.

لحذف مفتاح أو جزء أو إعدادات تطبيق `Public Sub DeleteSetting(AppName As String, [Section As String = Nothing], [Key As String = Nothing])`

بعـــد تكوين مشروع جيد، نضيف إلى المشروع نموذجين آخرين فيصبح فيه ثلاثة نماذج. تصميم النموذج الأول كما في الشكل المجاور

Form1

خصائص كائنات النموذج Form1

Object	Properties	Value
Form	FormBorderStyle	FixedSingle
	MaximizeBox	False
	Opacity	90%
	RightToLeft	Yes
	Text	كلمة المرور
Text Box	Name	txtp
	MaxLength	14
	PasswordChar	*
	Text	
Button	Name	btnok
	Text	&موافق
Button	Text	إل&غاء الأمر

مقاطع Form1 ومقطع زر الأمر موافق عند النقر كما يلي:

```
Private Sub btnok_Click(ByVal sender As System.Object, ByVal e As _
  System.EventArgs) Handles btnok.Click
    pwd = GetSetting("ap1", "sc2", "ke3", "00000")
    Static t As Byte
    If pwd = txtp.Text Then
        Dim frm As New Form2
        frm.Show()
        Me.Hide()
    Else
        t = t + 1
        If t = 3 Then
            MsgBox("إنتهت المحاولات", MsgBoxStyle.Critical, "خطأ")
            End       ' me.Close   أو   Application.Exit()
        End If
        MsgBox(" الكلمة خطأ ", MsgBoxStyle.Information Or _
             MsgBoxStyle.MsgBoxRight Or MsgBoxStyle.MsgBoxRtlReading _
              , " كلمة المرور ")
        txtp.Text = ""
        txtp.Focus()
        End If
End Sub
```

لكـــي يتعـــرّف الـــنموذج الأول والنموذج الثالث على متغير كلمة المرور pwd سنُعلن عنهُ في وحدة نمطية قياسية Module بعد إضافتها للمشروع من القائمة Project ثم الأمر Add Module... ثم Open ونكتب السطر التالي:

```
Module Module1
    Public pwd As String
End Module
```

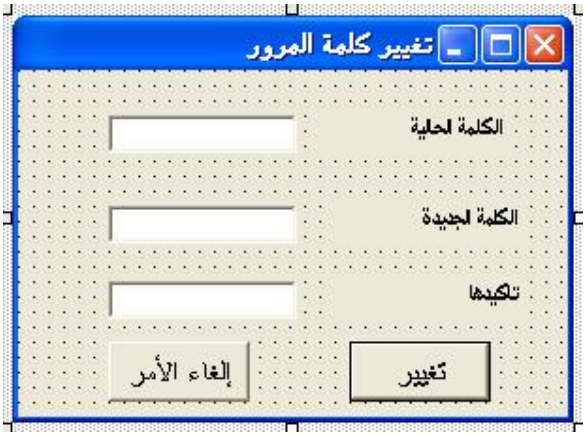
ونصمم في النموذج الثاني القائمة التالية

بإضافة عنصر تحكم MainMenu للنموذج ونكتب في كود البند السطرين التاليين :

```
Dim frm As New Form3
frm.ShowDialog()
```

ونكتب في المقطع Form2_Closed الأمر End

وتصميم النموذج الثالث Form3 كما يلي :



خصائص كائنات النموذج Form3

Object	Properties	Value
Form	BorderStyle	1 -Fixed Single
	Text	تغيير كلمة المرور
TextBox	Name	txtpold
	MaxLength	14

	TabIndex	0
	PasswordChar	*
TextBox	Name	txtpnew
	MaxLength	14
	PasswordChar	*
	Visible	False
TextBox	Name	txtpnew1
	MaxLength	14
	PasswordChar	*
	Visible	False
Label	Name	lblpOld
	Text	ادخل الكلمة الحالي
Label	Name	lblpnew
	Text	الكلمة الجديدة
	Visible	False
Label	Name	lblpnew1
	Text	تأكيدهـــــا
	Visible	False
Button	Name	btnCh
	Text	تغيير

مقاطع النموذج Form3، نكتب في مقطع تغيير عند النقر

```
If txtpnew.Text = txtpnew1.Text Then
    Dim m As String
    If txtpnew.Text = "" Then
        m = "  تم تغيير الكلمة بقيمة فارغة "
    Else
        m = " تم تغيير الكلمة "
    End If
    pwd = txtpnew.Text
    SaveSetting("ap1", "sc2", "ke3", pwd)
    MsgBox(m, MsgBoxStyle.Information, " تغيير الكلمة")
    Me.Dispose()
Else
    MsgBox(" يجب تأكيد نفس الكلمة ", MsgBoxStyle.Information, "")
    txtpnew.Text = ""
    txtpnew1.Text = ""
    txtpnew.Focus()
End If
```

ونكتب في مقطع مربع النص txtpOld عند النقر

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles txtpold.TextChanged
    If pwd = txtpold.Text Then
        txtpold.Visible = False
        lblpOld.Visible = False
        txtpnew.Visible = True
        lblpnew.Visible = True
        txtpnew1.Visible = True
        lblpnew1.Visible = True
        txtpnew.Focus()
        btnch.Enabled = True
    End If
End Sub
```

عبارة تصدّ الأخطاء Try وتركيبها كما يلي:

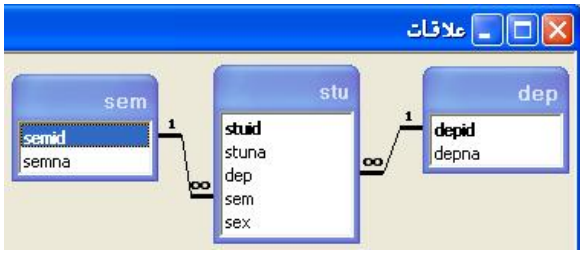
```
Try
    [ tryStatements ]
[ Catch [ exception [ As type ] ] [ When expression ]
    [ catchStatements ] ]
[ Exit Try ]
...
[ Finally
    [ finallyStatements ] ]
End Try
```

مشاريع قواعد البيانات

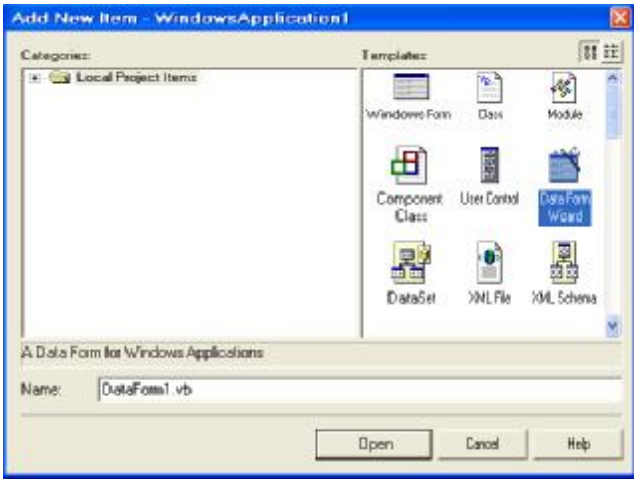
من طرق الاتصال بقاعدة البيانات الأكثر أهمية هي قواعد بيانات ملقِّم SQL (SQL server)، و OLE DB (Object Linking and Embedding) (ربـــط وتضـــمين الكائنات) وهو موفّر عام للبيانات للوصول إلى قواعد بيانات Access (Jet) و قـــواعد بيانات SQL server ، و ODBC (Open Database Connectivity) ، وبيانات XML (Extensible Markup Language) (لغة الترميز القابلة للتوسع)، وموفّر بيانات Oracle.

خطوات إنشاء نموذج بيانات باستخدام معالج نماذج البيانات Data Form Wizard

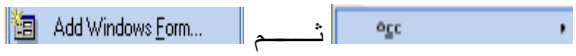
بعـــد إنشـــاء قاعـــدة البيانات باستخدام Access أو باستخدام Visual Data من بيئة Visual Basic نقوم بالاتصال بقاعدة البـــيانات من خلال نموذج في المشروع يُسمى نموذج بيانات ويوفـــر معـــالج نماذج البيانات خطوات سهلة للاتصال بقاعدة البيانات وإنشاء نموذج البيانات وكما يلي



العلاقـــات بين الجدول بعد تحديد ميزة فرض التكامل المرجعي Referential Integrity وتتالي التحديث، بدون تتالي الحذف.



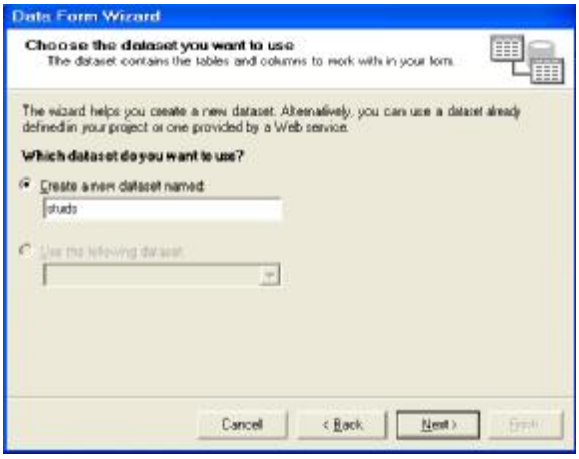
مـــن القائمة Project نختار الأمر Add Windows Form أو Add New Item فيظهـــر إطـــار إضافة بند جديد لتحديد نـــوع الـــنموذج فنخـــتار Data Form Wizard ، ويمكـــن الوصـــول إلـــى هذه المرحلة من خلال النقر بالأيمن على اسم الحـــل في إطار مستكشف الحل (Solution Explorer) ثم



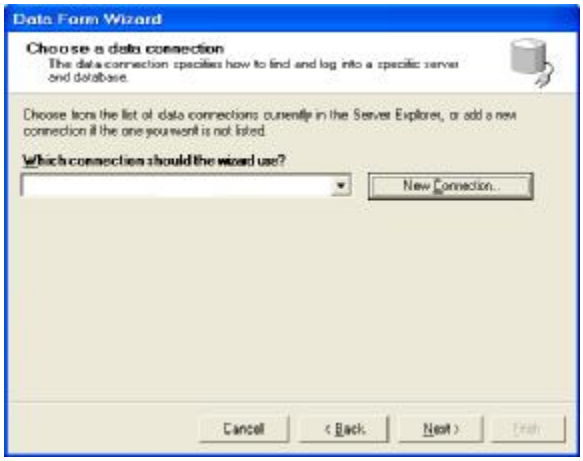
ثـــم أو

. وبعـــد الـــتحديد ســـيُعطي VB اســـما افتراضياً للنموذج ثم بالنقر على Open يبدأ عمل معالج نماذج البيانات

j الخطوة الأولى من معالج نماذج البيانات هي إظهار رسالة ترحيبية



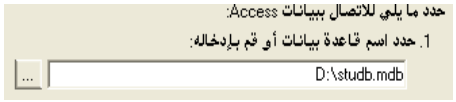
k فـــي الخطوة الثانية يخيّرنا معالج نماذج البيانات بين إنشاء مجمـــوعة بيانات جديدة Dataset أو استخدام مجموعة بيانات موجـــودة إن ســـبق لنا إنشاءها. ومجموعة البيانات عبارة عن لقطـــة snapshoot لقاعدة بيانات مصغّرة متكونة من الجداول والاســـتعلامات التي سنحددها من قاعدة البيانات في الخطوات التالـــية وهـــي بـــديل لمجمـــوعة الســـجلات Recordset في الإصـــدارات الســـابقة من VB ولكنها ذات إمكانات أكبر وعن طريقها يتم الاتصال بالبيانات في VB .NET مهما كانت طرق الاتصال.



l فـــي الخطـــوة الثالثة يخيّرنا معالج نماذج البيانات بين إنشاء اتصـــال جديـــدة New Connection... أو استخدام اتصال موجود سبق لنا إنشاؤه. وفي حالة اختيار New Connection فعليـــنا تحديـــد طريقة الاتصال في الخطوات التالية بتحديد نوع مُوفِّر (مُزوِّد) البيانات Data Provider

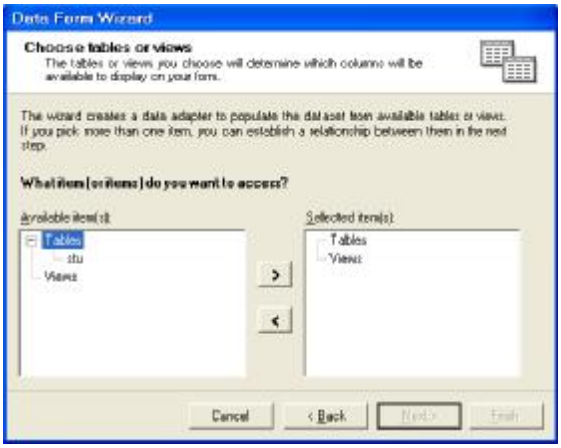
m ســيظهر إطــار خصائص الاتصال لتحديد الموفِّر وسنجد الموفِّر الافتراضي بالنقر على علامة التبويب الموفِّر Provider وهــو Microsoft OLE DB Provider for SQL Server ، سنستخدم في مثالــنا الموفِّــر Microsoft Jet 4.0 OLE DB Provider فـــنحددهُ ثـــم التالي >> أو انـتقاء علامـة التـبويب اتصـال Connection

n وفي هذه الخطوة نحدد ملف قاعدة البيانات بالنقر على ... ثم نحدد الملف

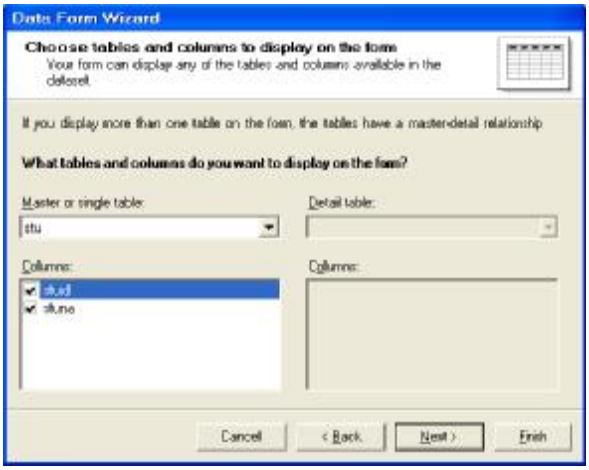


ثــم النقــر على اختبار الاتصال وفي حالة نجاح الاتصال تظهر الرسالة التالية

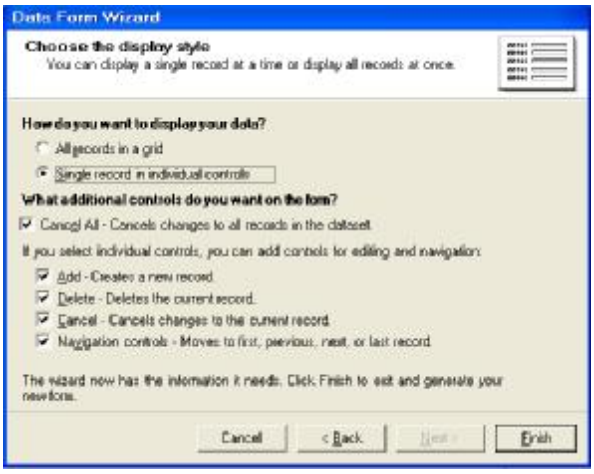
ثــم موافق ثــم موافق فينتهــي العمل في إطار خصائص الاتصال ونعود إلى معالج نماذج البيانات فننقر على Next >



o فـــي هذه الخطوة نحدد الجدول أو الجداول أو الاستعلامات المراد إظهار بياناتها من خلال النموذج وذلك بتحديد الجدول أو الاســتعلام وهـــي تحت Views ثم النقر على الزر > أو بالنقر المزدوج على اسم الجدول أو الاستعلام ثم Next >



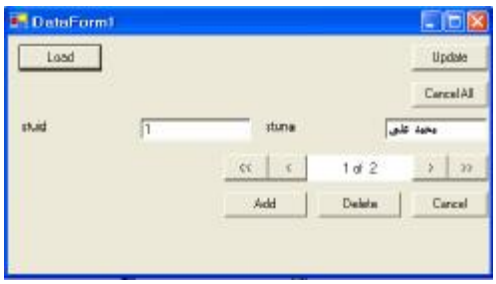
p هــذه الخطــوة مسئولة عن تحديد الحقول التي ستظهر في كائنات النموذج، ثم Next >



q هــذه الخطوة مسئولة عن تحديد طريقة عرض البيانات في شــبكة بيانات Grid (صفحة بيانات) أو بشكل مفرد لكل سجل فنحدد طريقة العرض المرغوبة ثم النقر على Finish فتظهر الرســالة التالــية لتخيــرنا تضــمين الكود صيغة احتواء قاعدة البــيانات علــى كلمة مرور أو لا ففي مثالنا لم نقم بتعيين كلمة مرور لقاعدة البيانات

فنخــتار تضمين أو عدم تضمين كلمة مرور حسبما نرغب في تصميم المشروع، فيتم إضافة نموذج بيانات إلى المشروع

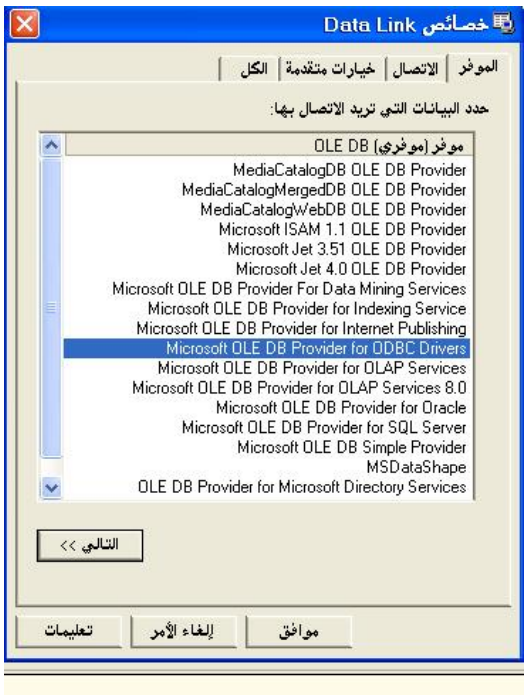
وبعــد أن نجعــل التنفيذ يبدأ بنموذج البيانات (من خلال القائمة Project > Properties ثم نختار من الخانة Startup object: اسم النموذج) سيظهر كما يلي بعد النقر على زر تحميل البيانات Load



فــيمكن قــص كــود الــزر Load ولصــقهُ فــي المقطع Form_Load ليــتم تحمــيل البيانات بمجرد التنفيذ. كما نقوم بتعريب النموذج وقت التصميم من خلال تنسيق كائنات النموذج وتغيير الخاصية Text لها.

في حالة اختيار نوع مُوفِّر آخر للاتصال بالبيانات فإنّ إطار خصائص الاتصال سيتغير حسب نوع المُوفِّر. كما أنّ معالج نماذج

البـــيانات لا يدعم المُوفِّر ODBC .NET . ففي حالة استخدام الموفـــر **ODBC .NET** فـــإنّ اسم الموفر سيظهر في إطار خصائص الاتصال وكما في الشكل المجاور ولكن معالج نماذج البيانات في VB .NET لا يدعم هذا الموفر لذلك في هذه الحالة يجب تصميم النموذج بدون المعالج. وقبل ذلك يجب أن نكون قد أنشأنا مصدر بيانات وكما يلي



إنشاء مصدر بيانات Open Database Connectivity (ODBC)

j في Windows 9x نشغّل من لوحة التحكم Control Panel البرنامج 

ODBC Data Sources (32 bit) (مصادر بيانات)

k فـــي WinXP فنشـــغّل البرنامج 

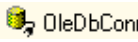
(Administrative Tools) أدوات إدارية ثم 

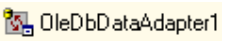
أو 

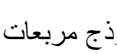
(BDE Administrator) ثم نختار من القائمة Object الأمر ... ODBC Administrator

ثـــم في علامة التبويب DSN المستخدم (user) ننقر على إضافة ثم نحدد نوع سواقة مصدر البيانات Microsoft Access Driver ثـــم إنهـــاء فيظهر إطار تكوين مصدر البيانات فنكتب اسم مصدر البيانات dbstu مثلاً، ثم تحديد لاختيار ملف قاعدة البيانات وهو الملف الذي أنشأناهُ باستخدام برنامج Access . ثم موافق ثم موافق . ولن نستخدم هذا الأسلوب لأنهُ قديم نسبياً.

تصميم النموذج وكود المعالج

بعد أن نعود إلى تصميم نموذج البيانات والذي قام بإنشائه معالج نماذج البيانات سنجد أنّ من بين عناصر التحكم المضافة إلـــى النموذج في مكان الكائنات غير المرئية the component tray ما يلي: كائن اتصال 

، كائن محوّل بـــيانات 

، وكائن مجموعة بيانات 

وفي النموذج مربعات نصوص Textbox ومتسميات توضـــيحية Labels وأزرار Buttons ، فلكـــل مـــربع نص تم ضبط الخاصية 

تحت الخاصية 

على اسم الحقل.

الكود The Code

نجـــد أنّ المعـــالج قد كتب الكود التالي (في تصنيف النموذج حيث اسم النموذج stuDataForm) وهو للاتصال بجدول stu في قاعدة البيانات D:\Ihsan\Databases\studb.mdb حيث يحتوي الجدول على خمسة حقول هيstuId, stuNa, sex, dep, sem

باستخدام موفر البيانات Microsoft Jet4.0 OLE DB واسم مجموعة البيانات هو studs وطريقة العرض سجل مفرد Single record مع تضمين كلمة مرور، وفيما يلي الكود الذي كتبهُ معالج النماذج بدون كود منطقة كود تكوين النموذج

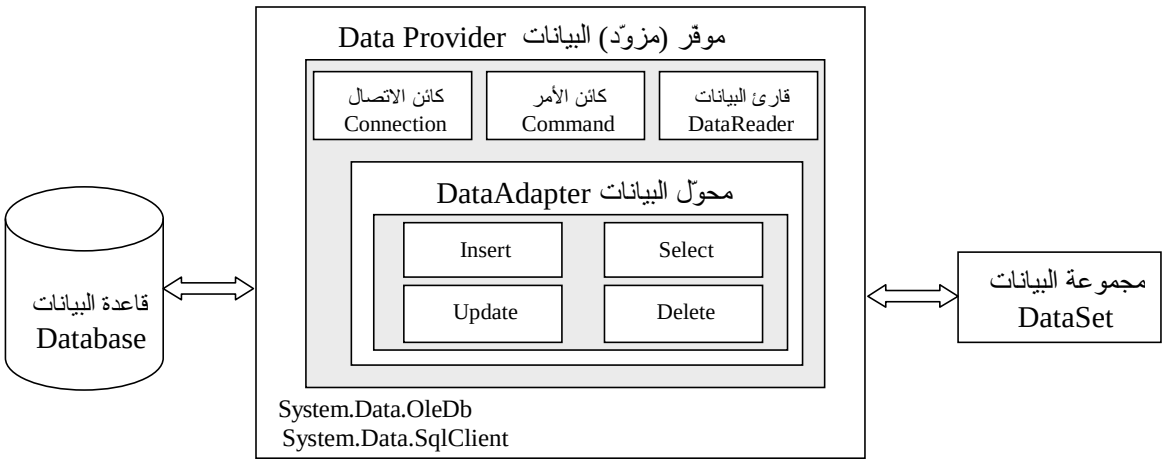
```
Public Class stuDataForm
    Inherits System.Windows.Forms.Form
    Windows Form Designer generated code
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As _
      System.EventArgs) Handles btnCancel.Click    ' _____
        Me.BindingContext(objstuds, "stu").CancelCurrentEdit()
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As _
         System.EventArgs) Handles btnDelete.Click   ' ___
        If (Me.BindingContext(objstuds, "stu").Count > 0) Then
            Me.BindingContext(objstuds, "stu").RemoveAt(Me.BindingContext _
               (objstuds, "stu").Position)
            Me.objstuds_PositionChanged()
        End If
    End Sub
    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As _
         System.EventArgs) Handles btnAdd.Click    ' _____
        Try
            'Clear out the current edits
            Me.BindingContext(objstuds, "stu").EndCurrentEdit()
            Me.BindingContext(objstuds, "stu").AddNew()
        Catch eEndEdit As System.Exception
            System.Windows.Forms.MessageBox.Show(eEndEdit.Message)
        End Try
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles btnUpdate.Click    ' ___
        Try
            'Attempt to update the datasource.
            Me.UpdateDataSet()
        Catch eUpdate As System.Exception
            'Add your error handling code here.
            'Display error message, if any.
            System.Windows.Forms.MessageBox.Show(eUpdate.Message)
        End Try
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As _
         System.EventArgs) Handles btnLoad.Click    ' ____
        Try
            'Attempt to load the dataset.
            Me.LoadDataSet()
        Catch eLoad As System.Exception
            'Add your error handling code here.
            'Display error message, if any.
            System.Windows.Forms.MessageBox.Show(eLoad.Message)
        End Try
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnNavFirst_Click(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles btnNavFirst.Click     ' __________
        Me.BindingContext(objstuds, "stu").Position = 0
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles btnLast.Click    ' __________
        Me.BindingContext(objstuds, "stu").Position = (Me.objstuds.Tables _
            ("stu").Rows.Count - 1)
        Me.objstuds_PositionChanged()
    End Sub
```

```
    Private Sub btnNavPrev_Click(ByVal sender As System.Object, ByVal e As _
           System.EventArgs) Handles btnNavPrev.Click  '
        Me.BindingContext(objstuds, "stu").Position = (Me.BindingContext _
          (objstuds, "stu").Position - 1)
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnNavNext_Click(ByVal sender As System.Object, ByVal e As _
          System.EventArgs) Handles btnNavNext.Click  '
        Me.BindingContext(objstuds, "stu").Position = (Me.BindingContext _
           (objstuds, "stu").Position + 1)
        Me.objstuds_PositionChanged()
    End Sub
    Private Sub btnCancelAll_Click(ByVal sender As System.Object, ByVal e As _
         System.EventArgs) Handles btnCancelAll.Click '
        Me.objstuds.RejectChanges()
    End Sub
    Private Sub objstuds_PositionChanged()
        Me.lblNavLocation.Text = (((Me.BindingContext(objstuds, "stu").Position _
     + 1).ToString + " of  ") + Me.BindingContext(objstuds, "stu").Count.ToString)
    End Sub
    Public Sub UpdateDataSet()
        'Create a new dataset to hold the changes that have been made to the main
        ' dataset.
        Dim objDataSetChanges As stuApp.studs = New stuApp.studs
        'Stop any current edits.
        Me.BindingContext(objstuds, "stu").EndCurrentEdit()
        'Get the changes that have been made to the main dataset.
        objDataSetChanges = CType(objstuds.GetChanges, stuApp.studs)
        'Check to see if any changes have been made.
        If (Not (objDataSetChanges) Is Nothing) Then
            Try
                'There are changes that need to be made, so attempt to update the
                'datasource(by)
                'calling the update method and passing the dataset and any
                'parameters.
                Me.UpdateDataSource(objDataSetChanges)
                objstuds.Merge(objDataSetChanges)
                objstuds.AcceptChanges()
            Catch eUpdate As System.Exception
                'Add your error handling code here.
                Throw eUpdate
            End Try
            'Add your code to check the returned dataset for any errors that
            ' may have been
            'pushed into the row object's error.
        End If
    End Sub
    Public Sub LoadDataSet()
        'Create a new dataset to hold the records returned from the call to
        'FillDataSet.
        'A temporary dataset is used because filling the existing dataset would
        'require the databindings to be rebound.
        Dim objDataSetTemp As stuApp.studs
        objDataSetTemp = New stuApp.studs
        Try
            'Attempt to fill the temporary dataset.
            Me.FillDataSet(objDataSetTemp)
        Catch eFillDataSet As System.Exception
            'Add your error handling code here.
            Throw eFillDataSet
        End Try
        Try
            'Empty the old records from the dataset.
            objstuds.Clear()
            'Merge the records into the main dataset.
            objstuds.Merge(objDataSetTemp)
        Catch eLoadMerge As System.Exception
```

```
            'Add your error handling code here.
            Throw eLoadMerge
        End Try
    End Sub
    Public Sub UpdateDataSource(ByVal ChangedRows As stuApp.studs)
        Try
          'The data source only needs to be updated if there are changes pending.
            If (Not (ChangedRows) Is Nothing) Then
                'Open the connection.
                Me.OleDbConnection1.Open()
                'Attempt to update the data source.
                OleDbDataAdapter1.Update(ChangedRows)
            End If
        Catch updateException As System.Exception
            'Add your error handling code here.
            Throw updateException
        Finally
            'Close the connection whether or not the exception was thrown.
            Me.OleDbConnection1.Close()
        End Try
    End Sub
    Public Sub FillDataSet(ByVal dataSet As stuApp.studs)
        'Turn off constraint checking before the dataset is filled.
        'This allows the adapters to fill the dataset without concern
        'for dependencies between the tables.
        dataSet.EnforceConstraints = False
        Try
            'Open the connection.
            Me.OleDbConnection1.Open()
            'Attempt to fill the dataset through the OleDbDataAdapter1.
            Me.OleDbDataAdapter1.Fill(dataSet)
        Catch fillException As System.Exception
            'Add your error handling code here.
            Throw fillException
        Finally
            'Turn constraint checking back on.
            dataSet.EnforceConstraints = True
            'Close the connection whether or not the exception was thrown.
            Me.OleDbConnection1.Close()
        End Try
    End Sub
End Class
```

تقنية ADO .NET (ActiveX Data Objects .NET)

يُبيّن الشكل التالي مجموعة الكائنات الخاصة بتقنية ADO .NET للوصول إلى البيانات



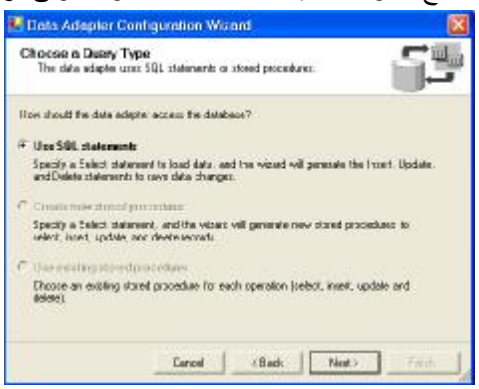
يُمكنــنا في نموذج البيانات قص كود المقطع الخاص بزر الأمر Load عند النقر ولصقه في المقطع Form_Load ليــتم فــتح الاتصـــال بقاعدة البيانات وتحميل البيانات من محوّل البيانات إلى مجموعة البيانات ثم إقفال الاتصال بمجرد التنفيذ أو

تحميل النموذج إلى الذاكرة الرئيسية (RAM). نلاحظ أنّ بيانات حقلي القسم والفصل تُعرض في مربعي نص وأنّ المعروض هو الرقم فنرغب بعرض الاسم (اسم القسم أو اسم الفصل) بدلاً من الرقم ذلك لأنّ النموذج هو واجهة للتعامل مع المستخدم، فنضع في النموذج مربع تحرير وسرد ComboBox اسمهُ cboDep لعرض أسماء الأقسام في بنوده ونفرّغ الخاصية Text. ومن خصائص مربع التحرير والسرد الخاصية Items وهي مصفوفة مسئولة عن بنود مربع التحرير والسرد، والفهرس Index للمصفوفة Items يبدأ من 0 إلى عدد البنود-1 ، والخاصية SelectedIndex مسئولة عن فهرس البند المنتقى وفي حالة عدم انتقاء بند فإنّ قيمتها هي 1- ، الخاصية SelectedItem مسئولة عن البند المنتقى نفسه وهو عادة نص، الخاصية Items.Count عدد البنود فهي بطبيعة الحال تزيد بواحد على أكبر قيمة ممكنة في الخاصية SelectedIndex ذلك لأنّ الخاصية SelectedIndex تبدأ من الصفر Zero-based . ومن أساليب Methods مربع التحرير والسرد الأسلوب Items.Add لإضافة بند جديد، الأسلوب Items.Insert الأسلوب Items.Remove الأسلوب Items.Clear.

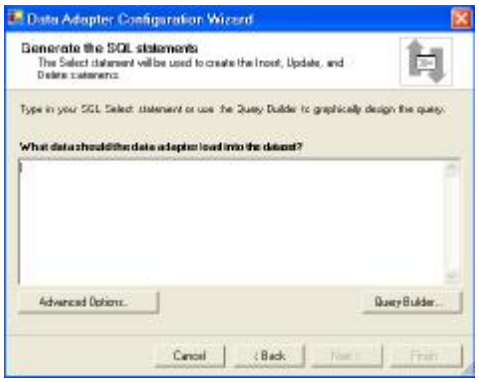
سنقوم بجعل مربع التحرير والسرد يعرض أسماء الأقسام وعند التنقل عبر السجلات يُعرض فيه اسم القسم لذلك الطالب وعند التحديث (الإضافة أو التغيير) في مربع التحرير والسرد يتم التغيير في الرقم الموجود في مربع النص editdep، ويُمكننا عمل ذلك بعدة طرق سنذكر منها طريقتين الأولى سنستخدم معالج محولات البيانات وهي بثلاث مراحل وهي كما يلي:

أولاً عرض أسماء الأقسام في مربع التحرير والسرد cboDep ويتم ذلك بالخطوات التالية

1. إضافة محوّل بيانات OleDbDataAdapter لتصميم النموذج فيبدأ معالج محولات البيانات بالعمل فالخطوة الأولى ترحيبية و في الخطوة الثانية نختار الاتصال connection وهو نفس الاتصال الذي استخدمناه سابقاً للاتصال بقاعدة البيانات، أما في الخطوة الثالثة فيظهر الإطار المجاور وهو لتحديد نوع الاستعلام أو طريقة وصول محول البيانات إلى البيانات في قاعدة البيانات، والاختيار المتاح هو Use SQL statements ويعني استخدام عبارات SQL لغة الاستعلام البنيوية Structured Query Language

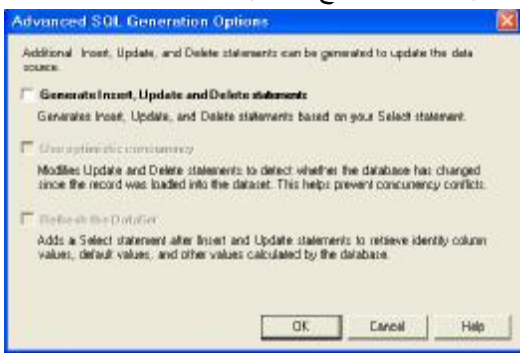


في الخطوة الرابعة نختار Query Builder... مُنشئ الاستعلام فيظهر إطارهُ



فنقوم بإضافة جدول dep إلى شبكة تصميم الاستعلام ثم نحدد الحقلين وكما في الشكل التالية

ستتم بذلك كتابة عبارة SQL ثم نختار OK فنعود إلى الخطوة الرابعة من المعالج ونختار Advanced Options... لتحديد



أوامر محوّل البيانات فنعطّل خانة التدقيق Generate Insert, Update and Delete statements وهناك

أربعة أوامر لمحوّل البيانات هي

SelectCommand لاختيار الحقول (الأعمدة)

InsertCommand لإدراج سجل (صف)

UpdateCommand لتحديث البيانات

DeleteCommand لحذف سجل (صف)

بذلك لم نستخدم إلا أمر الاختيار ثم OK ثم Next > في معالج محولات البيانات والخطوة الأخيرة هي خلاصة لاختياراتنا في الخطوات السابقة نختار منها Finish . ثم في تصميم النموذج نحدد محوّل البيانات ونضبط الخاصية (Name) oleDbDadep

2. إضافة كائن مجموعة بيانات DataSet فيظهر إطار إضافة مجموعة بيانات نختار Untyped dataset ثم OK ونضبط الخاصية (Name) objdepds

3. سنعبئ بنود مربع التحرير والسرد من خلال الكود وكما يلي:

أ. نُعلن في قسم الإعلانات العامة للنموذج عن مصفوفة stuDataForm (Declarations) ستخزّن أرقام الأقسام Dim arrdep( ) As String

ب. نكتب الكود التالي في المقطع FillDataSet فبعد السطر Me.OleDbConnection1.Open()الذي يفتح الاتصال مع قاعدة البيانات نكتب Me.oleDbDadep.Fill(Me.objdepds) بذلك يتم تعبئة مجموعة البيانات من محوّل البيانات. وفي الحقيقة فإنّ مجموعة البيانات عبارة عن بيانات في الذاكرة من عدة جداول قد تكون بينها علاقات يُحفظ مخططها في ملف xsd.

ت. نكتب في نفس المقطع FillDataSet بعد السطر الذي يغلق الاتصال Me.OleDbConnection1.Close() ما يلي:

```
Dim k As Byte, rno As Byte, strdep As String
rno = Me.objdepds.Tables.Item(0).Rows.Count - 1  '  عدد السجلات -1
For k = 0 To rno
    Me.cboDep.Items.Add(Me.objdepds.Tables.Item(0).Rows(k).Item(1))
    strdep = strdep & Me.objdepds.Tables.Item(0).Rows(k).Item(0) & ","
Next
arrdep = Split(strdep, ",")
```

حتى هذه المرحلة ننفذ Run لنتأكد من عملنا، يجب أن تظهر الأقسام في مربع التحرير والسرد.

ثانياً عرض اسم القسم للسجل الحالي في مربع التحرير والسرد cboDep عند التنقل عبر السجلات ويتم ذلك بإضافة الكود التالي للمقطع objstuds_PositionChanged والذي كتبهُ المعالج ومهمتهُ عرض رقم تسلسل السجل والعدد الكلي للسجلات أثناء التنقل

```
Dim k As Byte, po As Short = -1
For k = 0 To Me.objdepds.Tables.Item(0).Rows.Count - 1
    If arrdep(k) = Me.editdep.Text Then po = k : Exit For
Next
Me.cboDep.SelectedIndex = po
```

ثم ننفذ لتجربة الكود، فعند التنقل يجب أن يُعرض القسم في مربع التحرير والسرد.

ثالـــثاً عند التحديث (إضافة أو تغيير) في مربع التحرير والسرد cboDep يتم تحديث مربع النص editdep وذلك بإضافة الكود التالي للمقطع الخاص بمربع التحرير والسرد عند تغيير البند المنتقى cboDep_SelectedIndexChanged

```
If Me.cboDep.SelectedIndex <> -1 Then  ' هناك بند منتقى
    Me.editdep.Text = arrdep(Me.cboDep.SelectedIndex)
End If
```

عند التنفيذ سنجد أنّ مربع التحرير والسرد أصبح مرتبطاً بحقل القسم في جدول الطلبة ويعرض أسماء الأقسام.

الطـــريقة الثانية لجعل مربع التحرير والسرد مرتبطاً بالحقل سنطبقها على حقل الفصل وبدون استخدام المعالج حيث نقوم بإضـــافة مـــربع تحرير وسرد (combo box) لتصميم نموذج الطلبة ونسميه cbosem ونفرّغ الخاصية Text وقت التصميم ثم نطوّر المقطع FillDataSet وبدون إضافة كائن محوّل بيانات أو مجموعة بيانات وقت التصميم للنموذج وكما يلي:

1. نُعلـــن عـــن كائن كمحوّل بيانات daSem ونكتب عبارة SQL التي تنتقي جميع حقول جدول الفصول (sem) ، كما نُعلن عن كائن مجموعة بيانات dsSem وذلك قبل العبارة Try

```
Dim daSem As New OleDb.OleDbDataAdapter("SELECT * FROM [sem]", OleDbConnection1)
Dim dsSem As New DataSet

Try
```

هذه العبارة موجودة في المقطع وهي بداية تصيّد الخطأ

2. تعبـــئة كـــائن مجموعة البيانات من محوّل البيانات بجدول الفصول بعد سطر فتح الاتصال الموجود في المقطع FillDataSet وذلك بالسطر التالي:

```
daSem.Fill(dsSem, "sem")
```

3. ثم نقوم بضبط ثلاث خصائص لمربع التحرير والسرد من خلال الكود هي:

DisplayMember مسئولة عن اسم الحقل الذي يحتوي النص المراد عرضه الموجود في الخاصية DataSource

ValueMemder مسئولة عن اسم حقل القيمة المقابلة للنص والموجود في الخاصية مصدر عنصر التحكم DataSource

DataSource الخاصية مصدر عنصر التحكم وهي اسم مجموعة بيانات أو مصفوفة يستند عليها مربع التحرير والسرد ببنوده

ويتم ذلك بكتابة الكود التالي بعد سطر إغلاق الاتصال لنجعل فترة فتح الاتصال أقل ما يمكن

```
cbosem.DisplayMember = "sem.semna"
cbosem.ValueMember = "sem.semid"
cbosem.DataSource = dsSem
cbosem.DataBindings.Add("SelectedValue", objstuds, "stu.sem")
```

حساسة لحالة الأحرف

اسم الحقل

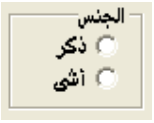
للتحديث

الســـطر الأخيـــر الســـابق لربط الخاصية SelectedValue لمربع التحرير والسرد بحقل الفصل في جدول الطلبة وذلك لتحديث الفصل حسب الاختيار الذي يحدده المستخدم في مربع التحرير والسرد، فاستخدمنا اسم كائن مجموعة البيانات لجدول الطلبة.

الخاصـــية DataBindings (ربط البيانات) تُعيد ربط بيانات بعنصر تحكم. وتستخدم لإضافة كائن ربط لمجموعة الربط لعنصر التحكم ControlBindingsCollection لربط أي خاصية لعنصر التحكم ربطها بالخاصية التي يملكها الكائن. فأول وسيطة عند الإضافة باستخدام الأسلوب Add هي اسم الخاصية كنص، والوسيطة الثانية اسم مجموعة البيانات، والثالثة لاسم الحقل كنص.

نلاحظ أننا في الأسلوب الأول لم نستغني عن مربع النص editdep ، بينما في الأسلوب الثاني يُمكننا الاستغناء عن مربع النص editsem بحذفه. الأسلوب الثاني يحتاج معرفة بعبارات لغة SQL ، وفي حالة وجود فراغات في أسماء الجداول أو أسماء الفصول فنضع اسم الجدول أو اسم الحقل بين مُعقفات [ ] مثلا "[جدول الطلبة].[رقم القسم]" .

بالنسبة للحقول المرمزة ترميزاً مغلقاً فنضع إطار مجموعة Group Box ثم نرسم بداخله أزرار خيار Radio Buttons بعـــدد القـــيم الممكـــنة الظهور ونضبط الخاصية Text للمجموعة ولأزرار الخيار بالنص المناسب، فمثلاً لحقل الجنس نكتب في الخاصية Text للمجموعة على الجنس، ولأحد أزرار الخيار ذكر وللآخر أنثى كما نضبط خاصية Name لها كما يلي:

RadM للذكر و RadF للأنثى 

ونكتب في المقطع objstuds_PositionChanged الكود التالي:

```
Select Case editsex.Text
    Case 1
        Me.RadM.Checked = True
        Me.RadF.Checked = False
    Case 2
        Me.RadM.Checked = False
        Me.RadF.Checked = True
    Case Else     '  إذا كان الحقل غير معبأ
        Me.RadM.Checked = False
        Me.RadF.Checked = False
End Select
```

ثم نكتب في كل من زري الخيار في الحدث عند التغيير سطراً مهمتهُ تغيير القيمة في مربع النص وكما يلي:

```
Private Sub RadM_CheckedChanged(ByVal sender As System.Object, ByVal e _
    As System.EventArgs) Handles RadM.CheckedChanged
    If Me.RadM.Checked = True Then Me.editsex.Text = 1
End Sub
Private Sub RadF_CheckedChanged(ByVal sender As System.Object, ByVal e _
   As System.EventArgs) Handles RadF.CheckedChanged
    If Me.RadF.Checked = True Then Me.editsex.Text = 2
End Sub
```

عـند التنفـيذ سـنجد أنّ أزرار الخيار أصبحت مرتبطة بمربع النص ولكن لا نستطيع الاستغناء عن مربع النص بل نقوم بضبط الخاصية Enabled على False وتغيير موقعهُ بحيث لا يُرى بضبط الخاصية X على 20- مثلاً وهي تتبع الخاصية Location بحيث لا يُرى وحذف التسمية المرافقة له.

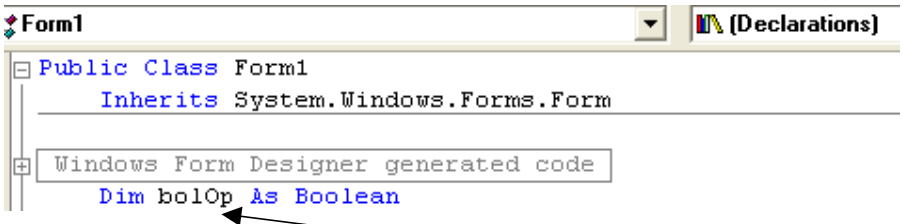
أمـثلة عن عَرض النماذج: (مثال اللاشفافية) لعرض النموذج بشكل متدرج من حيث الشفافية، وكذلك عند إغلاق النموذج باسـتخدام زر أمر فيختفي بشكل تدريجي نضع Timer ونضبط الخاصية Enabled = True والخاصية Interval = 10 ، وبالنسـبة للنموذج فنضبط الخاصية Opacity=0% (اللاشفافية)، والخاصية CancelButton على اسم زر الأمر إغلاق، ونكتب في مقطع الموقّت السطرين التاليين:

```
If Me.Opacity >= 1 Then Me.Timer1.Stop()
Me.Opacity += 0.05
```

ونكتب في المقطع الخاص بالنموذج حدث عند الإغلاق Closed الكود التالي:

```
Do While Me.Opacity > 0.1     '  DataForm1_Closedفي المقطع
     Me.Opacity -= 0.05
Loop
```

أو نستخدم الأسلوب التالي: نُعلن عن متغيّر منطقي



الإعلان عن متغير منطقي في قسم الإعلانات العامة للنموذج

```
bolOp = True
```

نخصص القيمة True للمتغير bolOp في المقطع Form_Load وكالتالي

بعد ذلك نكتب الكود التالي في مقطع الموقّت

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As _
        System.EventArgs) Handles Timer1.Tick
    If bolOp Then
        If Me.Opacity < 1 Then Me.Opacity = Me.Opacity + 0.05
        If Me.Opacity = 1 Then Me.Timer1.Enabled = False
    Else
        If Me.Opacity > 0.1 Then Me.Opacity = Me.Opacity - 0.05
        If Me.Opacity <= 0.1 Then Application.Exit()
    End If
End Sub
```

ثم نكتب السطرين التاليين في المقطع الخاص بزر الأمر إغلاق عند النقر

```
bolOp = False
Me.Timer1.Enabled = True
```

عند التنفيذ سيظهر النموذج تدريجياً وعند النقر على إغلاق أو ضغط المفتاح Esc سيختفي النموذج تدريجياً.

مثال (رسم خط) لعرض خط Line في النموذج نقوم برسمه وقت التنفيذ بكتابة الكود التالي في الحدث Form_Load

```
Me.Show()
Dim g As Graphics = Me.CreateGraphics
Dim pen1 As New Pen(Color.Red)
pen1.Width = 1
g.DrawLine(pen1, 0, 100, Me.Width, 100)
```

يُمكن فهم هذا الأسلوب في الرسم بالخطوتين التاليتين:

1. تكوين كائن رسم Graphics Object وهذا الكائن يشبه لوحة الرسم، وذلك باستدعاء الأسلوب CreateGraphics فالسطر Dim g As Grephics = Me.CreateGraphics يُقابل السطرين التاليين: Dim g As Graphics
g = CreateGraphics

2. استخدام كائن الرسم في رسم الخطوط والأشكال، وذلك بتكوين قلم Pen للرسم أو فرشاة Brush والفرشاة أنواع، أو تكوين قلم من كائن Brush موجود، وعند تكوين القلم أو الفرشاة يجب تحديد اللون باستخدام العنصر Color ويُمكن تحديد عُرض الخط وكما في السطر التالي: Dim pr As New Pen(Color.Red, 2) ويُقاس العُرض في VB .NET بالبكسل Pixel ، والبكسل هي نقطة منفردة على الشاشة تشكل كافة المنظر المعروض فيعتمد طولها على دقة الشاشة فمثلاً (Pixel= $\frac{1}{72}$ Inch) بدقة شاشة 800×600 بكسل Resolution، بينما في VB6.0 كانت وحدة القياس الافتراضية التوِب Twip حيث أنَّه وحدة مسافة $\frac{1}{1440}$ Inch = Twip. بعد تكوين كائن الرسم والقلم أو الفرشاة نقوم باستخدام أحد أساليب كائن الرسم مثل DrawLine لرسم خط مستقيم، الأسلوب DrawRectangle لرسم مستطيل أو مربع، الأسلوبDrawEllipse لرسم قطع ناقص أو دائرة. وهناك أساليب رسم أخرى. ومن تركيبات الأسلوب `DrawLine` هو تحديد القلم وإحداثيات النقطتين وكما يلي:

```
Public Sub DrawLine(pen As System.Drawing.Pen, x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer)
```

و `DrawRectangle`

```
Public Sub DrawRectangle(pen As System.Drawing.Pen, x As Integer, y As Integer, width As Integer, height As Integer)
```

الأسلوب `DrawEllipse`

```
Public Sub DrawEllipse(pen As System.Drawing.Pen, x As Integer, y As Integer, width As Integer, height As Integer)
```

حيث النقطة (0 , 0) الافتراضية في النموذج هي الزاوية العليا اليسرى للنموذج، ووحدة القياس الافتراضية للإحداثيات هي البكسل وهو $\frac{1}{72}$ انج وحسب دقة الشاشة Resolution. ومن أمثلة الألوان للعنصر Color هي `Color.Red`، `Color.Pink`، `Color.Black`، `Color.White`، `Color.Yellow`، `Color.YellowGreen`.

لقد استخدمنا في الكود السابق الخاصية Width للنموذج وهي مسئولة عن عُرض النموذج وبشكل عام عن عُرض الكائن، ومن أخواتها الخاصية Height الارتفاع ، والخاصية Top أعلى، والخاصية Left يسار، فالخاصيتين Width و Height تتحكمان بحجم النموذج، بينما الخاصيتين Left و Top للتحكم بموقع النموذج على الشاشة.

مثال (واجه التطبيق) لجعل نماذج المشروع تُعرض بأسلوب واجهة متعددة المستندات (MDI) Multiple Document Interface نضبط الخاصية IsMdiContainer True للنموذج الأب، وعند عرض النموذج الابن من خلال النموذج الأب نحدد لهُ خاصية الأب MdiParent وكما يلي:

```
Dim frm As New Form2
frm.MdiParent = Me  ' الأب للنموذج الذي سيُعرض هو النموذج الحالي
frm.Show()
```

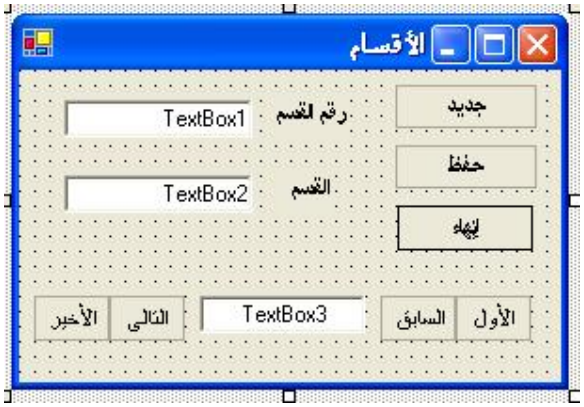
بينما الواجهة الافتراضية لنماذج التطبيق هي SDI (Single Document Interface).

مثال (النموذج المشروط) لعرض نموذج بشكل مشروط بمعنى لا يمكن الاستمرار بالعمل بالنماذج الأخرى بالتطبيق ما لم يتم الانتهاء من العمل بالنموذج المشروط وإغلاقهُ، فنستخدم الأسلوب ShowDialog ، ويجب أن لا يكون نموذج ابن، بمعنى أنّ النموذج المشروط يجب أن يكون SDI form . ويُسمى عند إذٍ Modal (مشروط). أما عند العرض باستخدام الأسلوب Show فيُسمى أسلوب العَرض بـ Modeless . وعَرض النموذج مشروطاً يختلف عن مفهوم عرض النموذج في المقدمة الذي تتحكم به الخاصية TopMost وهي خاصية منطقية عند ضبطها على True تجعل النموذج في مقدمة النماذج ولكن يُمكن التعامل مع بقية نماذج التطبيق على الرغم من وجوده.

مثال (الأداة DateTimePicker لاختيار التاريخ) إذا كان لدينا حقل تاريخ فيمكننا أن نربطهُ بمربع اختيار التاريخ عن طريق الخاصية Text التابعة للخاصية (DataBindings) ونختار التنسيق المناسب بالخاصية Format ، أو استخدام تنسيق مخصص وكما يلي:

```
dateTimePicker1.Format = DateTimePickerFormat.Custom
dateTimePicker1.CustomFormat = "MMMM dd, yyyy – dddd"
```

ربط البيانات باستخدام الكود فقط



نصمم نموذج الأقسام كما في الشكل المجاور وبعد وضع كائن اتصال نضبط الخاصية ConnectionString باستخدام معالج خصائص الربط والذي يظهر باختيار New Connection ثم ننسخ قيمة الخاصية ونلصقها بالكود بتخصيص ثابت نصي على مستوى الوحدة لها ثم إزالة أي علامات تنصيص بداخل قيمة الخاصية وكما سيأتي مع الثابت المسمى strConn. أسماء مربعات النصوص هي txtdepid ، txtdepna ، txtRecord. واسم النموذج frmdep ، ونضبط الخاصية FlatStyle لأزرار الأمر على Popup أو Flat .

1. نكتب في قسم الإعلانات العامة ما يلي:

```
(General)                    (Declarations)
Option Strict On
Option Explicit On
Imports System.Data.OleDb
Public Class frmdep
    Inherits System.Windows.Forms.Form
```

2. تكوين منطقة كود للثوابت والمتغيرات على مستوى الوحدة

```
#Region " الثوابت على مستوى الوحدة"
    ' ConnectionString  ثابت نص الاتصال يتم نسخة من قيمة الخاصية
    Const strConn As String = "Jet OLEDB:Global Partial Bulk Ops=2;" & _
                "Jet OLEDB:Registry Path=;Jet OLEDB:Database Locking Mode=1;" & _
                "Data Source=D:\studb.mdb;Mode=Share Deny None;" & _
                "Jet OLEDB:Engine Type=5;Provider=Microsoft.Jet.OLEDB.4.0;" & _
                "Jet OLEDB:System database=;Jet OLEDB:SFP=False;" & _
                "persist security info=False;Extended Properties=;" & _
                "Jet OLEDB:Compact Without Replica Repair=False;" & _
                "Jet OLEDB:Encrypt Database=False;" & _
                "Jet OLEDB:Create System Database=False;" & _
                "Jet OLEDB:Don't Copy Locale on Compact=False;" & _
                "User ID=Admin;Jet OLEDB:Global Bulk Transactions=1"

    ' عبارة SQL  الخاصة باختيار الحقول كثابت
    Const strSELECT As String = "SELECT * FROM  dep ORDER BY dep.depid"
    Dim conn As OleDbConnection      '   كائن الاتصال
    Dim cmd As OleDbCommand          '   كائن الأمر
    Dim da As OleDbDataAdapter       '   كائن محوّل البيانات
    Dim ds As DataSet                '   كائن مجموعة البيانات
    Private bNewMode As Boolean = False   '  لمعرفة السجل جديد أم لا
#End Region
```

3. منطقة كود للمقاطع

```
#Region "المقاطع"
    Private Sub RefreshData(Optional ByVal noRecToShow As Int32 = 0)
        ds.Clear()
        Try
            cmd.CommandText = strSELECT     '  نص كائن الأمر
            ClearForm()    '  استدعاء مقطع افراغ عناصر التحكم
            da.Fill(ds, "dep")
            Me.BindingContext(ds, "dep").Position = noRecToShow
        Catch ex As OleDbException
            MsgBox("RefreshData(): " & ex.Message)
        End Try

    End Sub
    Private Sub UpdateRecordNumber() '  لإظهار رقم تسلسل السجل والعدد الكلي للسجلات
        Dim noRec As Int32
        noRec = Me.BindingContext(ds, "dep").Count
        If noRec > 0 Then
            txtRecord.Text = CStr(Me.BindingContext(ds, "dep").Position + 1) & _
                 " من " & noRec
        Else
            txtRecord.Text = ""
        End If

    End Sub
    Private Sub ClearForm()    '  لإفراغ عناصر التحكم
        txtdepid.Text = ""
        txtdepna.Text = ""
    End Sub
#End Region
```

4. مقاطع أحداث الكائنات وضِعت بمنطقة خارج منطقة كود تكوين النموذج

```
Windows Form Designer generated code
#Region "مقاطع الأحداث"
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As _
         System.EventArgs) Handles MyBase.Load
        conn = New OleDbConnection(strConn)    '   كائن الاتصال
        cmd = New OleDbCommand                 '   كائن الأمر
        da = New OleDbDataAdapter              '   كائن محوّل البيانات
        ds = New DataSet                       '   كائن مجموعة البيانات
        cmd.CommandText = strSELECT
        cmd.Connection = conn
        da.SelectCommand = cmd
        conn.Open()
        RefreshData()    '    استدعاء مقطع لتعبئة مجموعة البيانات
        conn.Close()
        '   ربط عناصر التحكم بالحقول
        Me.txtdepid.DataBindings.Add(New System.Windows.Forms.Binding("Text", _
             Me.ds, "dep.depid"))
        Me.txtdepna.DataBindings.Add(New System.Windows.Forms.Binding("Text", _
             Me.ds, "dep.depna"))
        Me.BindingContext(ds, "dep").Position = 0  '  السجل الحالي هو الأول
        UpdateRecordNumber()    '  لعرض تسلسل السجل والعدد الكلي للسجلات

    End Sub
    ' مقطع خاص بزر الأمر السجل التالي '
    Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As _
           System.EventArgs) Handles btnNext.Click
        Dim iPos As Int32
        Dim iCount As Int32
        iPos = Me.BindingContext(ds, "dep").Position + 1
        iCount = Me.BindingContext(ds, "dep").Count
        If iPos < iCount Then
            Me.BindingContext(ds, "dep").Position += 1
            UpdateRecordNumber()
            bNewMode = False
        End If
    End Sub
    ' مقطع خاص بزر الأمر السجل السابق '
    Private Sub btnPrevious_Click(ByVal sender As System.Object, ByVal e As _
             System.EventArgs) Handles btnPrevious.Click
        Dim iPos As Int32
        iPos = Me.BindingContext(ds, "dep").Position + 1
        If iPos > 0 Then
            Me.BindingContext(ds, "dep").Position -= 1
            UpdateRecordNumber()
            bNewMode = False
        End If
    End Sub
    ' مقطع خاص بزر الأمر السجل الأول '
    Private Sub btnFirst_Click(ByVal sender As System.Object, ByVal e As _
            System.EventArgs) Handles btnFirst.Click
        Me.BindingContext(ds, "dep").Position = 0
        UpdateRecordNumber()
        bNewMode = False
    End Sub
    ' مقطع خاص بزر الأمر السجل الأخير '
    Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As _
    System.EventArgs) Handles btnLast.Click
        Me.BindingContext(ds, "dep").Position = Me.BindingContext(ds, "dep").Count - 1
        UpdateRecordNumber()
        bNewMode = False
    End Sub
```

```
    ' مقطع خاص بزر الأمر سجل جديد
    Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As _
              System.EventArgs) Handles btnNew.Click
        Me.BindingContext(ds, "dep").AddNew()
        txtRecord.Text = "سجل جديد"
        ClearForm()
        Me.txtdepid.Focus()
        bNewMode = True
    End Sub
     Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As _
               System.EventArgs) Handles btnUpdate.Click '  مقطع خاص بزر الأمر حفظ
         Dim sSQL As String
         Dim dbCmd As New OleDbCommand    '   كائن أمر
         Dim noRecToShow As Int32
         ' تحديد السجل الذي سيُعرض بعدما يتم التحديث بنجاح وفي هذه الحالة نفس السجل
         noRecToShow = Me.BindingContext(ds, "dep").Position
         conn.Open()
         If bNewMode Then
             ' في حالة سجل جديد
             bNewMode = False
             sSQL = "INSERT INTO dep (depid, depna)" & _
             " VALUES ('" & CLng(txtdepid.Text) & " ', '" & txtdepna.Text & "')"
             cmd.CommandText = sSQL    '  الخاصية نص الأمر
             Try
                 Me.BindingContext(ds, "dep").CancelCurrentEdit()
                 cmd.ExecuteNonQuery()    '   تنفيذ الأمر
                 RefreshData(noRecToShow)
                 UpdateRecordNumber()
             Catch ex As OleDbException
                 MsgBox("خطأ لم تتم عملية الحفظ: " & ex.Message)
             End Try
         Else
             '   التحديث في سجل موجود
             sSQL = "UPDATE dep SET depna='" & Me.txtdepna.Text & _
                   "' WHERE depid=" & txtdepid.Text
             cmd.CommandText = sSQL
             Try
                 cmd.ExecuteNonQuery()
                 RefreshData(noRecToShow)
                 UpdateRecordNumber()
             Catch ex As OleDbException
                 MsgBox(" خطأ لم يتم الحفظ: " & ex.Message)
             End Try
         End If
         conn.Close()
     End Sub
#End Region
End Class
```

ملاحظـــة: أتـــاح لنا الكود السابق التحرير والتحديث في اسم القسم في حالة السجل الجديد أو سجل موجود ولكن التحديث في رقم القســم غيــر ممكــن إلا في حالة السجل الجديد وذلك لأننا نحدّث بيانات السجل الموجود بالاعتماد على رقم القسم المخزّن في مجموعة البيانات Dataset.

التقارير Reports

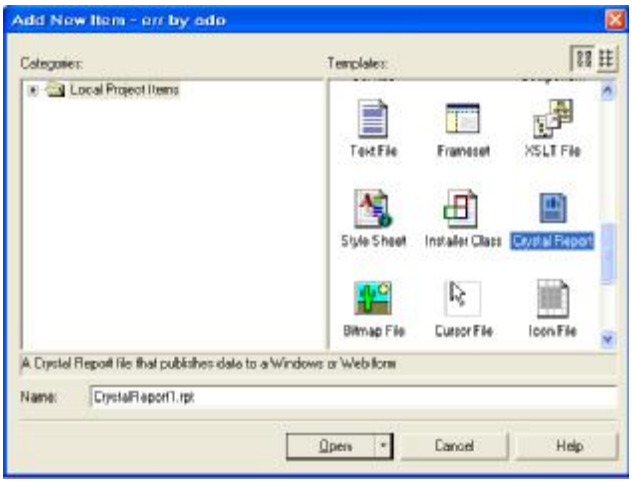
الأداة الرئيسية لتكوين وتنسيق التقارير في Visual Studio .NET هي Crystal Reports .NET ، حيث تحتوي هذه الأداة على مُنشئ التقارير والذي يقوم بعرض عدة واجهات تُسمى خبراء التقارير Report Experts لتصميم التقارير بسرعة. وتمتاز التقارير التي يُنشئها Crystal Reports .NET بإمكانية استخدامها في بيئة Windows وفي Web .

تساعدنا التقارير في تقديم البيانات بطريقة ذات معنى وإعدادها للطباعة على الآلة الطابعة، فيجب أن تكون هناك طابعة معرّفة بنظام التشغيل لكي نتمكن من تصميم التقرير. وليس بالضرورة أن يكون الجهاز مربوطاً بطابعة. (يُمكن إضافة طابعة عن طريق ابدأ> إعدادات > الطابعات ثم إضافة طابعة).

يحتوى تصميم التقرير على خمسة أجزاء أساسية هي :

1. رأس التقرير Report Header (Section1) : وهو يحتوى على ما يظهر في الجزء العلوي من أول صفحة فقط
2. رأس الصفحة Page Header (Section2) : يظهر في الجزء العلوي من كل صفحة وهو ينفع أن نضع فيه عناوين الأعمدة.
3. التفصيل Details (Section3) :لعرض السجلات.
4. تذييل التقرير Report Footer (Section4) : ما يظهر في الجزء السفلي من أخر صفحة
5. تذييل الصفحة Page Footer (Section5) : ما يظهر أسفل كل صفحة.

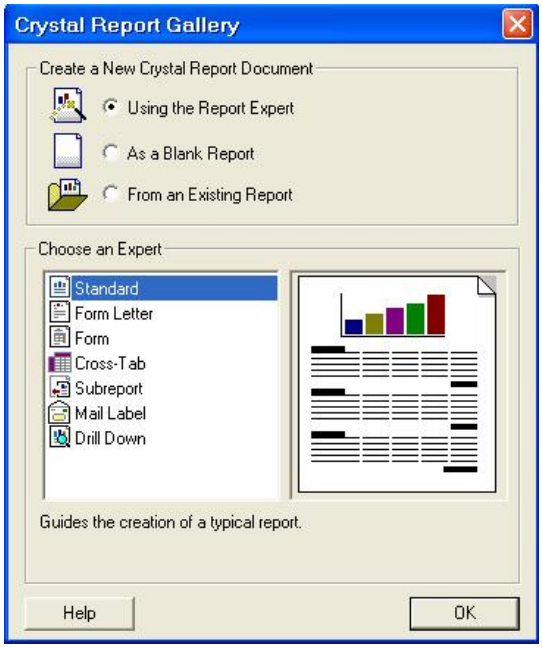
ولإضافة تقرير إلى المشروع نتّبع الخطوات التالية



1. من القائمة Project نختار الأمر Add Windows Form أو Add New Item فيظهر إطار إضافة بند جديد لتحديد نوع النموذج فنختار Crystal Report ، ويمكن الوصول إلى هذه المرحلة من خلال النقر بالأيمن على اسم الحل في إطار مستكشف الحل (Solution Explorer) ثم

ثم أو وكما سبق ذِكرهُ. وبعد التحديد سيُعطي البرنامج اسماً افتراضياً للتقرير بالامتداد rpt. ثم بالنقر على Open يبدأ عمل البرنامج Crystal Report

خطوة تسجيل البرنامج للحصول على ميزات أخرى وتحديث البرنامج فنختار Register Later لننتقل إلى الخطوة التالية لاختيار نوع التقرير



2. وأنواع التقارير المتاحة هي: Standard تقرير قياسي: وهـــي الأداة الأكثـــر استخداماً، وتحتوي على ثمانية علامات تبويب لاختيار مصدر البيانات والربط مع الجداول.

Form Letter نمـــوذج خطابـــات: لتكوين تقارير الخطابات (المراسلات)

Form نمـــوذج: لتكوين تقارير يُمكن طباعتها على نماذج موجودة مثل الفواتير كشوفات الحسابات

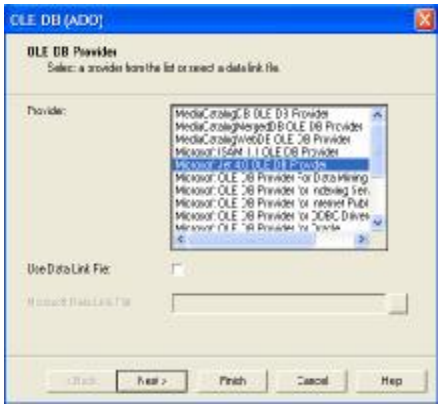
Cross-Tab تقرير جدولي: لتكوين تقارير تعرض البيانات بصـــورة مـــتقاطعة فقيم حقل أو أكثر كعناوين للصفوف وقيم حقـــل واحد فقط كعناوين أعمدة ومجاميع قيم حقل كقيم داخل الجدول

Subreport تقريـــر فرعي: لتكوين تقرير رئيسي وتقرير فرعي في نفس الوقت

Mail Label بطاقة العنونة: لإنشاء بطاقات وملصقات كعناوين المستعيرين، مواصفات المنتج للصقها على المنتَج.

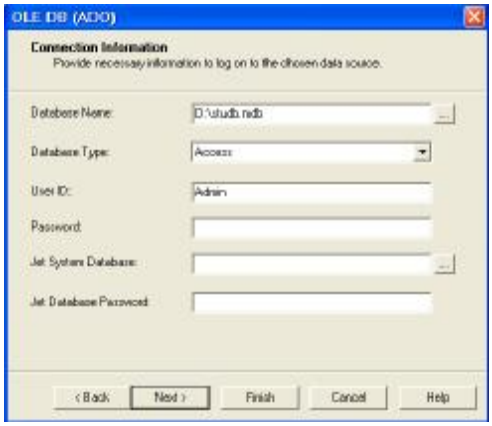
Drill Down تتبع التفصيلات: يقوم التقرير بإخفاء بعض الأقسام ويجعلها متاحة من خلال التنقيب Drill Down

3. والخطـــوة التي تلي اختيار نوع التقرير هي لتحديد مصدر البـــيانات ومعلومات عن التقرير فنختار في مثالنا هذا OLE DB (ADO) فيبدأ معالج الاتصال بالعمل لاختيار نوع موفّر البيانات

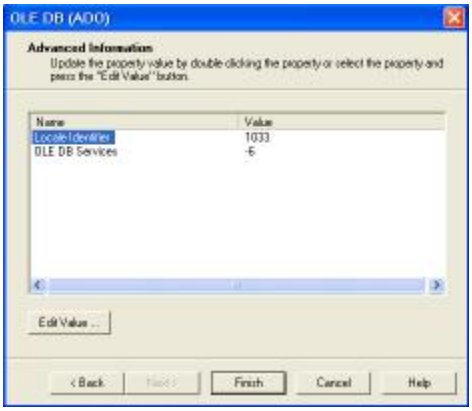


4. تحديد نوع موفّر البيانات Provider

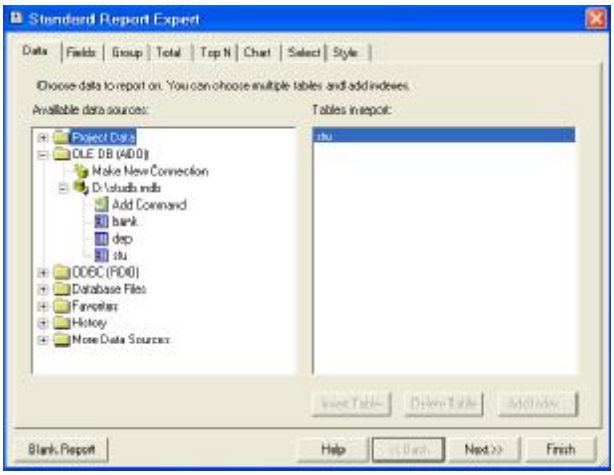
5. ثم تحديد اسم ومسار قاعدة البيانات



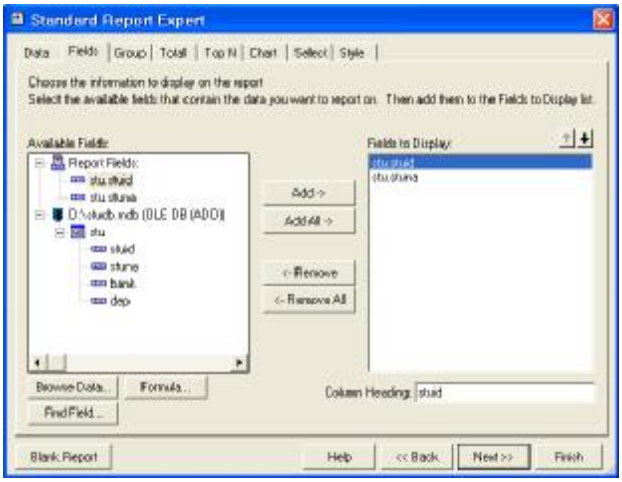
معلــومات متقدمة عن قيم بعض خصائص الاتصال، وبالنقر على Finish سنعود إلى خطوات خبير التقارير القياسية وكما في الخطوات التالية



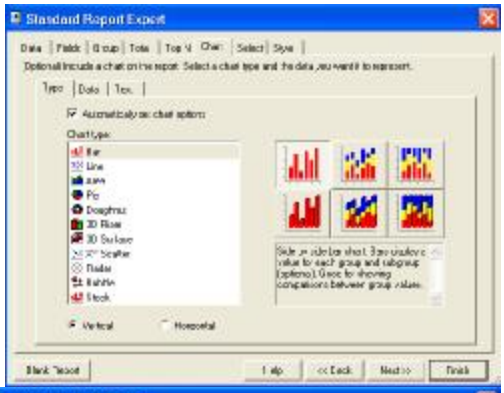
1. تحديد الجداول التي سيستند عليها التقرير



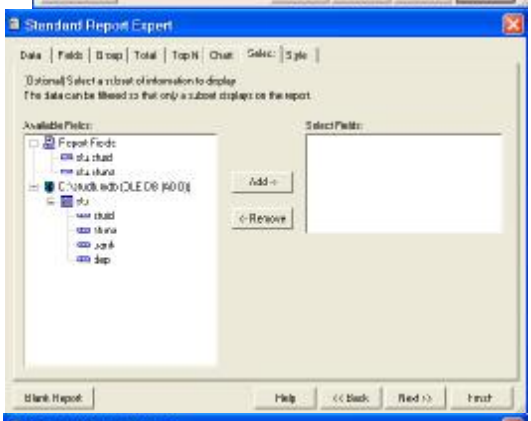
2. تحديد الحقول التي ستظهر في التقرير، كما يمكننا في هذه الخطــوة تغييــر رؤوس الأعمدة (عناوين الحقول) في الخانة Column Heading كمــا يُمكنــنا عمل ذلك فيما بعد بالنقر المــزدوج على مربع الحقل في تصميم التقرير ثم تغيير النص أو من إطار الخصائص. وفي الخطوة التي تليها (3)يتم تحديد حقــول التجمــيع إن رغبــنا بعمــل السـجلات المعروضة كمجمــوعات Group By. والخطوتين اللتان تليان (4 و 5) للمجموع والسجلات الخاصة بالحقل الذي تم التجميع عليه



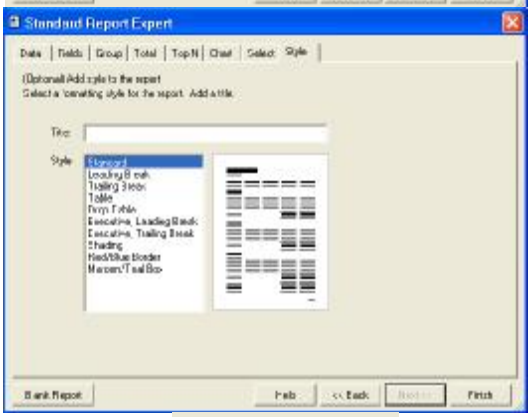
6. إذا رغبنا بتضمين التقرير مخططاً



7. لتحديد معايير (شروط) لتحديد السجلات التي ستُعرض



8. لكتابة عنوان للتقرير وتحديد نمط التقرير ثم Finish. بذلك سيتم تكوين التقرير في ملف بالامتداد rpt. والكود الخاص به في ملف بالامتداد vb. في نفس مجلّد الحل. ويظهر اسم التقرير في الإطار مستكشف الحل Solution Explorer



في تصميم النموذج الخاص بعرض التقرير نقوم بإضافة الأداة عارض التقرير CrystalReportViewer وهي موجودة ضمن علامة التبويب Windows Forms ومن أهم خصائص عنصر التحكم عارض التقرير هي مصدر التقرير ReportSource ، فنكتب في المقطع Form_Load السطر التالي

```
CrystalReportViewer1.ReportSource = New CrystalReport1
```

ثم نشغل الحل ليبدأ التشغيل بنفس النموذج الخاص بعرض التقرير فيظهر التقرير من خلال عنصر التحكم عارض التقرير الموجود في النموذج، ويجب حفظ المشروع قبل التنفيذ ليتم تحديث التغييرات التي نُجريها على تصميم التقرير.

لإنشاء مربع حقل رقم مسلسل ننقر بزر الماوس الأيمن على Running Total Fields في إطار Field Explorer ثم نختار ..New ونحدد حقل الرقم الدراسي ثم نختار الدالة Count

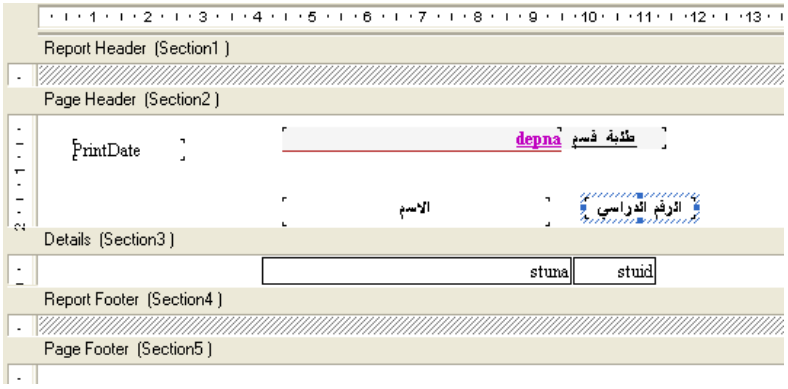
إذا كان الإطار Field Explorer مغلقاً فننقر على الأداة Toggle Field View في شريط أدوات 

لإظهاره. ولإزالة فواصل الآلاف نحذف الرمز من الخاصية ThousandSymbol لمربع الحقل stuid في تصميم التقرير.

ولإظهار حدود مربعات النصوص نستخدم الخصائص LineStyle (الأربعة) لمربعات النصوص.

مثال: عرض تقريرين (كل واحد على حدة) ولكن في نفس عارض التقارير من خلال زري أمر. وأحد التقريرين يُستخدم لعرض السجلات بَعد التصفية حسب معايير يحددها المستخدم وقت التنفيذ.

ننسخ قاعدة البيانات studb.mdb من مجلّد الأمثلة E:\3\db إلى مجلّد المشروع (داخل المجلّد Bin) ، ثم نبدأ بإنشاء تقرير جديد يســتند على جدول الطلبة stu واسم القسم من جدول الأقسام dep ، وفي الخطوة السابعة من معالج التقارير نحدد حقل رقم القسم من جدول الطلبة ليتم تصفية السجلات حسب قيمهُ



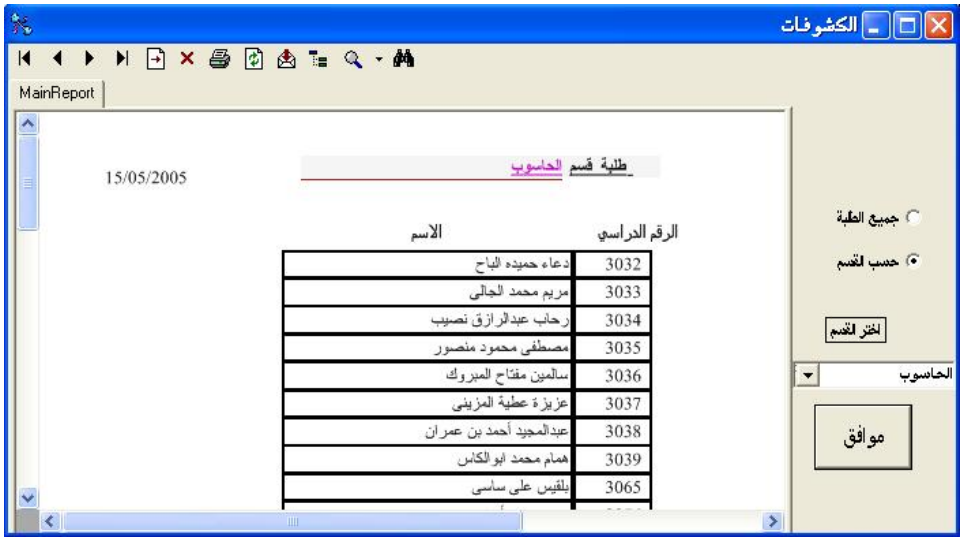
تصميم التقرير

ثم نصمم نموذجاً ونضع فيه زري راديو ومربع تحرير وسرد وزر أمر وعارض تقارير أضبط الخاصية Cursor لزر الأمر على Hand . وكود فصيلة النموذج كما يلي:

```
(General)                                   (Declarations)

Imports System.Data.OleDb

Public Class Form4
    Inherits System.Windows.Forms.Form

 Windows Form Designer generated code
    Dim sConn As String = "Provider=Microsoft.Jet.OLEDB.4.0;Password=;" & _
                          "Data Source=" & Application.StartupPath & "\studb.mdb"
    Dim strSELECT As String = "SELECT * FROM  stu ORDER BY stu.stuid"
    Dim conn As OleDbConnection      '   كائن الاتصال
    Dim cmd As OleDbCommand          '   كائن الأمر
    Dim da As OleDbDataAdapter       '   كائن محوّل البيانات
    Dim dsstu As DataSet             '   كائن مجموعة البيانات

    Private Sub Form4_Load(ByVal sender As System.Object, _
            ByVal e As System.EventArgs) Handles MyBase.Load
        Me.Show()
        Rad2.Focus()
        conn = New OleDbConnection(sConn)       '   كائن الاتصال
        cmd = New OleDbCommand                  '   كائن الأمر
        da = New OleDbDataAdapter               '   كائن محوّل البيانات
        ds = New DataSet                        '   كائن مجموعة البيانات
        cmd.CommandText = strSELECT
        cmd.Connection = conn
        da.SelectCommand = cmd
```

```
            conn.Open()
            da.Fill(ds, "stu")
            Dim strSql As String
            strSql = "SELECT  *  FROM  dep"    '  جدول الأقسام
            Dim dacbo As New OleDbDataAdapter(strSql, conn)
            Dim dscbo As New DataSet
            dacbo.Fill(dscbo, "dep")
            conn.Close()
            '   ربط مربع التحرير والسرد بالحقل اسم القسم للعرض  والرقم للمعايير
            Me.cbodep.DisplayMember = "dep.depna"
            Me.cbodep.ValueMember = "dep.depid"
            Me.cbodep.DataSource = dscbo
        End Sub
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    Dim dsrpt As New DataSet
    '  كود مقطع زر الأمر معاينة
    If Rad1.Checked Then  '  في حالة جميع الطلبة
        CRV.ReportSource = New CrystalReport3 ' عرض التقرير الخاص بجميع الطلبة
    Else    '   حالة الطلبة حسب القسم
            'إعلان عن كائن منيل ومتغير سلسلة بمستوى الكود
            Dim rpt As New CrystalReport4, cri As String
            cri = "{stu.dep}=" & cbodep.SelectedValue
            rpt.SetDataSource(dsstu)    ' مصدر التقرير هو مجموعة البيانات
            CRV.SelectionFormula = cri    '  تصفية السجلات حسب المعيار
            CRV.DisplayBackgroundEdge = False
            CRV.RightToLeft = RightToLeft.Yes
            CRV.ReportSource = rpt
        End If
    End Sub
    Private Sub Rad1_CheckedChanged(ByVal sender As System.Object, _
          ByVal e As System.EventArgs) Handles Rad1.CheckedChanged
        If Rad1.Checked Then  '  إذا اختار المستخدم جميع الطلبة
            Me.lbldep.Visible = False
            cbodep.Visible = False
        End If
    End Sub
    Private Sub Rad2_CheckedChanged(ByVal sender As System.Object, _
          ByVal e As System.EventArgs) Handles Rad2.CheckedChanged
        If Rad2.Checked Then   '  إذا اختار المستخدم زر الراديو حسب القسم
            Me.lbldep.Visible = True
            cbodep.Visible = True
        End If
    End Sub
End Class
```

ونضـــبط الخصائص LineStyle (الأربعة) لمربعي النص في التقرير على crLSSingleLine ، وعند التنفيذ سيظهر التقرير كما في الشكل التالي:



أما التقرير الآخر والذي يعرض جميع الطلبة فكما في المثال السابق.

تمرين:

1. صـــمم تقريـــراً يعرض الطلبة حسب القسم والفصل حيث يختارهما المستخدم من مربعي تحرير وسرد، الحقول المطلوبة الرقم الدراســـي والاســـم ويظهـــر في رأس الصفحة اسم القسم والفصل وقم بإدراج شعار المركز كصورة في رأس الصفحة وإظهار التاريخ الحالي.
2. أضف للنموذج في التمرين السابق أداة قائمة مختصرة ContextMenu تحتوي على بندين عند النقر بالأيمن على عارض التقارير تظهر القائمة واكتب الكود المناسب لكل بند وكما في الشكل التالي:

تلميح (Hint) اضبط الخاصية ContextMenu ContextMenu1 لعارض التقرير.