

ربط الفيچوال بيسك VB مع قواعد البيانات



المحتويات

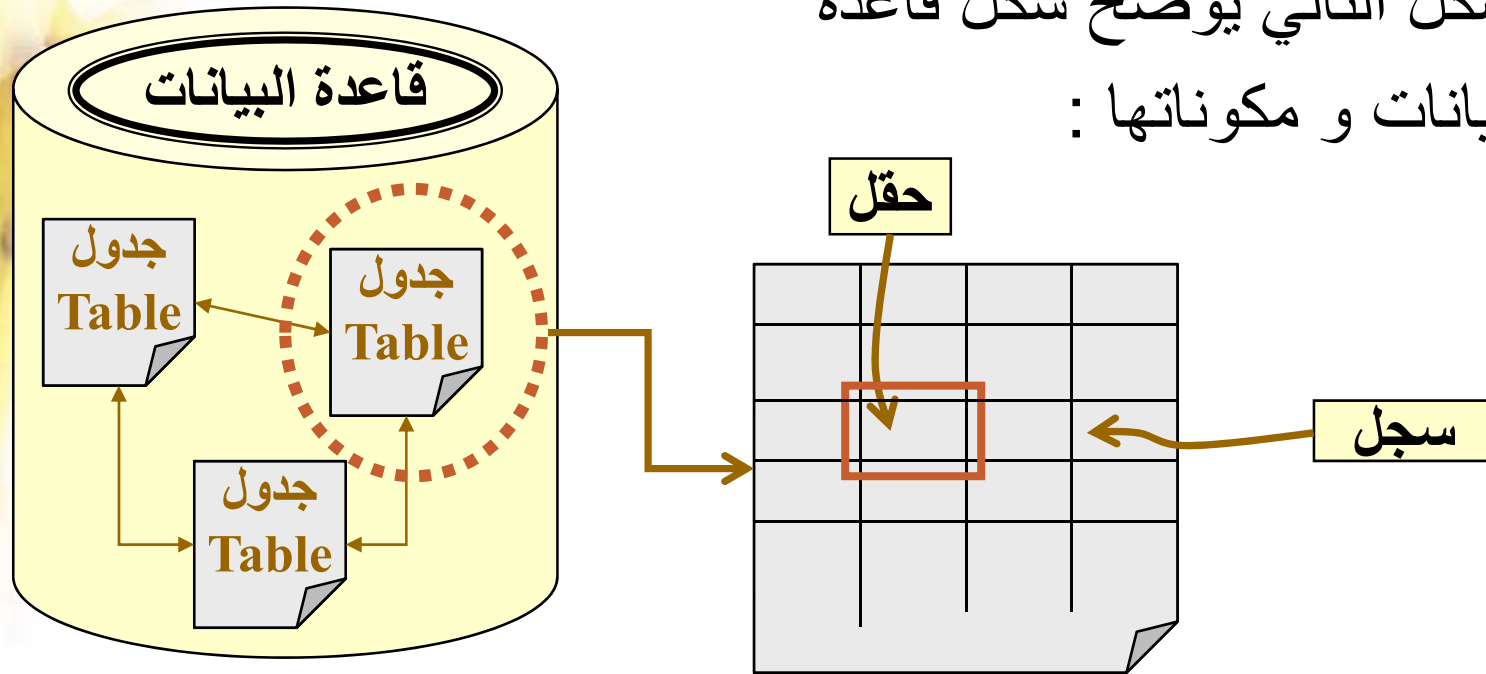
- ما هي قواعد البيانات ..
- المصطلحات جدول ، سجل و حقل .. ما هو معناها؟؟
- ما هي نظم ادارة البيانات DBMS و ما هي مكوناتها ...
- ما هو الـ Microsoft Database Jet ، و ما علاقته

بـ VB !!!

- انشاء قواعد البيانات بـ Visual Data Manager
- تقنيات الوصول الى قواعد البيانات
 - بالتفصيل الممل Data Control
 - بالتفصيل الممل ActiveX Data Objects (ADO)
 - انشاء التقارير باستخدام Data Report

ما هي قواعد البيانات ؟

- قواعد البيانات هي أي مجموعة من الجداول المنظمة التي تجمعها علاقة فيما بينها ...
- مثال على ذلك : اسماء الطلاب و درجاتهم الدراسية و بيانات المواد و المدرسين ، الخ ..
- الشكل التالي يوضح شكل قاعدة البيانات و مكوناتها :

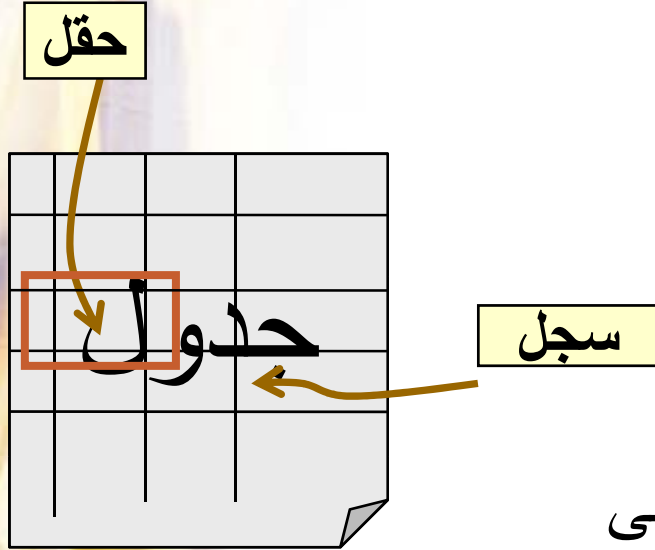


ما هي الجداول ؟ ما هي DBMS ؟

• الجدول عبارة عن مجموعة من الصفوف و الأعمدة ..

الصفوف تسمى سجلات ..

و تقاطع الصف مع العمود يسمى حقل ..



• نظام ادارة قاعدة البيانات **DBMS**

هو نظام يدير كافة العمليات التي تجرى على

قاعدة البيانات مثل عملية انشاء الجداول او التعديل عليها او

حتى حذفها من قاعدة البيانات ، و ايضاً عملية بناء العلاقات بين

جداول قاعدة البيانات .

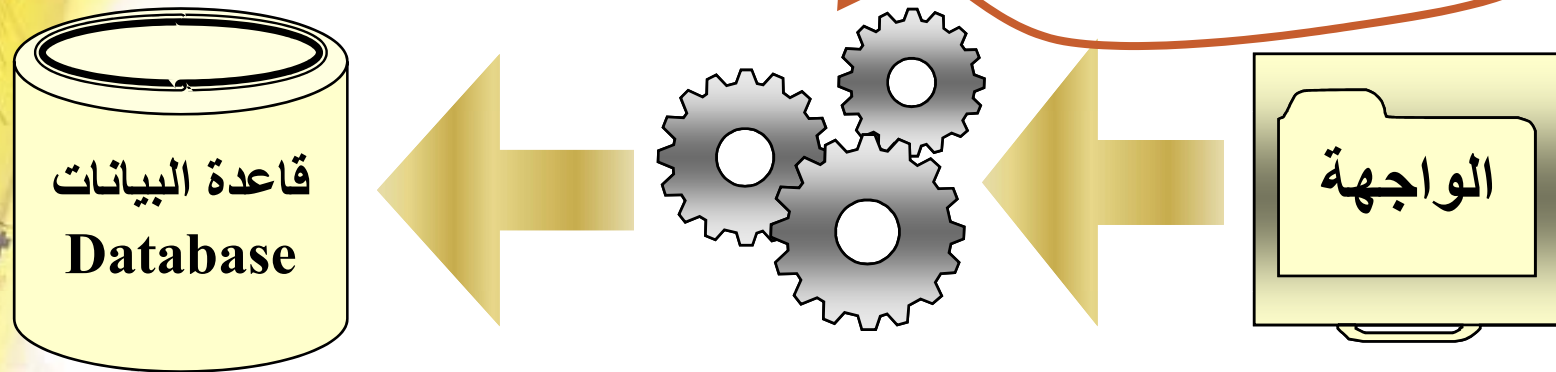
• مثال على انظمة ادارة قواعد البيانات **DBMS** : Access و

Oracle و MS-SQL Server و FoxPro و غيرهم الكثير ..

مكونات نظم ادارة قواعد البيانات

- يتكون نظام ادارة قاعدة البيانات من جزئيين هما :
 1. **الواجهة *User Interface*** التي يتعامل من خلالها مع المستخدم ..

2. **محرك قاعدة البيانات *Database Engine*** و هو عبارة عن الآلية التي من شأنها تنظيم التعامل مع قاعدة البيانات (من إضافة و حذف و تعديل ... الخ) و ذلك في سبيل ضمان تكامل بيانات قاعدة البيانات و سريتها (ما معني تكامل **Integrity** البيانات ؟) .

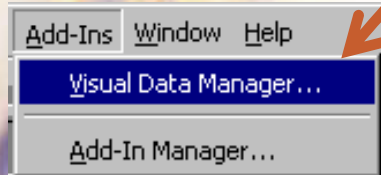


• Microsoft Database Jet هو احد الامثلة على محركات قاعدة البيانات و الذي يوفر الوسائل الأولية للفيجوال بيسك لكي يتعامل مع قواعد البيانات المختلفة (Access , Oracle .. الخ)

• لم يكن Microsoft Database Jet جزءاً من VB كما هو الآن (و ذلك حتى الاصدارة الثالثة من VB) مما استدعى استخدام برنامج الاتصال المفتوح لقواعد البيانات ODBC داخل بيئة الفيجوال بيسك و الذي يعتبر احد البرامج التي يقدمها ويندوز ، و يحتوى هذا البرنامج على مشغلات Drivers ، حيث ان كل مشغل يخص نوع معين من قواعد البيانات ، مثال على ذلك Microsoft Access Driver الذي يتيح التعامل مع قواعد البيانات المبنية بواسطة برنامج Access المعروف .

انشاء قواعد البيانات بـ Visual Data Manager

- كما هو ممكن انشاء قاعدة البيانات داخل برنامج الـ Access نفسه فإنه من الممكن ايضاً انشاء أي نوع من انواع قواعد البيانات داخل بيئة خاصة داخل VB تسمى Visual Data Manager، و ذلك كما يلي:

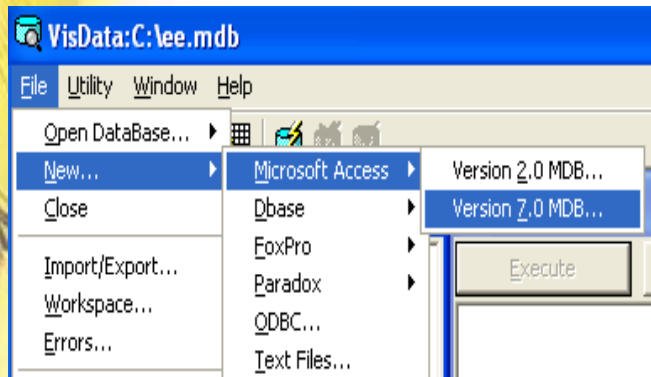


1. الخطوة الأولى فتح الفيچوال بيسك كالمعتاد و من ثم

الذهاب إلى قائمة visual data manager >> Add-ins...

2. بعد ذلك ستفتح لك نافذة البرنامج اذهب

إلى File>>new>>Microsoft access>>version 7.0 ..mdb



سيظهر لك مربع حفظ لتحديد الموقع الذي

تريد أن تحفظ فيه قاعدة بياناتك ..

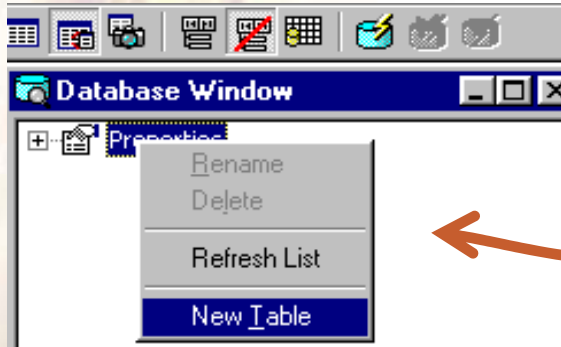
بعدها اختر حفظ

تابع

3. بعد ذلك ستظهر لك قاعدة البيانات ، حدد

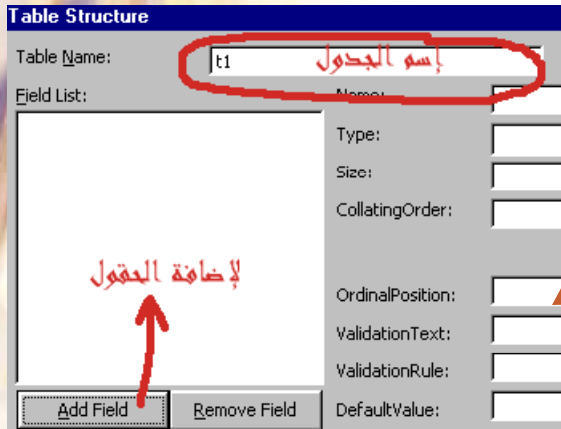
الخصائص **Properties** بالزر الأيمن

ثم اختر **New Table** من اجل انشاء جدول .



4. بعد ذلك ستفتح لك نافذة تكتب فيها أسم الجدول

و ثم أضف حقول الجدول الحقول كما هو مبين .

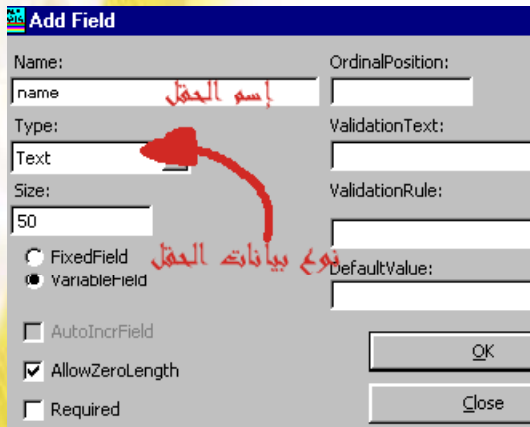


5. يتفتح لك نافذة إضافة الحقول ، اكتب أسم الحقل

في المكان المخصص و حدد نوع بيانات الحقل

أي هل هي رقمية أو حرفية ... الخ ، ثم كرر العملية

حتى تنهي جميع الحقول التي تريدها .



تقنيات الوصول الى قواعد البيانات

- التقنية **Data Access Objects (DAO)** : توفر هذه التقنية مجموعة من الكائنات (Objects) للتخاطب مع قاعدة البيانات، وإرسال الأوامر لها. وهي مصممة بشكل رئيس للتعامل مع قواعد بيانات أكسس (الى الاصدار 97) ، و يمكنك استخدامها في التعامل مع قواعد بيانات أخرى من خلال المحرك Jet الذي توفره.

- اداة التحكم في البيانات **Data Control** : و هي تقنية سهلة للتعامل مع قاعدة البيانات و هي احدى ادوات التحكم الموجودة في صندوق الأدوات القياسي ، و هي تمكننا من الاتصال بمجموعة من السجلات **Recordset** في قاعدة بيانات Jet للتعامل معها و كذلك ربطها بعدة ادوات في فيجوال بيسك مثل اداة **Textbox**..



تابع

- **تقنية Remote Data Objects (RDO) :** و طوّرت هذه التقنية بشكل رئيس لتوفر طبقة مُبسّطة لتقنية ODBC لمبرمجي لغة فيجوال بيسك.
- **تقنية ActiveX Data Objects (ADO) :** واستكمالاً لمسيرة التقدم والتطوير طرحت شركة مايكروسوفت بعدها نظام ADO، والتي قامت بتسهيل عملية التعامل مع قواعد البيانات بشكل كبير، وأصبحت الشكل النهائي للوصول لقاعدة البيانات ليس في لغة VB و حسب ، بل و في العديد من لغات البرمجة .

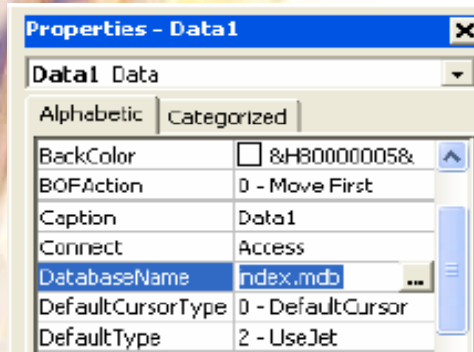
الربط بالأداة Data Control

- قبل عملية الربط يجب علينا أولاً تصميم قاعدة البيانات ، و يجب ان تكون قاعدة البيانات من نوع Access 97 على اقصى تقدير .
- خطوات الربط ستكون على النحو الآتي :

1. يتم انزال الاداة Data Control من صندوق الادوات الى الواجهة



Form ، و سيعطى لها الاسم Data1.



2. من خلال خصائص الاداة Data Control

يتم تحديد قاعدة البيانات المراد الاتصال بها

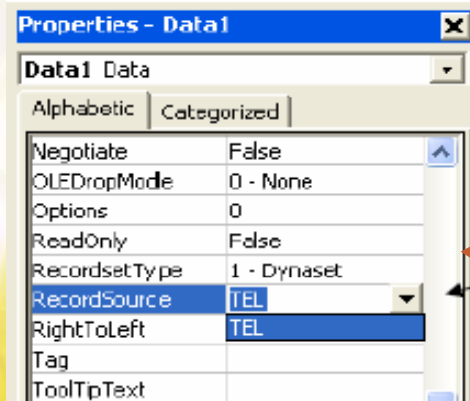
عن طريق الخاصية DatabaseName .

3. بعد ذلك يتم تحديد الجدول المراد الاتصال

به عن طريق الخاصية RecordSource

و التي معناها السجلات التي اود استرجاعها

من قاعدة البيانات ..



ربط مربعات النص Textboxes بحقول قاعدة البيانات

- يتم ربط قاعدة البيانات بمربعات النص Textbox عن طريق الاداة Data Control من اجل عرض حقول البيانات فيها ، و يتم ذلك بإتباع الخطوات التالية :

1. اختر الخاصية DataSource من خصائص

مربع النص المراد عرض حقل من البيانات

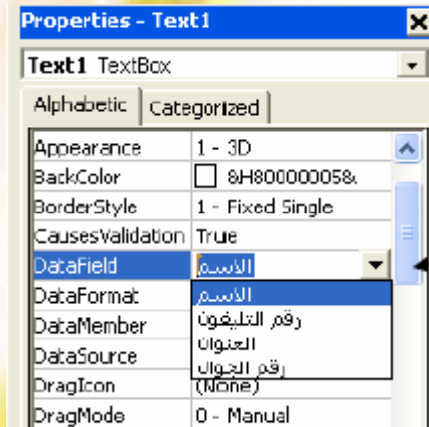
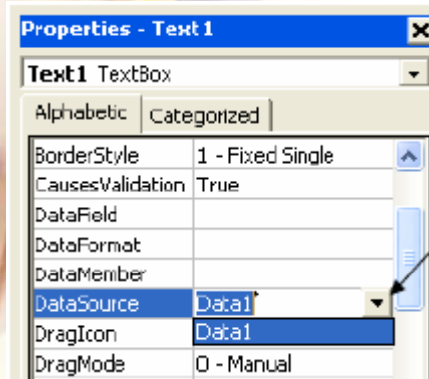
من خلاله و ضع فيه اسم الـ Data Control

التي تصلنا بقاعدة البيانات (و التي سيكون اسمها

في هذا السياق Data1) .

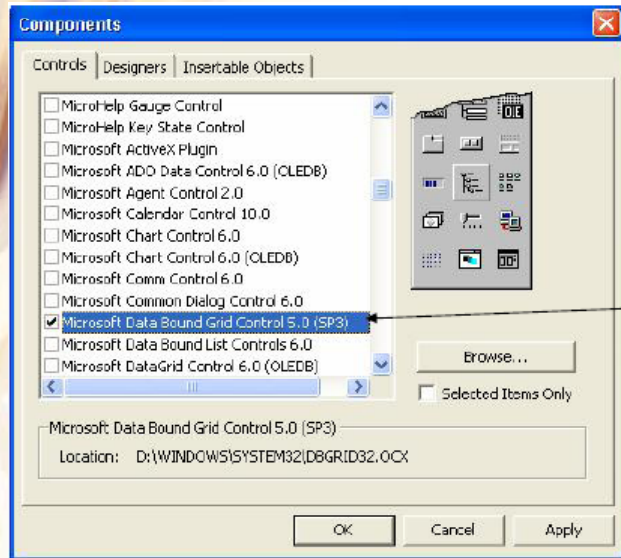
2. غير الخاصية DataField من خصائص مربع النص

الى اسم الحقل المراد عرضه .



عرض البيانات بـ Data Grid

- من الممكن كذلك عرض البيانات داخل اداة خاصة تعرف باسم Data Bound Grid و التي تتميز بعرض بيانات جدول ما في قاعدة البيانات عل شكل جدول و ذلك بحسب الخطوات التالية :

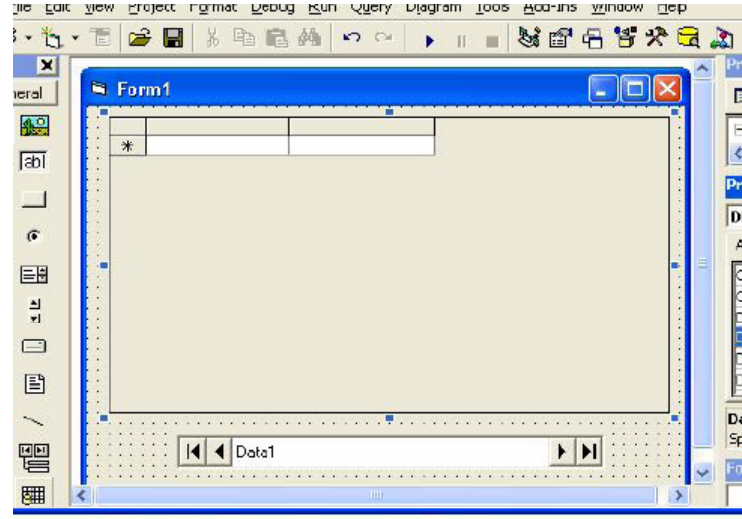


1. من القائمة Project نختار الامر Component ، و الذي سيظهر مجموعة الادوات الاضافية التي يوفرها VB ، و نختار منها الاداة Data Bound Grid و بعدها نضغط موافق .



2. من مربع الادوات نختار الاداة الجديدة .

3. تظهر لنا الاداة على الواجهة Form كما في الشكل التالي .



4. الآن من خصائص الاداة Data Grid نحدد في الخاصية DataSource اسم الـ Data Control التي تصلنا بقاعدة البيانات ، و التي سيكون اسمها Data1 ، و ذلك لأن Data1 سوف تحوي مسار قاعدة البيانات المطلوبة و الجدول المطلوب .

العمليات الرئيسية على سجلات قاعدة البيانات

- إنشاء سجل جديد في قاعدة البيانات :

Data1.RecordSet.Addnew

حيث ان :

Data1 : هو اسم الـ Data Control التي تصلنا بقاعدة البيانات المطلوبة ..

RecordSet : يقصد به مجموعة السجلات (التابعة للجدول المراد إضافة سجل اليه) و التي ترتبط بها الأداة Data1 .

- تعديل بيانات السجل الحالي :

Data1.Recordset.Edit

حيث يقوم هذا الامر بحفظ التعديلات التي تجرى على السجل الحالي .

العمليات الرئيسية على سجلات قاعدة البيانات

- حفظ البيانات (او التعديلات) في قاعدة البيانات :

Data1.RecordSet.Update

و هي تجرى بعد عملية الاضافة او التعديل (فقط) من اجل حفظ السجل الجديد داخل قاعدة البيانات .

- حذف سجل من قاعدة البيانات :

Data1.Recordset.Delete

Data1.Refresh

حيث ان هذا الامر يقوم بحذف السجل الحالي ، و الخاصية Refresh تقوم تحديث بيانات قاعدة البيانات بعد عملية الحذف .

العمليات الرئيسية على سجلات قاعدة البيانات

• اوامر التنقل بين السجلات :

Data1.Recordset.MoveFirst ← الانتقال الى السجل الاول

Data1.Recordset.MoveLast ← الانتقال الى السجل الأخير

Data1.Recordset.MoveNext ← الانتقال الى السجل التالي

Data1.Recordset.MovePrevious ← الانتقال الى السجل السابق

Data1.Recordset.Move (#no) ← المرور على السجلات
بحسب قيمة #no

العمليات الرئيسية على سجلات قاعدة البيانات

- اوامر البحث بين السجلات :

Data1.Recordset.(طريقة البحث) " col_name = 'المطلوب' "

حيث ان col_name هو اسم العمود (في الجدول) المطلوب البحث بين حقوله
عن القيمة المطلوبة ... و طرق البحث كما يلي :

Data1.Recordset.FindFirst ←

اول سجل يطابق المطلوب من البحث

Data1.Recordset.FindLast ←

اخر سجل يطابق المطلوب من البحث

Data1.Recordset.FindNext ←

السجل التالي الذي يطابق المطلوب من البحث

Data1.Recordset.FindPrevious ←

السجل السابق الذي يطابق المطلوب من البحث

العمليات الرئيسية على سجلات قاعدة البيانات

- امر الوصول الى عمود في الجدول :

Data1.RecordSet![اسم العمود]

مثال :

Data1.RecordSet![*name*]

- الدالة EOF (Data1. Recordset. EOF) تعيد القيمة True اذا كان المؤشر عند اخر سجل في الجدول .

- الدالة BOF (Data1. Recordset. BOF) تعيد القيمة True اذا كان المؤشر عند اول سجل في الجدول .

- الدالة Exclusive تحدد ما اذا كان الآخرون يستطيعون التعامل مع قاعدة البيانات اثناء استخدام التطبيق لها (Data1.Exclusive)

تطبيقات عملية

- التطبيق الاول : لإضافة سجل الى جدول (البيانات موجودة في مربعات نص غير مرتبطة بقاعدة البيانات !!) .

```
Data1. Recordset. AddNew
```

```
Data1. Recordset![name] = Txt1.Text
```

```
Data1. Recordset![address] = Txt2.Text
```

```
Data1. Recordset![phone] = Txt3.Text
```

```
Data1. Recordset. Update
```

تطبيقات عملية

- التطبيق الثاني : الانتقال الى السجل التالي و عرض البيانات في مربعات نص غير مرتبطة بقاعدة البيانات .

```
If Data1. Recordset. EOF Then
```

```
    MsgBox " هذا آخر سجل "
```

```
Else
```

```
    Data1. Recordset. MoveNext
```

```
    Txt1. Text = Data1. Recordset![name]
```

```
    Txt2. Text = Data1. Recordset![address]
```

```
    Txt3. Text = Data1. Recordset![phone]
```

تطبيقات عملية

- التطبيق الثالث : البحث عن سجل معين بحسب اسم الشخص :

```
Dim name,str As String
```

```
name = InputBox(" أدخل الاسم المطلوب ", " بحث ")
```

```
If Len( name) = 0 Then
```

```
    MsgBox " لم تقم بإدخال أي بيانات "
```

```
    Exit Sub
```

```
End If
```

```
str= "name=" + name + " " ' str = (name = ' الاسم المطلوب ' )
```

```
Data1.Recordset.FindFirst str
```

```
If Data1.Recordset.NoMatch = True Then
```

```
    MsgBox " هذا الاسم غير موجود "
```

```
Else
```

```
    Txt1. Text = Data1. Recordset![name]
```

```
    Txt2. Text = Data1. Recordset![ address]
```

```
    Txt3. Text = Data1. Recordset![ phone]
```

```
End If
```

يقوم هذا الأمر بالبحث عن
اول سجل يطابق البحث ،
فإن وجده يقف المؤشر عنده

يقوم هذا الأمر بإعادة True
في حالة انه لم يجد أي سجل
مطابق للبحث

الربط بالأداة ADODC

• الأداة ADODC هي اختصار لـ ADO Data Control ، حيث تعتبر ADO من التقنيات الحديثة المهمة للارتباط بقواعد البيانات و التي يتم استخدامها ليس فحسب في لغة VB و لكن في العديد من لغة البرمجة . طريقة التعامل مع الأداة ADODC لا تختلف عن طريقة التعامل مع الأداة Data Control غير انها تتيح التعامل مع قواعد البيانات الحديثة من قواعد بيانات Access على عكس الأداة Data Control التي لا تتعامل مع قواعد بيانات احدث من Access97.

طريقة اضافة الأداة ADODC إلى المشروع كما يلي :

1. نختار الامر Component بالضغط على الزر الأيمن

للفأرة في مربع الادوات من اجل اختيار

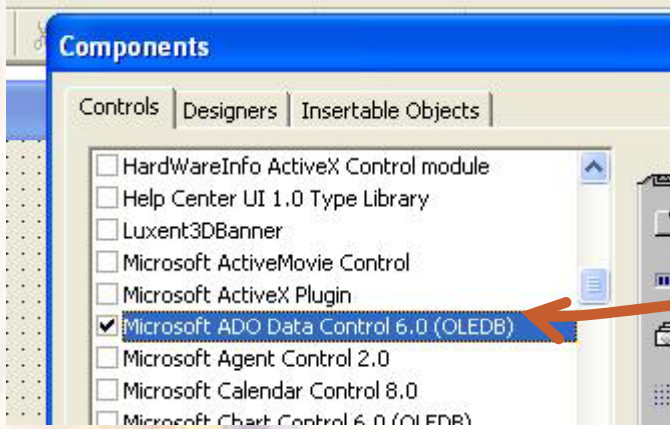
الاداة من مربع الحوار Component



2. من مربع الحوار Component

نقوم باختيار الاداة :

MICROSOFT ADO DATA CONTROL 6.0 (OLEDB)



3. بعد الضغط على زر موافق ، تتم إضافة الاداة

إلى ملف المشروع .



4. و اخيراً ، نقوم بإضافة الاداة إلى الـ Form.

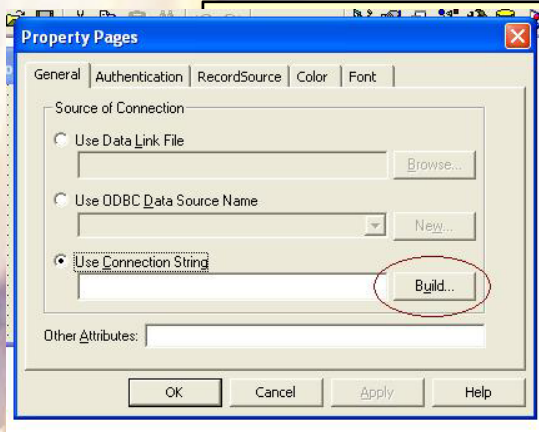


و يبقى معنا خطوات ربط الاداة ADODC بقاعدة

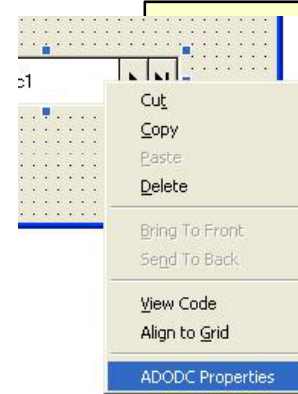
البيانات المراد الارتباط بها ..

خطوات ربط الاداة ADOBC بقاعدة البيانات

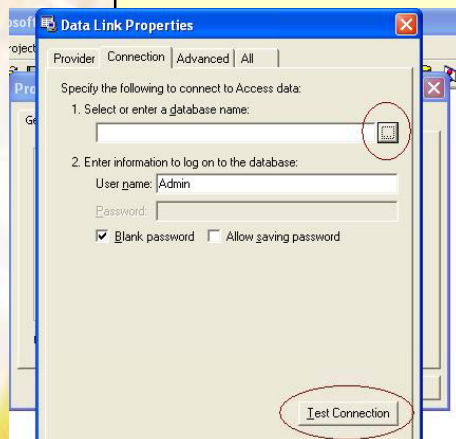
بعد ان تمت عملية اضافة الاداة **ADODC** إلى المشروع نقوم بربطها بقاعدة البيانات المطلوب الارتباط بها .



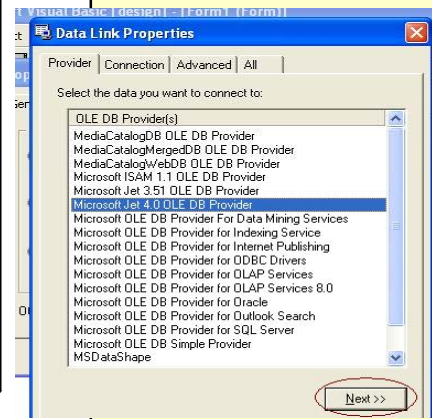
ثانياً : يتم الضغط على زر **Build** و الذي يبدأ سلسلة خطوات الارتباط بقاعدة البيانات.



اولاً : يتم ضبط خصائص الأداة بالدخول إلى خصائص الأداة من القائمة المنسدلة.

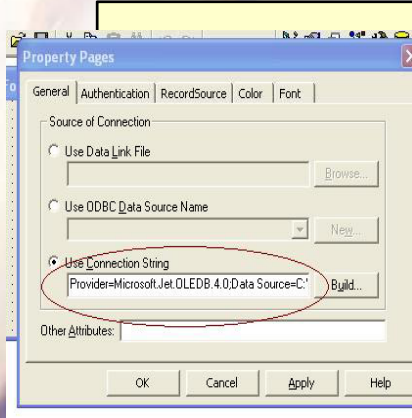


رابعاً : نحدد مسار قاعدة البيانات **Access** التي سيتم الارتباط بها ، و بعد ذلك نضغط على زر اختبار الاتصال للتأكد ان الاتصال قد تم .

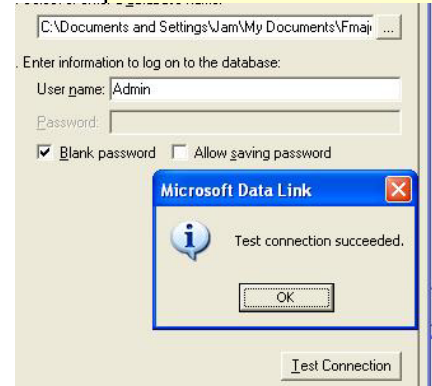


ثالثاً : في بداية الاتصال بقاعدة البيانات نحدد **المزود (المحرك) المراد** الاتصال به .. و هنا نحدد الاتصال بـ **Access** .

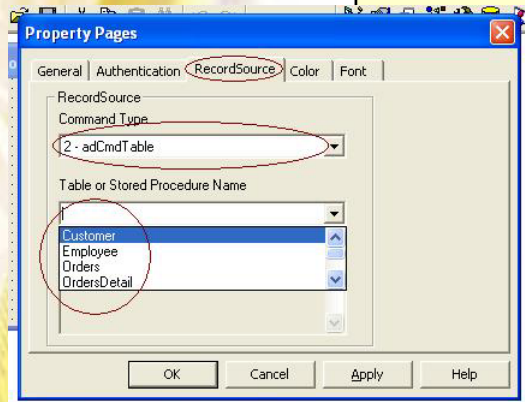
تابع خطوات ربط الاداة ADODC بقاعدة البيانات



سادساً : بعد ذلك سنعود إلى نافذة الخصائص من جديد ، و بها تفاصيل الارتباط ، بعدها نختار التبويب **RecordSource**



خامساً : في حالة ان الاتصال تم بنجاح ، سوف تظهر لنا النافذة الظاهرة في الشكل ، فنضغط على زر موافق.



اخيراً ، حدد ان نوع الاوامر **Command Type** سيكون على الجداول من القائمة المنسدلة الاولى ، ثم حدد الجدول الذي سيتم الارتباط به ، و من الممكن استخدام **جمل SQL** من اجل استرجاع السجلات التي سيتم الارتباط بها باستخدام **جملة Select** مما يتيح لنا التعامل مع اكثر من جدول في نفس الوقت (كيف؟).

عرض البيانات بـ Text BOXES و DataGridView

- يتم عرض بيانات حقول الجدول الذي تم الارتباط به بالأداة ADODC باستخدام مربعات النص بنفس الطريقة التي تم ذكرها عند ربط مربعات النص بالأداة Data Control (الخاصية Data Source لتحديد اداة الربط ADODC و الخاصية Data Filed لتحديد الحقل).

- من الممكن كذلك عرض البيانات داخل اداة خاصة تعرف باسم DataGridView (ليست الاداة Data Bound Grid التي تم استخدامها مع الاداة Data Control و لكنها تربط بنفس الطريقة و ذلك بتغير الخاصية Data Source إلى اداة الربط ADODC1) و هي كذلك تتميز بعرض بيانات جدول ما في قاعدة البيانات عل شكل جدول ، و يتم اختيارها من Component بإضافة الاداة Microsoft DataGridView



العمليات الرئيسية على الأداة ADODC

- إنشاء سجل جديد في قاعدة البيانات :

ADODC1.RecordSet.Addnew

حيث ان :

ADODC1 : هو اسم الـ ADODC التي تصلنا بقاعدة البيانات المطلوبة ..

RecordSet : يقصد به مجموعة السجلات (التابعة للجدول المراد إضافة سجل اليه) و التي ترتبط بها الأداة ADODC1 .

العمليات الرئيسية على الأداة ADODC

- حفظ البيانات (او التعديلات) في قاعدة البيانات :

ADODC1.RecordSet.Update

و هي تجرى بعد عملية الاضافة او التعديل (فقط) من اجل حفظ السجل الجديد داخل قاعدة البيانات .

- حذف سجل من قاعدة البيانات :

ADODC1.Recordset.Delete

ADODC1.Refresh

حيث ان هذا الامر يقوم بحذف السجل الحالي ، و الخاصية Refresh تقوم تحديث بيانات قاعدة البيانات بعد عملية الحذف .

العمليات الرئيسية على الأداة ADODC

• اوامر التنقل بين السجلات :

ADODC1.Recordset.MoveFirst ← الانتقال الى السجل الاول

ADODC1.Recordset.MoveLast ← الانتقال الى السجل الأخير

ADODC1.Recordset.MoveNext ← الانتقال الى السجل التالي

ADODC1.Recordset.MovePrevious ← الانتقال الى السجل السابق

ADODC1.Recordset.Move (#no) ← المرور على السجلات بحسب قيمة #no

العمليات الرئيسية على الأداة ADODC

- امر البحث بين السجلات هو الامر Find و الذي يحتوي على اربع وسائط ، يعتبر اولها الزامي و البقية اختياري :

ADODC1.Recordset.Find 1 , 2 , 3 , 4



1/ اسم العمود و
الحقل المراد البحث
عنه :
"Name= 'Ali' "

2/ نقطة بداية البحث
و توضع 0 افتراضياً
او يتم البدء في البحث
من النقطة المتوقف
عندها المؤشر حالياً

3/ اتجاه البحث و هو
احد خيارين :
adSearchBackward
أي وصولاً إلى اول سجل
adSearchForward
أي وصولاً إلى اخر سجل

4/ نقطة بداية البحث
و لكننا نحدد هنا
من أي صف سوف
يبدأ البحث

مثال :

Adodc1.Recordset.Find " Name = 'Ali' ", 0, adSearchForward, 1

العمليات الرئيسية على الأداة ADODC

- امر الوصول الى الحقل المشار اليه حالياً من عمود في الجدول :
ADODC1.RecordSet![اسم العمود]

مثال :

ADODC1.RecordSet![*name*]

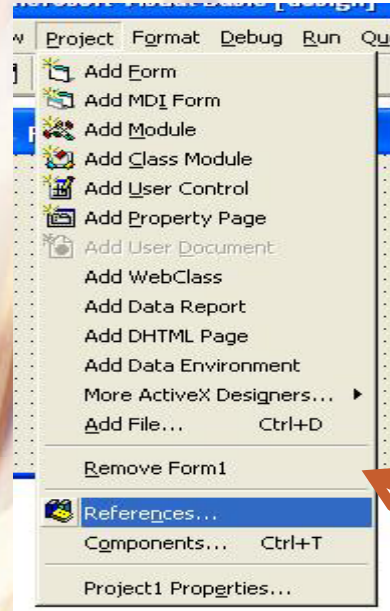
- الدالة EOF (ADODC1. Recordset. EOF) تعيد القيمة True اذا كان المؤشر عند اخر سجل في الجدول .
- الدالة BOF (ADODC1. Recordset. BOF) تعيد القيمة True اذا كان المؤشر عند اول سجل في الجدول .

الاتصال بقاعدة البيانات عن طريق المكتبة ADO

- تعلمنا في الموضوع السابق كيفية الاتصال و التعامل مع قاعدة البيانات باستخدام الأداة ADODC ، و في هذا الدرس سوف نتعلم – إن شاء الله – كيفية الاتصال و التعامل مع قاعدة البيانات مع خلال المكتبة ADO التي يوفرها لنا Visual Basic .
- المكتبة ADO تعتبر الأسلوب الجديد - قبل الأداة الجديدة ADO.NET - للتعامل مع قاعد البيانات ، فمثلا يمكننا الاتصال وقراءة محتويات قاعدة بيانات اكسس Access بدون الحاجة لوجود Access فهذه المكتبة توفر خدمة الاتصال والإجراءات المختلفة للتعامل مع قاعدة البيانات .

الاتصال بقاعدة البيانات عن طريق المكتبة ADO

خطوات اضافة المكتبة إلى ملف المشروع كما يلي :



1. من خلال القائمة Project نختار References

فتظهر لنا النافذة References - Project1 التي تحتوي على مكتبات عديدة ، ما يهمنا منها

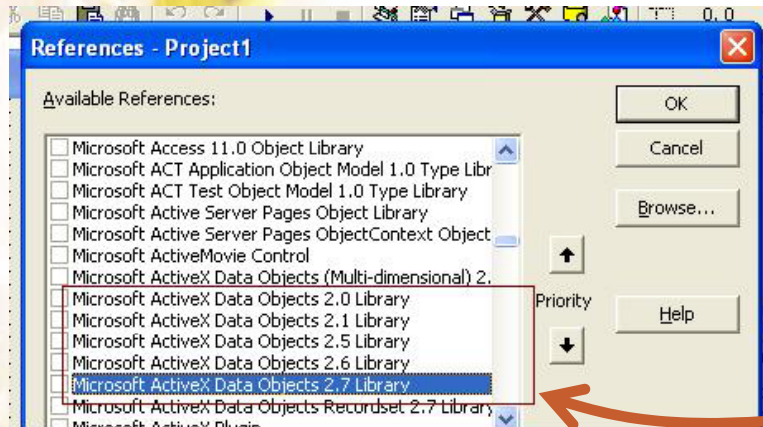
هو مكتبة ADO .

2. نقوم باختيار المكتبة :

Microsoft ActiveX Data Objects 2.X Library

حيث ان ال- X يعبر عن الإصدار المثبت على جهازك ، فأختر احد هذه الاصدارات و

اضغط موافق ليتم اضافة المكتبة ADO إلى ملف المشروع .



العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

أولاً : عملية الاتصال بقاعدة البيانات :

يتم في البداية تكوين متغير يعبر عن قاعدة البيانات ، الهدف منه هو توصيف الاتصال بقاعدة البيانات ، بمعنى من هو مزود (محرك) قاعدة البيانات و ما هو مسار قاعدة البيانات ، كما يلي :

`Dim db As New ADODB.Connection` (متغير عام)

حيث db يعتبر متغير (كائن بالمعنى الاصح) ، و هذا الكائن يستخدم لتوصيف الاتصال بقاعدة البيانات كما يلي :

`db.Provider = "Microsoft.JET.OLEDB.4.0;"`

`db.Open "c:\db1.mdb"`

الخاصية Open
تستخدم لتحديد
مسار قاعدة
البيانات

الخاصية Provider
تعني مزود (محرك)
قاعدة البيانات
و التي هي قاعدة
بيانات Access

ملاحظات عند الاتصال بقاعدة البيانات

• الجملة :

```
db.Provider = "Microsoft.JET.OLEDB.4.0; "
```

تستخدم للاتصال بقاعدة بيانات من نوع Access ، و عند الاتصال بقاعدة بيانات SQL نستخدم :

```
db.Provider = " SQLOLEDB. 1; "
```

• في حالة ان كانت قاعدة البيانات محمية بكلمة مرور (مثلاً jam هو اسم المستخدم و كلمة المرور هي 123) نقوم بفتح الجداول كما يلي :

```
db.Open "c:\SALE.mdb", "userID='jam'", "password=123"
```

• من الممكن اختصار جملة تخطيط المزود و فتح جدول من قاعدة البيانات باستخدام الخاصية **ConnectionString** و التي ستكون :

```
db. ConnectionString = "provider=microsoft.jet. OLEDB. 4. 0;data source= c:\SALE.mdb"
```

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

ثانياً : عملية فتح جدول من قاعدة البيانات :

يتم تكوين متغير يعبر عن الجدول (مجموعة السجلات) التي سيتم التعامل معه في قاعدة البيانات كما يلي :

`Dim rs As New ADODB.Recordset` (متغير عام)

حيث rs يعتبر متغير (كائن بالمعنى الاصح) ، و هذا الكائن يستخدم لتحديد مجموعة السجلات (الجدول) التي يتم الارتباط بها من اجل التعامل معها :

متغير قاعدة البيانات , "اسم الجدول" rs.Open
نوع التزامن المطلوب , نوع السجلات

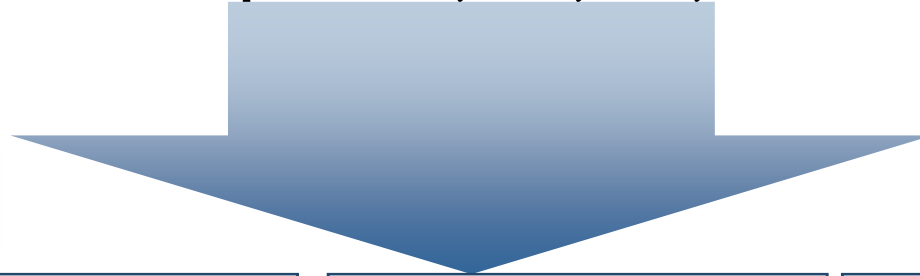
مثال :

`rs.Open "[Table1]", db, adOpenStatic, adLockReadOnly`

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

ثانياً : تابع عملية فتح جدول من قاعدة البيانات :

rs.Open 1, 2 , 3 , 4



اسم الجدول المراد
الاتصال به

اسم المتغير الذي تم
تعريفه من نوع
ADODB.Connection

نوع السجلات المرتبطة :
adOpenDynamic
تعامل مباشرة من السجلات
الحقيقية

adOpenForwardOnly
بالتحرك داخل السجلات في
الاتجاه للأمام فقط

adOpenKeyset
للقراءة فقط مع التعديل

adOpenStatic
للقراءة فقط

نوع التزامن :

adLockBatchOptimistic
غلق السجل من الآخرين اثناء
التعامل معه ، ثم اطلاق السيرفر
على التحديث

adLockOptimistic
غلق السجل من الآخرين اثناء
التعامل معه

adLockPessimistic
غلق السجل اثناء التعديل فقط

adLockReadOnly
السجل للقراءة فقط

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

عملية فتح جدول (جداول) من قاعدة البيانات باستخدام جمل SQL:

• استخدام جمل SQL يتيح مرونة كبيرة من اجل استرجاع السجلات المراد التعامل معها ، حيث اننا نستطيع هنا ان نحدد الاعمدة المراد استرجاعها ، او استخدام الشرط Where لتحديد السجلات المراد استرجعها و كذلك استرجاع السجلات من اكثر من جدول (بشرط وجود علاقة بين الجدول) و ذلك باستخدام JOIN .

مثال (1) : استرجاع اعمدة محددة (الأمثلة على قاعدة البيانات SALE) :

```
rs.Open "select CustomeName,Country from Customer",  
db,adOpenKeyset, adLockReadOnly
```

مثال (2) : استرجاع سجلات من جداول بينها علاقة :

```
rs.Open "select Customer.CustomeName,Orders.OrderDate  
FROM Customer INNER JOIN Orders ON  
Customer.CustomerID = Orders.CustomerID",  
db,adOpenKeyset, adLockReadOnly
```

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

استعراض بيانات حقول الجداول :

- امر الوصول إلى حقل في السجل المشار اليه حالياً هو :

rs![اسم العمود]

حيث **rs** هو اسم المتغير (الكائن) الذي تم تعريفه من نوع :

ADODB.Recordset

- لاحظ انه عند التعامل مع المكتبة ADO سوف تختصر اوامر الاداة ADODC التي تحوي ADODC1.Recordset إلى اسم المتغير (الكائن) rs الذي تم تعريفه من نوع ADODB.Recordset ثم بقية الاوامر المعتادة .

- مثال لعرض بيانات الحقل ID :

If Not IsNull(rs![ID]) Then Text1 = rs![ID]

حيث ان الغرض من **Not IsNull** هو التأكد من الحقل الذي تم ارجاعه ليس فارغاً

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

• إنشاء سجل جديد في قاعدة البيانات :

rs.Addnew

حيث ان :

rs : هو الكائن المعروف من نوع ADODB.Recordset و الذي سيجرى من خلاله التعامل مع بيانات قاعدة البيانات التي تم الاتصال بها .

• تعديل بيانات السجل الحالي :

rs.Update

حيث يقوم هذا الامر بحفظ التعديلات التي تجرى على السجل الحالي .

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

• حذف سجل من قاعدة البيانات :

rs.Delete

rs.MoveNext

حيث ان هذا الامر يقوم بحذف السجل الحالي ، و الخاصية MoveNext تقوم تقوم بالانتقال الى السجل التالي بعد الحذف.

- الدالة EOF (rs. EOF) تعيد القيمة True اذا كان المؤشر عند اخر سجل في الجدول .
- الدالة BOF (rs. BOF) تعيد القيمة True اذا كان المؤشر عند اول سجل في الجدول .

تطبيقات عملية

- التطبيق (1) : عملية الاضافة ، على اعتبار ان البيانات المراد اضافتها موجودة في مربعات النصوص Text Boxes :

rs. AddNew

rs![name] = Txt1

rs![address] = Txt2

rs![phone] = Txt3

rs. Update

- التطبيق (2) : عملية التعديل (حفظ التعديلات)، على اعتبار ان البيانات المراد تعديلها و حفظ التعديل موجودة في مربعات النصوص :

rs![name] = Txt1

rs![address] = Txt2

rs![phone] = Txt3

rs. Update

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

- ذكرنا سابقاً ان الامر [rs!] هو للوصول إلى حقل المشار اليه من العمود ، و كذلك يمكن الوصول الى الحقل المشار اليه حالياً من عمود في الجدول بالطريقة التالية :

rs.Fields(#no)

حيث ان #no تمثل رقم العمود بحسب ترتيب استرجاعه من قاعدة البيانات ، مثال :

rs.Fields(0) ←

للدلالة على اننا نريد الوصول إلى حقل اول عمود تم استرجاعه

- و كذلك يمكن استخدام الطريقة التالية :

rs.Fields![اسم العمود]

و للاسترجاع القيمة الحقل الذي يشار اليه :

Text1.text= rs.Fields![اسم العمود]. Value

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

• اوامر التنقل بين السجلات :

rs.MoveFirst ← الانتقال الى السجل الاول

rs.MoveLast ← الانتقال الى السجل الأخير

rs.MoveNext ← الانتقال الى السجل التالي

rs.MovePrevious ← الانتقال الى السجل السابق

rs.Move (#no) ← المرور على السجلات
بحسب قيمة #no

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

- امر البحث بين السجلات هو الامر Find و الذي يحتوي على اربع وسائط ، يعتبر اولها الزامي و البقية اختياري :

rs.Find 1 , 2 , 3 , 4



1/ اسم العمود و
الحقل المراد البحث
عنه :
"Name= 'Ali' "

2/ نقطة بداية البحث
و توضع 0 افتراضياً
او يتم البدء في البحث
من النقطة المتوقف
عندها المؤشر حالياً

3/ اتجاه البحث و هو
احد خيارين :
adSearchBackward
أي وصولاً إلى اول سجل
adSearchForward
أي وصولاً إلى اخر سجل

4/ نقطة بداية البحث
و لكننا نحدد هنا
من أي صف سوف
يبدأ البحث

مثال :

rs.Find " Name = 'Ali' ", 0, adSearchForward, 1

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

- الامر Find يسترجع سجل وحيد من السجلات التي تم الارتباط بها باستخدام الكائن rs ، و لكن ماذا اذا اردنا ان نسترجع اكثر من سجل اثناء عملية البحث ؟ الامر الذي يستخدم لذلك هو استخدام الخاصية Filter ، كما يلي :

```
rs.Filter = "Country = 'اليمن'"
```

حيث ان السجلات التي تم استرجعها من قاعدة البيانات باستخدام الامر rs.Open سوف يتم تصفيتها (فلتراتها) إلى السجلات التي قيمة الحقل فيها 'اليمن' من العمود Country .

لاحظ ان عملية التصفية باستخدام الامر Filter تقوم بالتأثير على بيانات التي تم استرجعها باستخدام rs.Open ، و من اجل الغاء عملية التصفية هذه ، و العودة إلى السجلات الاصلية التي تم استرجاعها باستخدام الامر rs.Open نقوم بالتالي :

```
rs.Filter = ""
```

العمليات الرئيسية التي تجرى باستخدام المكتبة ADO

- يستخدم الامر `rs.RecordCount` لمعرفة عدد السجلات التي تم استرجاعها باستخدام الامر `rs.Open` او الامر `rs.Filter`.
- الامر `db.Close` يستخدم لغلق قاعدة البيانات ، حيث ان `db` هو المتغير (الكائن) الذي تم تعريفه `ADODB.Connection` من اجل توصيف الاتصال بقاعدة البيانات .

عرض البيانات باستخدام المكتبة ADO

- عندما تحدثنا عن الاداة ADODC عرفنا انه من الممكن عرض البيانات داخل اداة خاصة تعرف باسم DataGrid وهي تتميز بعرض البيانات على شكل جدول ، و من الممكن ايضاً استخدامها مع المكتبة ADODB ، و يتم اختيارها من Component بإضافة الاداة Microsoft DataGrid :

- و من اجل عرض البيانات في الـ DataGrid باستخدام المكتبة ADO فإن الكود البرمجي الذي يلزمنا هو :

```
db.CursorLocation = adUseClient ←  
Set DataGrid1.DataSource = rs
```

للإشارة إلى ان جهة التعامل مع قاعدة البيانات ليست Server بل Client

حيث ان rs يعتبر كائن و لذلك نستخدم الامر Set عندما نريد ان نسند كائن إلى اخر . (لاحظ انه اولاً يتم اضافة الـ DataGrid إلى الفورم) .

عرض البيانات في DataList و ComboList

- من الممكن كذلك عرض الحقول داخل قوائم منسدلة ، و يستفاد من ذلك في تقييد عملية ادخال البيانات من المستخدم و ذلك بالاعتماد على جداول الترميز (وضح اكثر ؟) .
- تستخدم الاداتان DataList و ComboList من اجل عرض البيانات داخل قوائم منسدلة ، و يتم اضافة الاداتان من Microsoft DataList Control 6.0 Component بإضافة  و سوف يظهران بالشكل التالي
- من اجل عرض بيانات عمود ما فيهما (حقل) نقوم بالبحث في خصائصهما عن الخاصية RowSource و اسناد اداة الربط بقاعدة البيانات ADODC1 اليها و الخاصية ListField و اسناد اسم العمود المراد عرضه اليها .

عرض البيانات في DataList و ComboBox

- اما عرض البيانات داخل DataList و ComboBox باستخدام المكتبة ADODB فيلزم منا ذلك اسناد القيم إلى الخصائص RowSource و ListField برمجياً كما يلي (المثال التالي للأداة ComboBox و هو لا يختلف ابداً عن عرض البيانات برمجياً في الاداة DataList):

```
Set DataCombo1.RowSource = rs
```

```
DataCombo1.ListField = "CustomeName"
```

اسم العمود

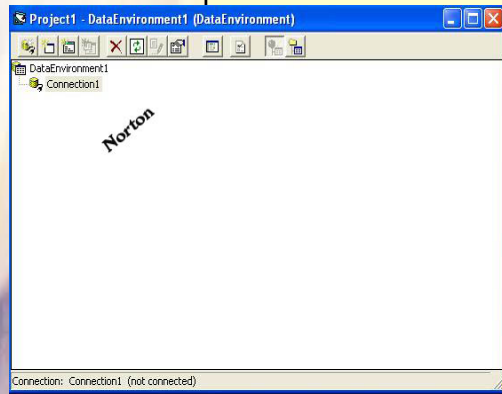


المرحلة الاخيرة ،، انشاء التقارير

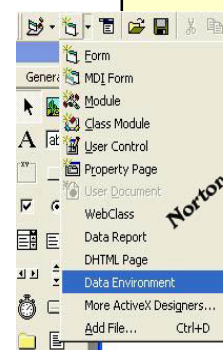
- تعتبر التقارير هي الخلاصة الاخيرة للبيانات ، و التي يستفيد منها المستخدم (و التي يقاس من خلالها جودة اداء النظام) ، و يتم ربط التقارير بقاعدة البيانات ، و عرض سجلات محددة منها اما بشكل مباشر او عن طريق جمل SQL ، و من الممكن جعل المستخدم تحديد التقرير الذي يريده و من ثم ترجمة ذلك إلى جملة SQL نعرض من خلالها التقرير المطلوب ، مع امكانية اخراج ذلك التقرير بالطابعة للمستخدم .
- يوفر Visual Basic بيئة جاهزة للاتصال بقاعدة البيانات تسمى هذه البيئة بـ DataEnvironment و بيئة اخرى جاهزة لإنشاء التقارير تسمى DataReport .

خطوات انشاء التقارير

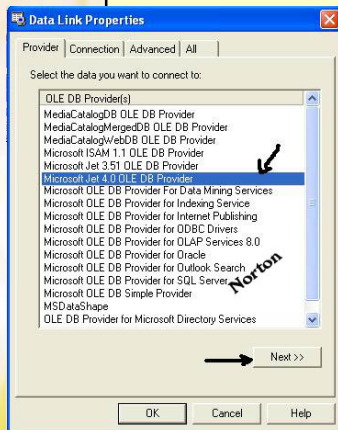
يتم انشاء التقرير على مرحلتين ، الاولى هي الاتصال بقاعدة البيانات عن طريق البيئة
DataEnvironment ثم انشاء التقرير عن طريق البيئة DataReport



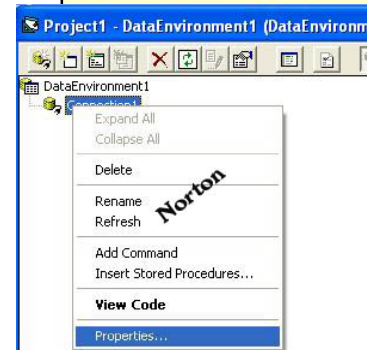
2/ يتم انشاء البيئة
DataEnvironment1
و التي سوف تحتوي على
عنصر Connection1
افتراضي نتصل من خلاله
بقاعدة البيانات



1/ يتم الاتصال بقاعدة
البيانات عن طريق البيئة
DataEnvironment ،
و التي يتم انشاءها من القائمة
Project ثم
add DataEnvironment



4/ في اول خطوة من خطوات
توصيف الاتصال بقاعدة البيانات
نحدد كما هو معتاد المزود
(المحرك) الخاص بقاعدة
البيانات المراد الاتصال بها

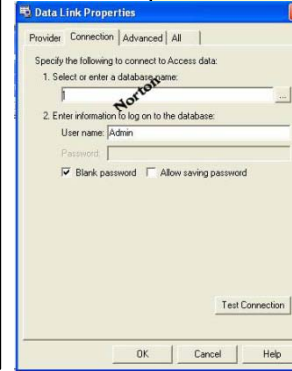


3/ من القائمة المنسدلة
للعنصر Connection1
نختار الامر خصائص ، من
اجل توصيف الاتصال بقاعدة
البيانات (مزود قاعدة البيانات +
مسار قاعدة البيانات)

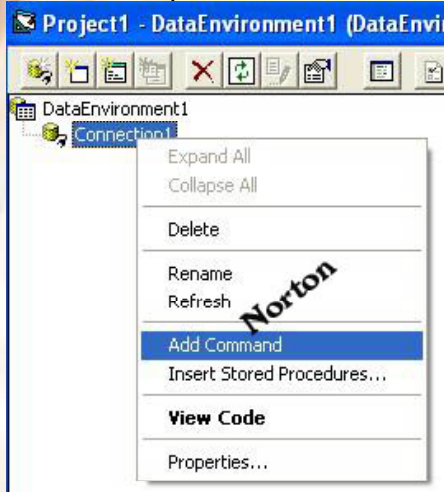
تابع خطوات انشاء التقارير



16/ أخيراً يتم التأكد من ان الاتصال تم بنجاح عن طريق تجربة الاتصال بالزر **Test Connection**

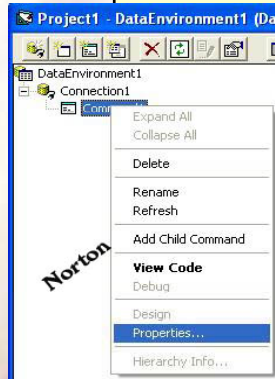


15/ ثانية خطوة من خطوات توصيف الاتصال بقاعدة البيانات هو ان نحدد مسار قاعدة البيانات من خلال الزر الذي يحتوي ثلاث نقاط

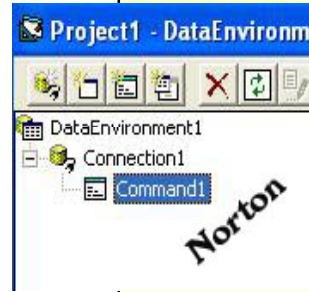


17/ بعد ان تم الاتصال بقاعدة البيانات ، نحدد من هي الحقول المراد عرضها في التقرير ، و قبل ذلك لابد من انشاء امر **Command** نجري من خلاله الاتصال بالحقول المراد عرضها و ذلك من القائمة **Connection1** ثم نختار الامر **Add Command** ، فيتم انشاء **Command** جديد ، و من الممكن انشاء اكثر من **Command** حيث انكل **Command** ممكن ان يشكل تقريراً ما ..

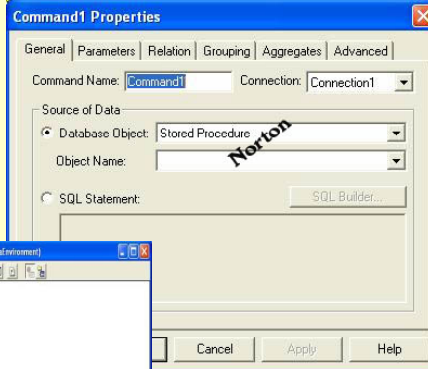
تابع خطوات انشاء التقارير



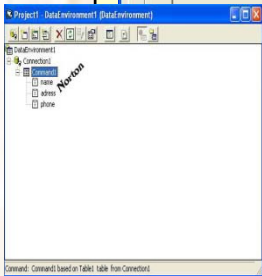
9/ الآن يتم الدخول إلى خصائص من القائمة المنسدلة الخاصة بالأمر Command من اجل تحديد الحقول المراد عرضها في التقرير .



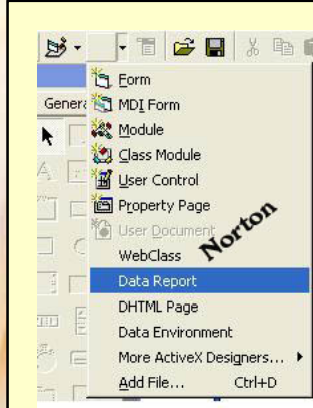
8/ نلاحظ الآن انه تم اضافة امر Command1 في Connection1 (التي توصلنا بقاعدة البيانات) ، و من الممكن ان يكون لدينا اكثر من Command على قاعدة البيانات الواحدة .



10/ من نافذة الخصائص ، العنصر Command Name يتيح لنا تغيير اسم ال- Command إلى أي اسم مناسب لنا ، و في الامر DataBase Object نحدد من القائمة المنسدلة اننا نريد الاتصال بجدول ، و في القائمة المنسدلة للأمر Object Name نحدد اسم الجدول ، و الخانة SQL Statement من الممكن نكتب جملة SQL نحدد من خلالها الحقول المطلوبة ، و بعد الموافقة نلاحظ ان الامر Command كون شجرة بالحقول المطلوبة



تابع خطوات انشاء التقارير



11/ بعد ان تمت عملية الارتباط بقاعدة البيانات ، و استخلاص الحقول المطلوبة ، نقوم بإنشاء التقرير ، و ذلك من القائمة **Project ثم Add Data Report** ، فنلاحظ انه تم انشاء التقرير.

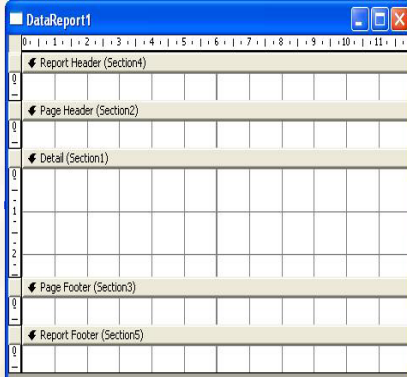


13/ بعد ذلك نقوم بتحديد ال **Command** الذي يحتوي على الحقول المراد عرضها في التقرير عن طريق الخاصية **Data Member** من خصائص التقرير .



12/ الآن نقوم بربط التقرير بال **DataEnvironment1** التي تحتوي توصيف قاعدة البيانات ، و ذلك من خلال الخاصية **Data Source** في خصائص التقرير **DataReport1**

تابع خطوات انشاء التقارير



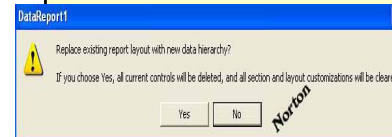
14/ لاحظ شاشة التقرير ، سوف تجد انها مقسمة إلى خمسة اجزاء كما يلي :

- الجزء الاول و الاخير Report Header – Report Footer يقصد به الرأس و التذييل الذي سوف يظهر في الصفحة الاولى من التقرير .
- الجزء الثاني و الرابع Page Header – Page Footer يقصد به الرأس و التذييل الذي سوف يظهر في كل صفحات التقرير .
- الجزء الاخير (الثالث) جز التفاصيل و ه مخصص لعرض تفاصيل التقرير فيه ، أي عرض حقول قاعدة البيانات

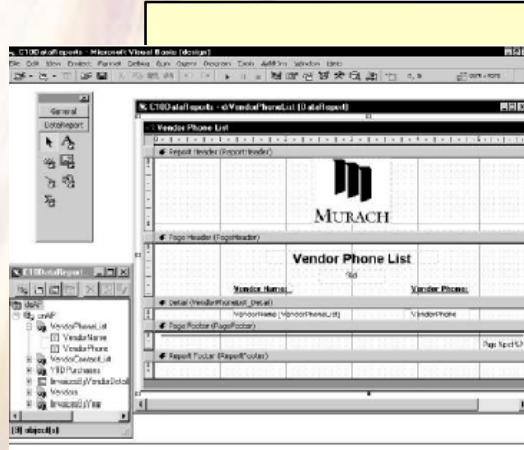


16/ الآن تبدأ مرحلة تصميم و تنسيق، بالنسبة لتنسيق التقرير من خلال صندوق الادوات نستطيع اضافة العديد من الادوات (Label, Image ...) و بالضغط على الزر الايمن و اختيار الامر Insert Control نستطيع اضافة ترقيم للصفحات او التاريخ ... الخ

15/ في أي مكان من شاشة التقرير اضغط بالزر الايمن و من القائمة المنسدلة اختر الامر Retrieve Structure و اضغط موافق على شاشة رسالة التنبيه التي تظهر .



تابع خطوات انشاء التقارير



17/ الآن تأتي المرحلة المهمة و هي عرض الحقول التي تم ادراجها ضمن الامر **Command1** داخل التقرير ، و يتم ذلك بفتح شاشة التقرير و شاشة الـ **DataEnvironment** في آن معاً ، ثم نقوم بسحب اسم الحقل من الشجرة الخاصة بالـ **Command** إلى جزء التفاصيل في التقرير ... فنلاحظ انه تم انشاء خانة خاصة بالحقل داخل التقرير و خانة اخرى تمثل **Label** يوضح اسم الحقل و هكذا حتى ننقل كل الحقول المطلوبة إلى التقرير ... نستطيع تنسيق الحقول و

البيانات التي يظهرها التقرير من خلال خصائص كل خانة و من المستحب نقل **Labels** الخاص بكل جزء إلى قسم **Page Header** من التقرير حتى يكون ظاهر في كل صفحة



18/ تأتي الآن اخر مرحلة .. و هي عرض التقرير داخل المشروع من خلال احدى النوافذ **Forms** و ذلك باستخدام الامر **DataRepot1.Show**

خطوات انشاء التقارير الحي ☺

• إلى الآن تم تصميم تقارير ناجحة باستخدام الـ Data Report ، غير ان هذه التقارير تقوم بعرض كل السجلات التي تحتويها قاعدة البيانات ... و حتى في حالة استخدام جملة SQL فإننا نلزم المستخدم بعرض محدد .. و من المرونة ان نجعل المستخدم ان يحدد نوع التقرير الذي يريده ، و ذلك (على سبيل المثال) بأن نجعل المستخدم يحدد ان التقرير الذي يريده هو ناتج عن البحث بحسب حقل معين (و ليكن تاريخ الطلبية ...) ، و يكون عرض التقرير ناتج ذلك البحث ، و هكذا فإننا سوف نكون قد كونا تقريراً يعرض الخانات المطلوبة حيث ان جملة الاستعلام التي يعتمد عليها التقرير سوف تعتمد على الشرط الذي يحدده المستخدم و هذا ما يسمى بالتقرير الحي ..

خطوات انشاء التقارير الحي ☺

- بعد ان ننتهي من تجهيز DataEnvironment تحتوي على Command وكذلك من ربط التقرير بـ DataEnvironment نقوم بما يلي :
 - 1/ من خلال تصميم التقرير نغير الخاصية Data Field لمربعات النصوص إلى اسماء الحقول التي سوف تعرضها ..
 - 2/ في داخل الفورم .. و عند نقطة البحث التي سيتم من خلالها استدعاء التقرير نضيف الشفرة التالية :

تابع خطوات انشاء التقارير الحي 😊

```
Private Sub Command1_Click()  
  
Dim name, SQLstr As String  
name = InputBox("ادخل اسم الدولة")  
  
SQLstr = "select * from Customer where Country = '" & name & "'"   
  
DataEnvironment1.Commands(1).CommandType = adCmdText  
DataEnvironment1.Commands(1).CommandText = SQLstr  
DataEnvironment1.Commands(1).Execute  
  
DataReport1.Show  
  
End Sub
```

الشفرة السابقة تقوم بالطلب من المستخدم ان يدخل اسم الدولة التي سيكون على اساسها التقرير ، و من ثم ينشئ استعلاماً مناسباً يسنده إلى الخاصية التالية من اجل جعل Command1 ينفذ الاستعلام :

DataEnvironment1.Commands(1).CommandText

و قبل ذلك يجعل Command1 مستعداً لاستقبال استعلام عن طريق الخاصية :

DataEnvironment1.Commands(1).CommandType = adCmdText

و من ثم ينفذ الاستعلام بالأمر :

DataEnvironment1.Commands(1).Execute

و اخيراً يعرض التقرير عن طريق الامر :

DataReport1.Show

تم بحمد الله تعالى