

تعريف ومعالجة البيانات في الأكسيس بواسطة DAO

Par Christophe WARIN
www.developpez.com
Traduit par Med ALAOUI SOSSE
www.arabmoheet.net

تأليف : كريستوف وارين
تعريب : محمد علوي صوصي
(أبو تماضر)

De: Christophe Warin [visual.christophe@laposte.net]

Date: mar. 21/02/2006 17:41

À: MED ALAOUI SOSSE

Cc:

Objet: RE: Privé à Mr Christophe WARIN

Pièces
jointes :

[Afficher sous forme de page Web](#)

Bonjour

Je ne vois aucune objection à ce que mon tutoriel soit traduit en arabe tant que mon nom (et idéalement mon site web warin.developpez.com) figure en tant que auteur à coté du votre (en tant que traducteur). Une fois votre travail terminé, et si vous le souhaitez, vous pouvez m'en faire parvenir une copie que je déposerai sur le site www.developpez.com afin d'élargir la portée de votre traduction à nos visiteurs plus à l'aise avec la langue arabe.

Mes félicitations pour votre initiative.

ترجمة نص رسالة الترخيص بالنشر :

بعد التحية

لا أرى مانعا أن يترجم النص إلى العربية ، ما دام يحفظ لاسمي (والأحرى لموقعي warin.developpez.com) حق التأليف ، إلى جانب اسمك كمترجم . وفي حال انتهاء اشتغالك على النص ، وإن تكرمت ، زدوني بنسخة منه ، لأنشرها على الموقع www.developpez.com بغاية توسيع أفق ترجمتك إلى زوار موقعي ممن تيسر لهم فهم اللغة العربية .
تهنئتي لك على المبادرة .

فهرسة

القسم الأول : مدخل

1 - مدخل

القسم الثاني : تراتبية النموذج DAO

2 - تراتبية النموذج DAO :

- 2.1 - النموذج الكائني
- 2.2 - تذكير حول مفهوم المجموعات collections
 - 2.2.1 - الوصول إلى عنصر ضمن مجموعة
 - 2.2.2 - تعداد عناصر مجموعة
 - 2.2.3 - تصفح عناصر مجموعة
 - 2.2.4 - إضافة عنصر
 - 2.2.5 - حذف عنصر

القسم الثالث : الوصول إلى قاعدة بيانات

3 - الوصول إلى قاعدة بيانات

3.1 - الكائن DBEngine

- 3.1.1 - عموميات
- 3.1.2 - قائمة الخصائص
 - 3.1.2.1 - DefaultPassword
 - 3.1.2.2 - DelfaultType
 - 3.1.2.3 - DefaultUser
 - 3.1.2.4 - Errors
 - 3.1.2.5 - Inipath
 - 3.1.2.7 - Properties
 - 3.1.2.8 - SystemDB
 - 3.1.2.9 - Version
 - 3.1.2.10 - Workspaces
- 3.1.3 - قائمة الطرائق
 - 3.1.3.1 - BeginTrans
 - 3.1.3.2 - CommitTrans
 - 3.1.3.3 - CompactDatabase
 - 3.1.3.4 - CreateDatabase
 - 3.1.3.5 - CreateWorkspace
 - 3.1.3.6 - Idle
 - 3.1.3.7 - OpenDatabase
 - 3.1.3.8 - RepairDatabase
 - 3.1.3.9 - Rollback

- 3.2 – الكائن Workspace
 - 3.2.1 – عموميات
 - 3.2.2 - قائمة الخصائص
 - 3.2.2.1 – Databases
 - 3.2.2.2 – IsolateODBCTrans
 - 3.2.2.3 – Name
 - 3.2.2.4 – Properties
 - 3.2.2.5 – Type
 - 3.2.3 - قائمة الطرائق
 - 3.2.3.1 – BeginTrans
 - 3.2.3.2 – Close
 - 3.2.3.3 – CommitTrans
 - 3.2.3.4 – CreateDatabase
 - 3.3.3.5 – OpenDatabase
 - 3.2.3.6 – Rollback
 - 3.2.4 – الولوج إلى مجموعة عمل مخصصة
- 3.3 – الكائن Database
 - 3.3.1 - عموميات
 - 3.3.1.1 – الطريقة CurrentDb
 - 3.3.1.2 – الطريقة CodeDb
 - 3.3.2 - قائمة الخصائص
 - 3.3.2.1 – CollatingOrder
 - 3.3.2.2 – Connect
 - 3.3.2.3 – Containers
 - 3.3.2.4 – Name
 - 3.3.2.5 – Properties
 - 3.3.2.6 – QueryDefs
 - 3.3.2.7 – RecordsAffected
 - 3.3.2.8 – Recordsets
 - 3.3.2.9 – Relations
 - 3.3.2.10 – TableDefs
 - 3.3.2.11 – Transactions
 - 3.3.2.12 – Updatable
 - 3.3.2.13 - Version
 - 3.3.3 – قائمة الطرائق
 - 3.3.3.1 – Close
 - 3.3.3.2 – CreateProperty
 - 3.3.3.3 – CreateQueryDef
 - 3.3.3.4 – CreateRelation
 - 3.3.3.5 – CreateTableDef
 - 3.3.3.6 – Execute
 - 3.3.3.7 – NewPassword
 - 3.3.3.8 - OpenRecordset

القسم الرابع : هيكل قاعدة البيانات في نموذج DAO

- 4 - DAO وهيكل قاعدة البيانات
 - 4.1 - عموميات
 - 4.2 - الخصائص النوعية
 - 4.2.1 - تعريفات
 - 4.2.2 - قائمة الخصائص للكائن Property
 - 4.2.2.1 - Inherited
 - 4.2.2.2 - Name
 - 4.2.2.3 - Type
 - 4.2.2.4 - Value
 - 4.2.3 - استدعاء الخصائص
 - 4.2.4 - تعديل قيم الخصائص
 - 4.2.5 - إنشاء خصائص جديدة
 - 4.3 - الكائن TableDef
 - 4.3.1 - تعريفات
 - 4.3.2 - قائمة الخصائص
 - 4.3.2.1 - Attributes
 - 4.3.2.2 - Connect
 - 4.3.2.3 - DataCreated
 - 4.3.2.4 - Fields
 - 4.3.2.5 - Indexes
 - 4.3.2.6 - LastUpdated
 - 4.3.2.7 - Name
 - 4.3.2.8 - Properties
 - 4.3.2.9 - RecordCount
 - 4.3.2.10 - SourceTableName
 - 4.3.2.11 - Updatable
 - 4.3.2.12 - ValidationRule
 - 4.3.2.13 - ValidationText
 - 4.3.3 - قائمة الطرائق
 - 4.3.3.1 - CreateField
 - 4.3.3.2 - CreateIndex
 - 4.3.3.3 - CreateProperty
 - 4.3.3.4 - OpenRecordset
 - 4.3.3.5 - RefreshLink
 - 4.3.4 - إجراءات على الجداول في قاعدة بيانات
 - 4.3.4.1 - عرض الجداول بقاعدة البيانات
 - 4.3.4.2 - إعادة تسمية جدول
 - 4.3.4.3 - حذف جدول
 - 4.3.4.4 - إنشاء جدول

4.4 – الكائن Field

4.4.1 - تعريف

4.4.2 – قائمة الخصائص

AllowZeroLength 4.4.2.1 –

Attributes 4.4.2.2 –

CollatingOrder 4.4.2.3 –

DataUpdatable 4.4.2.4 –

DefaultValue م 4.4.2.4 –

FieldSize 4.4.2.5 –

ForeignName 4.4.2.6 –

Name 4.4.2.7 –

OrdinalPosition 4.4.2.8 –

Properties 4.4.2.9 –

Required 4.4.2.10 –

Size 4.4.2.11 –

SourceField 4.4.2.12 –

SourceTable 4.4.2.13 –

Type 4.4.2.14 –

ValidationRule 4.4.2.15 –

ValidationText 4.4.2.16 –

Value 4.4.2.17 –

VisibleValue 4.4.2.18 –

4.4.3 – قائمة الطرائق

AppendChunk 4.4.3.1 –

CreateProperty 4.4.3.2 –

GetChunk 4.4.3.3 –

4.4.4 - إجراءات على حقل بالكائن TableDef

إعادة تسمية حقل 4.4.4.1 -

حذف حقل 4.4.4.2 -

إنشاء حقل 4.4.4.3 -

نسخ حقل 4.4.4.4 -

4.5 - الكائن Index

4.5.1 - تعريفات

4.5.2 - قائمة الخصائص

DistinctCount 4.5.2.1 –

Fields 4.5.2.2 –

Foreign 4.5.2.3 –

IgnoreNulls 4.5.2.4 –

Name 4.5.2.5 –

Properties 4.5.2.6 –

Required 4.5.2.7 –

Unique 4.5.2.8 –

- 4.5.3 – قائمة الطرائق
 - 4.5.3.1 – CreateField
 - 4.5.3.2 – CreateProperty
 - 4.5.4 – إجراءات على الفهارس
 - 4.5.4.1 – إنشاء فهرس
 - 4.5.4.2 – حذف مفتاح أساس
- 4.6 – الكائن Relation
 - 4.6.1 – تعريفات
 - 4.6.2 – قائمة الخصائص
 - 4.6.2.1 – Attributes
 - 4.6.2.2 – Fields
 - 4.6.2.3 – ForeignTable
 - 4.6.2.4 – Name
 - 4.6.2.6 – Properties
 - 4.6.2.7 – Table
 - 4.6.3 – قائمة الطرائق
 - 4.6.4 – مثال عن تعريف الخصائص
 - 4.6.5 – إجراءات على العلاقات
 - 4.6.5.1 – إنشاء علاقة
 - 4.6.5.2 – تعداد العلاقات باستعمال حقل معطى
 - 4.6.5.3 – حذف علاقات لجدول
- 4.7 – الكائن QueryDef
 - 4.7.1 – تعريفات
 - 4.7.2 – قائمة الخصائص
 - 4.7.2.1 – Connect
 - 4.7.2.2 – DateCreated
 - 4.7.2.3 – Fields
 - 4.7.2.4 – LastUpdated
 - 4.7.2.5 – MaxRecords
 - 4.7.2.6 – Name
 - 4.7.2.7 – Parameters
 - 4.7.2.8 – Properties
 - 4.7.2.9 – RecordsAffected
 - 4.7.2.10 – ReturnsRecords
 - 4.7.2.11 – SQL
 - 4.7.2.12 – Type
 - 4.7.2.13 – Updatable
 - 4.7.3 – قائمة الطرائق
 - 4.7.3.1 – CreateProperty
 - 4.7.3.2 – Execute
 - 4.7.3.3 – OpenRecordset
 - 4.7.4 – إجراءات على الاستعلامات
 - 4.7.4.1 – إنشاء استعلام
 - 4.7.4.2 – تعديل الكود لاستعلام
 - 4.7.4.3 – عرض الاستعلامات

القسم الخامس : الوصول إلى البيانات

- 5 - الوصول إلى البيانات
 - 5.1 - تعريف
 - 5.2 - أصناف السجلات
 - 5.2.1 - انطلاقا من كائن Database
 - 5.2.2 - انطلاقا من كائن QueryDef
 - 5.2.3 - انطلاقا من نموذج
 - 5.3 - الوضع والتنقلات
 - 5.3.1 - التنقل بين سجلات Recordset
 - 5.3.2 - البحث عن سجل
 - 5.3.3 - حفظ الوضع
 - 5.4 - معالجة البيانات
 - 5.4.1 - قراءة سجل
 - 5.4.2 - تعديل سجل
 - 5.4.3 - إضافة سجل
 - 5.4.4 - حذف سجل
 - 5.4.5 - تعداد السجلات
 - 5.4.6 - تصفية البيانات
 - 5.5 - القراءة بالكتلة
 - 5.5.1 - عموميات
 - 5.5.2 - أمثلة
 - 5.6 - قائمة الخصائص
 - 5.6.1 - AbsolutePosition
 - 5.6.2 - BOF
 - 5.6.3 - Bookmark
 - 5.6.4 - Bookmarkable
 - 5.6.5 - CacheSize
 - 5.6.6 - CacheStart
 - 5.6.7 - Connection
 - 5.6.8 - DataCreated
 - 5.6.9 - EditMode
 - 5.6.10 - EOF
 - 5.6.11 - Fields
 - 5.6.12 - Filter
 - 5.6.13 - Index
 - 5.6.14 - LastModified
 - 5.6.15 - LastUpdated
 - 5.6.16 - LockEdits
 - 5.6.17 - Name
 - 5.6.18 - NoMatch

PercentPosition	5.6.19	–
Properties	5.6.20	–
RecordCount	5.6.21	–
Restartable	5.6.22	–
Sort	5.6.23	–
StillExecuting	5.6.24	–
Transactions	5.6.25	–
Type	5.6.26	–
Updatable	5.6.27	–
UpdateOptions	5.6.28	–
ValidationRule	5.6.29	–
ValidationText	5.6.30	–
5.7 - قائمة الطرائق		
AddNew	5.7.1	–
CancelUpdate	5.7.2	–
Clone	5.7.3	–
Close	5.7.4	–
CreateQueryDef	5.7.5	–
Delete	5.7.6	–
Edit	5.7.7	–
FindFirst, FindLast, FindNext, FindPrevious	5.7.8	–
GetRows	5.7.9	–
Move, MoveFirst, MoveLast, MoveNext, MovePrevious	5.7.10	–
OpenRecordset	5.7.11	–
Requery	5.7.12	–
Update	5.7.13	–

القسم السادس : 6 - خلاصة

6 - خلاصة

القسم الأول :


مدخل

كانت الوصول للبيانات DAO أو (Data Access Objects) هي من صنف المكتبات الحيوية DLL . وتتضمن مجموعة من الكائنات البرمجية التي تمكن من الوصول إلى البيانات . واستعمال هذه الأدوات يمكن بالمرّة من تشكيل هيكل قاعدة البيانات وكذا معالجة بياناتها .

من أهداف هذا الدرس ؛ أن يضع بين يديك صورة عن الإمكانيات اللامحدودة التي توفرها هذه المكتبة على مستوى هيكل قواعد البيانات ، وكذا معالجة بياناتها .


و من خلال ما يقدمه الدرس من أمثلة متعددة سيتمكن القارئ من هيكلية البيانات ، وكذا اكتشاف إمكانية إنشاء الجداول وتعديل الحقول ، أو إنشاء العلاقات ، وبناء استعلامات .. معالجة بيانات بنموذج (إضافة / حذف سجلات ، ...)

ستجد أيضا كل الخصائص والطرائق التي تتعلق بكل كائن ضمن قوائم ، غير أنني أدوه إلى أن هذه القوائم والشروح لا تغني عن تعليمات الأكسيس التي تظل معدنا لا ينضب للمعرفة بها .

<p>بعض الأجزاء من هذا الدرس ليست مفصلة بالشكل الكافي ، لأنني وجدتھا لا تستجيب لاستعمالات تقليدية لـ DAO مع المحرك Jet . حتى لا يتقل هذا الدرس على قارئه سوف لن تجد معلومات كافية حول ما يلي :</p> <ul style="list-style-type: none"> • الوصول إلى البيانات في مجال ODBC Direct • النسخ المماثلة • كائنات DAO التي لا تهم الوصول إلى البيانات مثل : (الأمان ، المستخدم ، ...) 	
--	---

و قبل البدء :

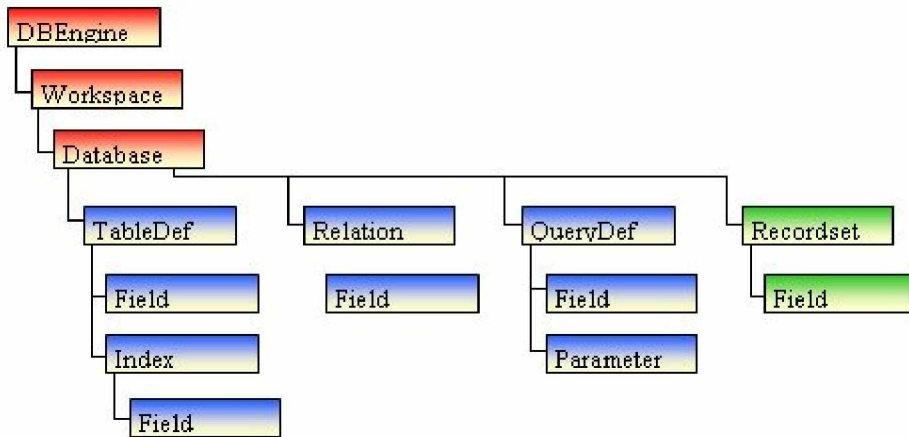
<p>مجموع الطرائق والكائنات التي تمكن من الوصول إلى البيانات بواسطة DAO مجمعة في ملف مكتبة حيوية ، لذا قبل الاستعمال قم بإضافة مرجع هذه المكتبة إلى مشروعك .</p> <p>Microsoft DAO 3.5 Object Library بالنسبة لـ Access 97 .</p> <p>Microsoft DAO 3.6 Object Library بالنسبة للإصدارات التالية ..</p>	
---	---

<p>كيف تضيف مرجعا إلى مشروعك بالأكسيس ؟</p> <p>من داخل محرر VBA بالأكسيس ، اختر قائمة الأوامر أدوات (Tools /Outils) ثم الخيار Références</p> <p>إذا لم تصادف هذه المكتبة ضمن قائمة المراجع اضغط على زر Browse ثم اتبع المسار التالي لإضافتها إلى القائمة :</p> <p>C:\Program Files\Common\Microsoft Shared\DAO\dao3xc.dll</p>	
---	---

القسم الثاني :

- تراتبية النموذج DAO

هذا تمثيل لهيكل الكائن : DAO



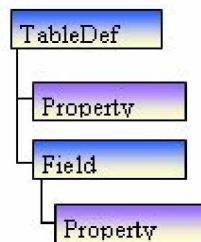
كما يمكن أن تلاحظ ؛ النموذج DAO يمتلك بنية شجرية (متفرعة) ترتكز إلى الكائن DBEngine وعلاقات التفرع هي من فئة واحد لمتعدد ، وهذا يعني أن كل كائن أب يتضمن مجموعة نوعية من الكائنات الأبناء ، حيث أن الاسم يصبح نمطيا :

اسم المجموعة = نوع الكائن الإبن + (علامة الجمع بالإنجليزي : "s")

فم ثلا الكائن DBEngine يتضمن مجموعة نوعية من الكائنات Workspace تسمى

. Workspaces

وتحديدا كذلك ؛ فهذا التصميم اختزل التفرعات إلى المجموعات الكائنية لمعالجة البيانات (باللون الأخضر) والمجموعات الكائنية للتعريف (اللون الأزرق) بغاية تحسين القراءة ، وألمح إلى أن الكائن DAO يتضمن مجموعة من الخصائص (property) جمعت في مجموعة باسم : properties تخص كل كائن في نموذج DAO وإليك نموذج عن الخطأة المختزلة في TableDef مثلا :



استعمال DAO يتطلب التصريح بالعديد من المتغيرات مما يجعل الكود مربكا في القراءة

- لذلك أنصح بتوحيد أسماء المتغيرات والكائنات البرمجية وقد وجدت جدولا طريفا يقترح توحيدا للإسم أضمنه هذا الكتاب فيما يلي:



- هذه المشاركة منقولة من تلخيص لأبي هاجر نشرت بإحدى المواقع العربية :

أولاً : كائنات قاعدة البيانات

الكائن	بداية الاسم	مثال
Table	tbl	tblCustomer
Query	qry	qryOverAchiever
Form	frm	frmCustomer
Report	rpt	rptInsuranceValue
Macro	mcr	mcrUpdateInventory
Module	bas	basBilling

ثانياً : حقول الجدول

نوع الحقل	بداية الاسم	مثال
Binary	bin	binInternal
Byte	byt	bytFloorNum
Counter	lng	lngPKCnt
Currency	cur	curSalary
Date/Time	dtm	dtmHireDate
Double	dbl	dblMass
Integer	int	intUnit
Long	lng	lngPopulation
Memo	mem	memComments
Ole	ole	oleEmpPhoto
Single	sng	sngScore
Text	str	strFirstName
Yes/No	ysn	ysnDiscounted

ثالثاً : عناصر التحكم في النماذج والتقارير

نوع العنصر	بداية الاسم	المثال
Chart (graph)	cht	chtSales
Check box	chk	chkReadOnly
Combo box	cbo	cboIndustry
Command button	cmd	cmdCancel
Frame (object)	fra	fraPhoto
Label	lbl	lblHelpMessage
Line	lin	linVertical
List box	lst	lstPolicyCode

المثال	بداية الاسم	نوع العنصر
optFrench	opt	Option button
grpLanguage	grp	Option group
brkPage1	brk	Page break
subContact	sub	Subform/report
txtoginName	txt	Text box
tglForm	tgl	Toggle button

رابعاً : المتغيرات في الوحدة النمطية

مثال	بداية الاسم	نوع المتغير
Dim ctlVapor As Control	ctl	Control
Dim curSalary As Currency	cur	Currency
Dim dbCurrent As Database	db	Database
Dim docRelationships as Document	doc	Document
Dim dblPi As Double	dbl	Double
Dim fldLastName as Field	fld	Field
Dim frmGetUser As Form	frm	Form
Dim gruManagers as Group	gru	Group
Dim idxOrderId as Index	idx	Index
Dim intRetVal As Integer	int	Integer
Dim lngParam As Long	lng	Long
Dim objGraph As Object	obj	Object
Dim prmBeginDate as Parameter	prm	Parameter
Dim prpUserDefined as Property	prp	Property
Dim qdfPrice As QueryDef	qdf	QueryDef
Dim recPeople as Recordset	rec (or rst)	Recordset
Dim relOrderItems as Relation	rel	Relation
Dim sngLoadFactor As Single	sng	Single
Dim strUserName As String	str	String
Dim tblVendor As Table	tbl	Table
Dim tdfBooking as TableDef	tdf (or tbd)	TableDef
Dim varInput As Variant	var	Variant
Dim wrkPrimary as Workspace	wrk (or wsp)	Workspace
Dim ysnPaid As Boolean	ysn	Yes/No

- 2.2 تذكير حول مفهوم المجموعات collections

بما أن DAO نموذج كائني يتضمن كثيرا من المجموعات ، أقترح عليك عزيزي القارئ فيما يلي تذكيرا بمعالجة المجموعات الكائنية في Visual Basic :

- 2.2.1 الوصول إلى عنصر ضمن مجموعة

هناك تقنيتان للوصول إلى كائن ضمن مجموعة :
وصول عبر فهرس المجموعة : وهذا يعني أن تحدد مكان الكائن ضمن تسلسل المجموعة مع العلم أن أول عنصر في المجموعة فهرسه = صفر

مثال للوصول إلى العنصر الثالث في المجموعة :

`MaCollection.Item(2)`

وصول عبر المفتاح : وهذا يعني استعمال اسم الكائن عندما يكون هذا معروفا .
مثال :

`MaCollection.Item ("MonItem")`

<p>الخاصية <code>Item</code> هي خاصية افتراضية للكائن من نوع مجموعة Collection ، لذلك يمكن ذكرها أو أهملها في التركيب فتقول :</p> <p><code>MaCollection.Item ("MonItem")</code></p> <p>أو</p> <p><code>MaCollection.("MonItem")</code></p>	
---	---

- 2.2.2 تعداد العناصر ضمن مجموعة

في لغة VBA المجموعات تتضمن خاصية `Count` والتي تحدد عدد العناصر ضمن مجموعة التركيب :

`MaCollection.Count`

و حيث إن أول عنصر في المجموعة يأخذ الفهرس صفر فإن آخر عنصر سيأخذ الفهرس `Count - 1`



- 2.2.3 تصفح عناصر مجموعة

هناك طريقتان لتصفح عناصر مجموعة :

- إذ يمكن تمرير عناصر مجموعة من خلال الحلقة التكرارية **For.... Next** التي ترتكز إلى فهرس المجموعة حتى آخر عنصر بها وهذا مثال :

```
Dim i As Integer, nbMax As Integer
' Workspaces : تعداد عناصر المجموعة
nbMax = DBEngine.Workspaces.Count - 1
' حلقة تكرارية ترتكز إلى الفهرس '
For i = 0 To nbMax
' إظهار أسماء عناصر المجموعة '
MsgBox DBEngine.Workspaces(i).Name
Next i
```

والطريقة الثانية : تعتمد على الحلقة التكرارية :
For Each # In # Next •
 لتصفح عناصر المجموعة واحدا واحدا
 ومثال ذلك :
 (لكل عنصر Workspace من المجموعة Workspaces ضمن الكائن
 DBEngine أظهر اسم العنصر)

```
Dim oWks As DAO.Workspace
For Each oWks In DBEngine.Workspaces
    إظهار أسماء عناصر المجموعة '
MsgBox oWks.Name
Next oWks
```

وهذه الطريقة الأخيرة لها ميزة فريدة لتصفح عناصر مجموعة ، خاصة في المجموعات الغير متجانسة .
 تصور مجموعة كائنات من نوع 01 و 02 ، و نرغب فقط في تصفح عناصر النوع 01 فإن
 الطريقة الثانية هي الأنجع .

مثال :

```
Dim oTmpObjet as Type01
For Each oTmpObjet In MaCollection
    العمليات المرغوب إنجازها '
Next oTmpObjet
```

- 2.2.4 إضافة عنصر إلى المجموعة

المجموعات في DAO مجهزة بالطريقة **Append** التي تمكنها من إضافة عنصر إلى المجموعة
 تركيب :

```
MaCollection.Append MonObjet
```

يجب التأكيد أنه عند إضافة عنصر جديد إلى المجموعة لا يمكن استخدامه إلا بعد إنشاء نسخة instance عنه ، وبالمثل لا يمكن إضافة عنصر إلى المجموعة إذا كان في هذه المجموعة عنصر آخر يحمل الاسم نفسه . وستصادف العديد من الأمثلة عن الطريقة **Append** خلال تصفحك لهذا الكتاب .



- 2.2.5 حذف عنصر ضمن مجموعة

الطريقة **Delete** عند تطبيقها على المجموعات في DAO تمكن من حذف العنصر المرسل اسمه كقيمة للمعلمة
 تركيب :

```
MaCollection.Delete "MonItem"
```

القسم الثالث :

- الاتصال بقاعدة بيانات

هذه القسم بكامله سيعنى بكيفيات فتح قاعدة بيانات ، ولذلك سأعتني بشرح الطرق و الخصائص المتعلقة بكل كائن من الكائنات المستعملة في الاتصال بقاعدة البيانات و سأضع الكثير من أمثلة الاستعمال ضمن لغة VBA

3.1 - الكائن DBEngine

3.1.1 - عموميات

الكائن **DBEngine** : ويتعلق الاسم بمحرك قاعدة البيانات . إنه عبارة عن مركب برنمي (logiciel) لمعالجة وتأويل الاستعلامات المرسله إليه ، و محاورة ملف قاعدة البيانات . و هو لا يحتاج إلى إنشاء نسخة (Instance) ولا حتى مراقبة كيفية اشتغاله ما دام كائن تطبيق الأكسيس **Access.Application** يمتلك ضمنا الخاصية : **DBEngine** ، ترجع قيمة محرك قاعدة البيانات قيد الاستعمال .

يقترح الكائن **DBEngine** من ضمن خصائصه قائمة بمجالات العمل (تنتمي للمجموعة : **Workspaces**) ، و أيضا مجموعة خواص أخرى تتعلق بتدبير شؤون المستخدمين هذه الخواص ليست ذات أهمية في حالة الاستخدام العادي (أي دون تدبير الأمان) . ومع ذلك لا يجب إهمال محرك قاعدة البيانات **DBEngine** لأنه يظل المسؤول عن تغليف encapsulation طرائق الجلسات transactions وكذا طرائق إصلاح قاعدة البيانات وضغطها .

3.1.2 - قائمة الخصائص :

سيتم التعرض فيما يلي لجميع خصائص **DBEngine** .

3.1.2.1 - الخاصية DefaultPassword

Type : String

يتعلق الأمر بكلمة مرور افتراضية ستستعمل عند إنشاء مجال عمل **Workspace** جديد

3.1.2.2 - الخاصية DelfaultType

Type : Long

هذه الخاصية تتعلق بالنوع الافتراضي لكائنات **Workspace** التي ستنشأ ، والقيم الممكنة لهذه الخاصية :

القيمة	مجال استعمالها
1	dbUseJet في حال استعمال محرك Jet
2	dbUseODBC في حال الاتصال المباشر بواسطة ODBC مباشر

يجب التأكيد أنه في حال الاتصال المباشر بواسطة ODBC مباشر؛ يعوض الكائن **Database** بكائن من نوع **Connection** . 

- 3.1.2.3 الخاصية *DefaultUser*

Type : String

يتعلق الأمر باسم المستخدم الذي سيعمل افتراضياً عند إنشاء كائن جديد **Workspace** وهذه الخاصية تستعمل في نفس الوقت مع الخاصية **DefaultPassword**

- 3.1.2.4 الخاصية *Errors*

Type : Collection

وهذه المجموعة تتضمن جميع أنواع الأخطاء التي قد يصادفها محرك قاعدة البيانات **DBEngine** الختاً في **DAO** معرف بقيمة رقمية ترتبط بوصف نصي مختصر عن الختأ هو الذي يظهر في مضمون رسالة الختأ .

في كل مرة تفشل عملية (أو إجراء **DAO**) ، ينضاف إلى المجموعة **Errors** كائن أو عدة كائنات **Error** ، ثم يتولى **VBA** بعد تمام الإجراء بإعلان فشل العملية ، ويرفع بيانات عن الختأ ذي الرقم الأعلى بين الأخطاء المتراكمة في المجموعة **Errors**

في حال ختأ **DAO** ، كائن **Error** وحيد ينشأ لتمثيل الختأ ، والختأ الذي يعلنه **VBA** هو نفسه الختأ الذي يعلنه محرك قاعدة البيانات ، ففي محاولة الاتصال بقاعدة بيانات **Oracle** مثلاً من خلال تطبيق الأكسيس ، يمكن أن تحدث أخطاء متعددة بـ **Oracle** فينشئ مجموعة كائنات **Error** تمثل تلك الأخطاء تنضاف إلى كائنات **DBEngine.Errors** وستجد بهذه المجموعة :

مجموع أخطاء **Oracle**

كائن **Error** وحيد لـ **Odbc**

ولكن ختأ واحداً سيعلنه **VBA** .

إن تصفخ المجموعة **Errors** يسمح لك بتحسين وتيرة معالجة الأخطاء .


- 3.1.2.5 الخاصية *Inipath*

Type : String

الخاصية **Inipath** تتعلق بمسار ملف تهيئة محرك قاعدة البيانات ، إنه يتعلق عموماً بمدخل من ملف قاعدة التسجيل :

HKLM\Software\microsoft\Office10.0\Access\Jet4. (بالنسبة لأكسيس XP)

وهنا تختزن مجموعة خيارات داخلية لمحرك **Jet** مثل مسار ملف مجموعة العمل قيد الاستعمال **Workgroupe** .

وعلى الرغم من قلة أهمية معرفة هذا المسار بالنسبة للمطورين إلا أنه على جانب كبير منها بالنسبة لمحرك قاعدة البيانات . 

3.1.2.7 - الخاصية *Properties*

Type : Collection

وهي أيضا مجموعة تتضمن جميع الخصائص (**Property**) المتعلقة بمحرك قاعدة البيانات **DBEngine**

لتعميق المعرفة بالمجموعة **Properties** أحيك على الفقرة (4.2 الخواص النوعية)



3.1.2.8 - الخاصية *SystemDB*

Type : String

يتعلق الأمر هنا بمسار ملف مجموعة العمل ذي اللاحقة .mdw. والملف المستعمل افتراضيا هو المشار إليه بمدخل لملف .ini. أو بمدخل قاعدة التسجيل المخصص للخاصية **IniPath**

3.1.2.9 - الخاصية *Version*

للقراءة فقط .

Type : String

الخاصية **Version** ترجع رقم إصدار محرك Jet مثلا : **3.6**

3.1.2.10 - الخاصية *Workspaces*

Type : Collection

وهي مجموعة تتضمن كل كائنات مجال العمل المفتوحة و مجال العمل الافتراضي هو **Worspaces(0)** .

سيفصل القول في مجال العمل ضمن الفقرة : (3.2 الكائن Workspace)



3.1.3 - قائمة الطرائق

أقترح عليك فيما يلي قائمة بالطرائق امطبقة على كائنات الصنف **DBEngine**

3.1.3.1 - الطريقة *BeginTrans*

الطريقة **BeginTrans** تستعمل لبدء جلسة **transaction**
مثال :

DBEngine.BeginTrans

- 3.1.3.2 الطريقة CommitTrans

الطريقة **CommitTrans** توقف الجلسة مع تسجيل البيانات

مثال :

```
DBEngine.CommitTrans
```

- 3.1.3.3 الطريقة CompactDatabase

الطريقة **CompactDatabase** تنسخ وتضغط قاعدة البيانات . وسجل أن قاعدة البيانات يجب أن تغلق أولاً قبل إنجاز هذا الإجراء .

تركيب :

```
DBEngine.CompactDatabase olddb, newdb, [locale], [options], [password]
```

Olddb تتعلق ب ملف المصدر newdb يتعلق بملف الهدف وهو مخالف للمصدر ، والخيارات الأخرى تتعلق أساسا باللغة و الإصدارة والترميز (السنن) المستعمل وكلها ثابتة يوفرها الأكسيس لمحرك قاعدة البيانات .



مثال :

```
DBEngine.CompactDatabase "c:\bd1.mdb", "c:\bd2.mdb"
```

- 3.1.3.4 الطريقة CreateDatabase

الطريقة **CreateDatabase** تمكن من إنشاء قاعدة بيانات جديدة ، وهي ترجع الكائن من نوع **DAO.Database**.

تركيب :

```
Set database = DBEngine.CreateDatabase (name, locale, [options])
```

name : يتعلق بالإسم الكامل لقاعدة البيانات
Locale : ثابتة تمثل أسلوب الفرز المطبق (خاص باللغة المستعملة) ، **dbLangArabic** للغة العربية و **dbLangGeneral** للغات الأوروبية .
أما الأخيرة فهي اختيارية وتخص ميزات إضافية كتخصيص الترميز و إصدار قاعدة البيانات ...

مثال :

```
Set db = DBEngine.CreateDatabase("c:\mabase.mdb", dbLangArabic)
```

3.1.3.5 - الطريقة CreateWorkspace

هذه الطريقة CreateWorkspace تسمح بإنشاء مجال عمل جديد **Workspace** وترجع الكائن المنشأ .

تركيب :

```
Set workspace = DBEngine.CreateWorkspace(name, user, password, [type])
```

Name : اسم الكائن مجال العمل، ولا يسمح إلا باسم فريد ضمن المجموعة **Workspaces** .
User : تمثل اسم المستخدم لهذا المجال ، و **password** لكلمة السر (14 حرفا كحد أقصى) و
ثنائي التوثيق هذا ، يجب أن يكون مدونا بالملف **.mdw**. المعين في الخاصية **SystemDB** وإلا فإن خطأ
سيعلنه محرك قاعدة البيانات .
أما المعلمة الأخيرة فهي اختيارية و تخص نوع المجال الذي سينشأ ، وفي حال عدم التصريح به ،
سيكون هو النوع الافتراضي للخاصية **DefaultType** .

مثال :

```
Set Wks = DBEngine.CreateWorkspace("MonEspace", "Toto", "mdp#@#!mJ")
```

3.1.3.6 - الطريقة Idle

الطريقة **Idle** تعطي إمكانية للمحرك **JET** أن يعالج مهامه الموضوعه بالانتظار ، ويلاحظ أن
هذه الطريقة تتضمن معلمة اختيارية ، إن ساوت القيمة **dbRefreshCache** سيعاد إنعاش الذاكرة بالبيانات
الأكثر حداثة في قاعدة البيانات .

مثال :

```
DBEngine.Idle
```

3.1.3.7 - الطريقة OpenDatabase


بالضبط كما يعني الاسم فإن هذه الطريقة تمكن من فتح قاعدة بيانات ، وإرجاع نوعها
DAO.Database

تركيب :

```
Set database = workspace.OpenDatabase (dbname, options, read-only, _  
connect)
```

Dbname : تمثل مسار الوصول إلى قاعدة البيانات ، هذه المعلمة يمكن أيضا أن يكون اسما
لمصدر بيانات **ODBC** ، وإذا كانت هذه المعلمة فارغة ، ومتغيرة الاتصال = **"ODBC"** فإن صندوق حوار
سيفتح أليا للمساءلة عن اسم ومسار مصدر البيانات .
أما المعلمة **option** : فتمكن من تحديد ما إذا كانت القاعدة ستفتح في الوضع المشترك فتأخذ القيمة

(False) أو الوضع الخاص وتأخذ القيمة (True)
 read-only : تفتح القاعدة في وضع للقراءة فقط إذا ساوت هذه المعلمة (True)
 و أخيرا المعلمة connect : و تستعمل لتخصيص نوع القاعدة وكذا كلمة سر النظام المستعملة لفتح ملف القاعدة .


الكائن Database المنشأ يضاف آليا إلى مجال العمل Workspaces(0)	
---	---

مثال :

```
Set db = DBEngine.OpenDatabase("c:\bd1.mdb", False, False, _
  "MS Access;PWD=mdp#@#!e)
```

3.1.3.8 - الطريقة RepairDatabase

الطريقة RepairDatabase تستدعى لإصلاح قاعدة البيانات .

يجب التنويه أن هذه الطريقة مدجوزة للكائن DBEngine للتحقق من التوافق مع الإصدارات السابقة ، و من المنصوح به استعمال الطريقة CompactDatabase بديلا عن هذه .	
---	---

3.1.3.9 - الطريقة Rollback

هذه الطريقة تستعمل لنتهي جلسة فتحت بالطريقة BeginTrans .

مثال :

```
DBEngine.Rollback
```

3.2 - الكائن Workspace

3.2.1 - عموميات

- يمكن أن نميز صنفين من مجالات العمل :
- الصنف الأول : **مجال العمل Jet** وهو مجال يستعمل فيه محرك قاعدة البيانات Microsoft Jet للوصول إلى مصدر البيانات وهذا المصدر يمكن أن يأخذ شكل ملف :
Microsoft Jet (.mdb)
ODBC
Paradox
ISAM
 - الصنف الثاني : **مجال العمل ODBC Direct** وهو تقنية للوصول إلى البيانات مباشرة من غير المرور عبر المحرك Jet.

فقط المجال الأول سأتناوله في هذا الكتاب



الكائن **Workspace** يمثل مجال عمل ينشأ في محرك قاعدة البيانات (**DBEngine**) ، إنه في الحقيقة جلسة منه تكرر لمستخدم بيانات ، إلا في حالة ما إذا رغبت تدبير فتح القاعدة في مجال مؤمن ، إن استعمال الكائن **Workspace** غير ذي أهمية ، فالكائن **Workspaces(0)** هو المجال الافتراضي للمحرك **DBEngine** ، فهو ينشأ تلقائياً بالأكسيس كلما استدعي الكائن DAO ، وكذا ، عموماً يمكنك استعمال مرجع تلقائي لهذا الكائن عندما تستعمل الطرائق : **CodeDB** أو **CurrentDB** المرتبطة بالكائن **Access.Application** .

ومع ذلك يظل الكائن **Workspace** مثله مثل الكائن **DBEngine** قادراً على تنفيذ طرائق الجلسة ، و أيضاً على استعمال (**Parameterization**) وتدبير أنواع كائن السجلات (**Recordset**) بالمحرك **Jet** . واستعماله يصبح أساسياً عندما نرغب باستعمال أفضل للوصول إلى البيانات .

وكما وُضح في الفقرة السابقة فإن الكائن **DBEngine** يظل مرجعاً لخلق كل كائن **Workspace** بواسطة الطريقة **CreateWorkspace** .

الطريقة **CreateWorkspace** تتطلب معرفاً وحيداً للكائن المنشأ . وإنشاء مثل هذه الكائنات نادر ، وأيضاً ، من الممكن استعمال متغير يحتوي تاريخ إنشاء الكائن كاسم لكائن **Workspace** المراد إنشاء نسخة عنه .



وهذا مثال عن إنشاء الكائن **Workspace** :

```
Dim oWks As DAO.Workspace
Set oWks = DBEngine.CreateWorkspace(Format(Now(), _
"yyyymmddhhnss"), "Utilisateur1", "mpd#ç#", dbUseJet)
```

3.2.2 - قائمة الخصائص :

ستجد تالية قائمة الخصائص المطبقة على كائن من صنف **Workspace** :

3.2.2.1 - الخاصية Databases

Type : Collection

مجموعة تضم كل الكائنات **Database** المفتوحة ضمن مجال العمل

3.2.2.2 - الخاصية IsolateODBCTrans

Type : Boolean

إذا ساوت هذه الخاصية القيمة **False** وهي القيمة الافتراضية ، فإنه بالإمكان تنفيذ عدة جلسات متزامنة على اتصال واحد مباشر ODBC ، ولأن قواعد البيانات SQL Server لا تقبل هذه الصيغة من التشغيل، فإن اتصال الـ ODBC بهذا النوع من قواعد البيانات يتطلب تثبيت هذه الخاصية في القيمة **True**

3.2.2.3 - الخاصية Name


Lecture seule.

Type : String

الخاصية **Name** تتعلق باسم الكائن **Workspace** ، وهذا الاسم وحيد في مجموعة الكائنات **DBEngine.Workspaces** .

3.2.2.4 - الخاصية Properties

كجميع الكائنات في DAO ، الكائن **Workspace** يمتلك مجموعة **Properties** تتضمن كل خصائص هذا الكائن .

مزيد من المعلومات حول مجموعة الخصائص ستجدها لاحقا في الفقرة (4.2) الخصائص (النوعية)	
---	---

3.2.2.5 - الخاصية Type

Lecture seule.

Type : Long

تحدد نوع مجال العمل ، قيمة هذه الخاصية تساوي تلك القيمة التي حددتها في طريقة الإنشاء **DBEngine.CreateWorkspace** ، فإذا لم يتم تعيين هذه القيمة أثناء الإنشاء ؛ فإنها ستساوي قيمة الخاصية **DefaultType** لمحرك قاعدة البيانات الكائن **Workspaces(0)** ينشأ تلقائياً بالأكسيس عند استدعاء كائن من كائنات DAO والخاصية **Workspaces(0).Type** إذ ذاك تساوي افتراضياً **dbUseJet** أما القيمة الأخرى الممكنة هي **dbUseODBC**

3.2.3 - قائمة الطرائق

فيما يلي قائمة بمجموعة الطرائق المنتمية للصف **Workspace** ومعظمها مشترك مع الصف **DBEngine**

- 3.2.3.1 الطريقة *BeginTrans*

الطريقة **BeginTrans** تستهل (أو تفتح) جلسة عمل

مثال :

```
DBEngine.Workspaces(0).BeginTrans
```

- 3.2.3.2 الطريقة *Close*

الطريقة **Close** تغلق مجال العمل المستدعاة منه .

سجل أنه عندما تغلق مجالا ، فإن جلسات العمل المرتبطة به و التي هي بالانتظار تندثر ، وبالمثل إذا كان الكائن **Workspace** يمتلك كائنات **Database** أو **Connection** فإنها تغلق معه ، مما ينجم عنه إغلاق مجموعة السجلات المرتبطة به أيضا .



إغلاق كائن **Workspace** يجب أن يكون حدثا مفكرا بعواقبه ، إذ قد يتسبب بخطر فقدان

البيانات



مثال :

```
DBEngine.Workspaces(0).BeginTrans
```

- 3.2.3.3 الطريقة *CommitTrans*

الطريقة **CommitTrans** توقف الجلسة مع تسجيل البيانات

مثال :

```
DBEngine.Workspaces(0).CommitTrans
```

- 3.2.3.4 الطريقة *CreateDatabase*

الطريقة **CreateDatabase** تمكن من إنشاء قاعدة بيانات جديدة ، وإرجاع كائن من نوع

DAO.Database

مثال :

```
Set database = Workspace.CreateDatabase (name, locale, [options])
```


المعلمة **name** : تتعلق بالاسم الكامل لقاعدة البيانات .
 المعلمة **locale** : ثابتة تمثل نظام الفرز المطبق (خاص باللغة المستعملة) ، **dbLangArabic** ،
 اللغة العربية و **dbLangGeneral** للغات الأوربية .
 أما المعلمة الأخيرة **options** : فهي اختيارية وتخص ميزات إضافية كتخصيص الترميز و إصدار
 قاعدة البيانات

مثال :

```
Set db = DBEngine.Workspaces(0).CreateDatabase("c:\mabase.mdb",  
dbLangArabic)
```

3.3.3.5 - الطريقة *OpenDatabase*

بالضبط كما يعني الاسم فإن هذه الطريقة تمكن من فتح قاعدة بيانات ، وإرجاع نوعها
DAO.Database

تركيب :

```
Set database = workspace.OpenDatabase (dbname, options, read-only, _  
connect)
```

Dbname : تمثل مسار الوصول إلى قاعدة البيانات ، هذه المعلمة يمكن أيضا أن يكون اسما
 لمصدر بيانات **ODBC** ، وإذا كانت هذه المعلمة فارغة ومتغيرة الاتصال = "**ODBC**" فإن صندوق حوار
 سيفتح للمساءلة عن اسم ومسار مصدر البيانات .
 أما المعلمة **option** : تمكن من تحديد إذا ما كانت القاعدة ستفتح في الوضع المشترك فتأخذ القيمة
(False) أو الوضع الخاص وتأخذ القيمة **(True)**
read-only : تفتح القاعدة في وضع للقراءة فقط إذا ساوت هذه المعلمة **(True)**
 و أخيرا المعلمة **connect** يستعمل لتخصيص نوع القاعدة وكذا كلمة سر النظام المستعملة لفتح
 ملف القاعدة .

الكائن **Database** المنشأ يضاف آليا إلى المجموعة **Databases** التي هي من
 خصائص المجال **Workspaces** بعد إنشائه .



مثال :

```
Set db = DBEngine.Workspaces(0).OpenDatabase("c:\bd1.mdb", _  
False, False, "MS Access;PWD=mdp#@#le")
```

3.2.3.6 - الطريقة *Rollback*

هذه الطريقة تستعمل لنتهي جلسة عمل فتحت بالطريقة **BeginTrans** .

مثال :

```
DBEngine.Workspaces(0).RollBack
```


3.2.4 - الولوج إلى مجموعة عمل منصبة

كما أشير إليه أعلاه ، الكائن **Workspace** يصبح ذا أهمية عندما يتعلق الأمر بفتح قاعدة بيانات مؤمنة ، ففي الواقع عندما تهم بفتح ملف **mdb** بواسطة الأكسيس ، يتم تعريفك (توثيق بياناتك) كمستخدم ضمن ملف مجموعة العمل (هذا الملف يمكن أن يكون افتراضيا **System.mdw** في حال لم تصرح بتدبير الأمان ، أو هو ملف من إنشائك في الحالة الأخرى) يجمع المعلومات الضرورية لتأمين قاعدة بياناتك ، غير أنه يمكن أن نحتاج الوصول إلى القاعدة بيانات أخرى بواسطة **DAO** وهذا يتطلب استعمال ملف **mdw** . آخر .

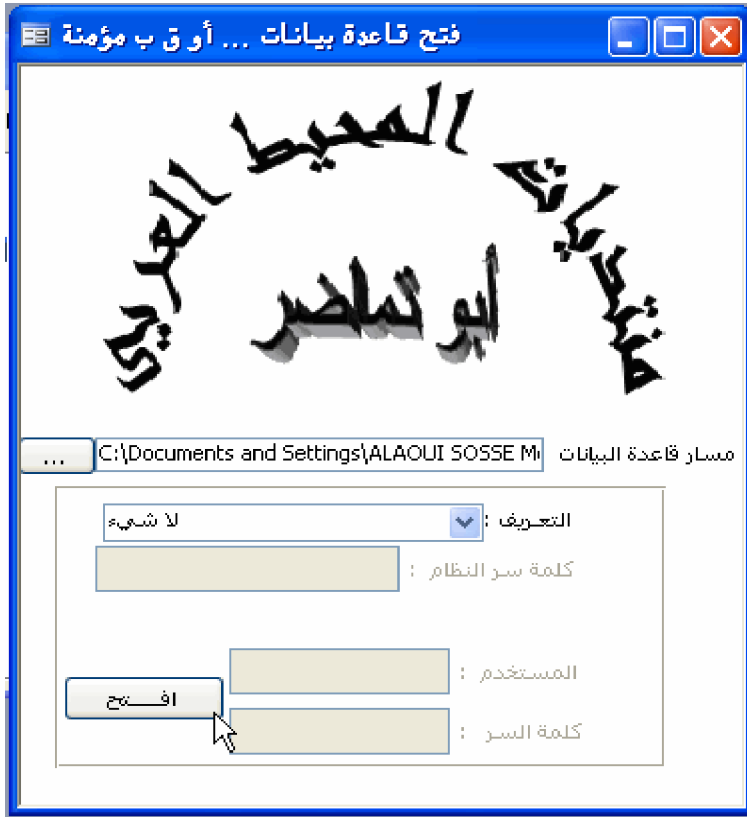
للاتصال في هذه الحال ، يلزم إنشاء نسخة لكائن آخر من نوع **DAO.DBEngine** ، وتعيين

المسار لملف مجموعة العمل المستعمل في الخاصية **DBEngine.SystemDB**

لكن هذه الطريقة تبقى أمام عائق كبير ؛ ففي الواقع ، إذا حاولت إنشاء نسخة جديدة من الكائن **DBEngine** ، وكنت أنفا قد استعملت في الكود بعض طرائق مكتبة **DAO** ، فإنه يستحيل تغيير قيمة الخاصية **SystemDB** للكائن الجديد المنشأ **DBEngine** هذه التي تستقي قيمتها من قيمة نظيرتها بالتطبيق الحالي **Application.DBEngine.SystemDB** ، و مع ذلك لا يعلن أي خطأ مهما حاولت تمكين التغيير .

إنه إذن من اللازم أن يتم إنشاء نسخة الكائن **DBEngine** قبل أي نداء للمكتبة **DAO** يتضمن الطرائق **CodeDb** و **CurrentDb** 

و أقترح عليك إنشاء واجهة تمكّنك من فتح أي قاعدة بيانات مع إمكانية تعيين الملف **mdw** .
لننشئ النموذج التالي:



إنه يتألف بالتوالي من الأعلى من الكائنات التالية :

- مربع نص (Tfichier)
 - مربع قائمة منسدلة (ListeSecu)
 - مربع نص (Tadmin)
 - مربع نص (Tuser)
 - مربع نص (TMDP)
 - زر أمر (Commande3) لتمرير المسار
 - زر أمر (Commande10) لفتح القاعدة
- وهذا الكود :

كود زر الأمر افتح :

```
Private Sub Commande10_Click()
    ' يفتح ملفا مع الأمان المطلوب '
    On Error GoTo err
    Dim strfichier As String
    Dim oVDb As DAO.Database
    Dim chaine As String
    Dim oVWk As DAO.Workspace
    Dim oDbe As DBEngine
    Dim strFichierMDW As String
    If Len(TFichier) <> 0 Then

        strfichier = TFichier
        ' يهيء مجال العمل '
        If ListeSecu.ListIndex = 0 Or ListeSecu.ListIndex = 2 Then
            Set oVWk = Workspaces(0) Else
            strFichierMDW = OuvrirUnFichier(Me.HWnd, "اختر ملف مجال العمل المطلوب", 1, _
                "Fichier mdw", "mdw")
            If strFichierMDW = "" Then
                Exit Sub
            Else
                Set oDbe = New DBEngine oDbe.SystemDB = strFichierMDW
                Set oVWk = oDbe.CreateWorkspace(Format(Now(), _
                    "yyyymmddhhnnss"), TUtilisateur, TMDP, dbUseJet)
            End If
        End If
        ' فتح قاعدة البيانات '
        If ListeSecu.ListIndex = 2 Or ListeSecu.ListIndex = 3 Then
            Set oVDb = oVWk.OpenDatabase(strfichier, False, False, _
                "MS Access;PWD=" & TAdmin)
        Else
            Set oVDb = oVWk.OpenDatabase(strfichier)
        End If
    Else

```

```

MsgBox "تحديد الملف", vbExclamation, "يجب أن تحدد ملفا"
End If
Exit Sub err:
MsgBox " & _
تأكد من صحة المسار ، وكذا من صحة" & " فشل الاتصال بقاعدة البيانات المعينة" & _
" خطأ في الاتصال...", vbCritical, "بيانات الأمان المدخلة"
End Sub
    
```

كود حدث عند التحميل للنموذج:

```

Private Sub Form_Load()
    'إفراغ الأدوات '
    Dim ctlT As Control
    For Each ctlT In Me.Controls
    If TypeOf ctlT Is TextBox Then ctlT.Value = ""
    If ctlT.Name <> "TFichier" Then ctlT.Enabled = False
    End If
    Next ctlT
    ListeSecu = "لا شيء"
End Sub
    
```

كود زر أمر تمرير المسارات :

```

Private Sub Commande3_Click()
    Dim strChemin As String
    strChemin = OuvrirUnFichier(Me.Hwnd, " اختر قاعدة بيانات Access ", _
    1, "ملفات Access", "mdb")
    If strChemin <> "" Then
    Me.TFichier = strChemin
    End If
End Sub
    
```

كود استدعاء صندوق الحوار فتح ملف :

" قد أحال صاحب الكتاب على قسم أسئلة وحلول بالموقع الذي ينشر فيه مقالاته ، وقد وجدت الكود التالي لصاحبه shwin به دالة تقترح معلمات متعددة لتخصيص صندوق الحوار (فتح ملف) و يجب أن يوضع في وحدة نمطية:"

```

' تصريحات API
Private Declare Sub PathStripPath Lib "shlwapi.dll" Alias "PathStripPathA"
(ByVal pszPath As String)
Private Declare Function GetOpenFileName Lib "comdlg32.dll" Alias _
"GetOpenFileNameA" (pOpenfilename As OPENFILENAME) As
Long
    
```

هيكلية الملف

Private Type OPENFILENAME

IStructSize As Long
 hwndOwner As Long
 hInstance As Long
 lpstrFilter As String
 lpstrCustomFilter As String
 nMaxCustFilter As Long
 nFilterIndex As Long
 lpstrFile As String
 nMaxFile As Long
 lpstrFileName As String
 nMaxFileName As Long
 lpstrInitialDir As String
 lpstrTitle As String
 flags As Long
 nFileOffset As Integer
 nFileExtension As Integer
 lpstrDefExt As String
 lCustData As Long
 lpfnHook As Long
 lpTemplateName As String

End Type

الثوابت

Private Const OFN_READONLY = &H1
 Private Const OFN_OVERWRITEPROMPT = &H2
 Private Const OFN_HIDEREADONLY = &H4
 Private Const OFN_NOCHANGEDIR = &H8
 Private Const OFN_SHOWHELP = &H10
 Private Const OFN_ENABLEHOOK = &H20
 Private Const OFN_ENABLETEMPLATE = &H40
 Private Const OFN_ENABLETEMPLATEHANDLE = &H80
 Private Const OFN_NOVALIDATE = &H100
 Private Const OFN_ALLOWMULTISELECT = &H200
 Private Const OFN_EXTENSIONDIFFERENT = &H400
 Private Const OFN_PATHMUSTEXIST = &H800
 Private Const OFN_FILEMUSTEXIST = &H1000
 Private Const OFN_CREATEPROMPT = &H2000
 Private Const OFN_SHAREAWARE = &H4000
 Private Const OFN_NOREADONLYRETURN = &H8000
 Private Const OFN_NOTESTFILECREATE = &H10000
 Private Const OFN_SHAREFALLTHROUGH = 2
 Private Const OFN_SHARENOWARN = 1
 Private Const OFN_SHAREWARN = 0

```
Public Function OuvrirUnFichier(Handle As Long, _
    Titre As String, _
    TypeRetour As Byte, _
    Optional TitreFiltre As String, _
    Optional TypeFichier As String, _
    Optional RepParDefaut As String) As String
```

' OuvrirUnFichier هي الدالة المستعملة في النموذج من أجل فتح صندوق حوار لاختيار ملف
' شرح المعلمات :

'Handle = (Me.Hwnd) مقبض النافذة

'Titre = عنوان صندوق الحوار

'TypeRetour (يحدد القيمة من نوع : String ، ترجعها الدالة)

'1 = المسار الكامل + اسم الملف

'2 = اسم الملف فقط

'TitreFiltre = Fichier Access: عنوان المصفوفة ، مثلا

' لا تستعمل هذه المعلمة إذا لم ترغب بتخصيص أي مصفوفة

'TypeFichier = MDB : لاحقة الملف من غير نقطة الفصل مثلا

' لا تستعمل هذه المعلمة إذا لم ترغب بتخصيص أي مصفوفة

'RepParDefaut = C:\windows\system32 يفتح افتراضيا مثلا على المجلد

' إذا تركت هذه المعلمة فارغة فإنه تلقائيا سيفتح على المجلد الذي تقع به قاعدة بياناتك

```
Dim StructFile As OPENFILENAME
```

```
Dim sFiltre As String
```

' إنشاء المصفوفة بالمعلمات المخصصة

```
If Len(TitreFiltre) > 0 And Len(TypeFichier) > 0 Then
```

```
    sFiltre = TitreFiltre & " (" & TypeFichier & ")" & Chr$(0) & "*" & TypeFichier & Chr$(0)
```

```
End If
```

```
sFiltre = sFiltre & "Tous (*.*)" & Chr$(0) & "*.*" & Chr$(0)
```

' تخصيص صندوق الحوار

```
With StructFile
```

```
.IStructSize = Len(StructFile) ' إنشاء نسخة (تهييء) حجم البنية
```

```
.hwndOwner = Handle ' تكنه مقبض النافذة
```

```
.lpstrFilter = sFiltre ' تطبيق المصفوفة
```

```
.lpstrFile = String$(254, vbNullChar) ' '0' x 254 تهييء الملف
```

```
.nMaxFile = 254 ' حجم الملف الأقصى
```

```
.lpstrFileTitle = String$(254, vbNullChar) ' '0' x 254 تهييء اسم الملف
```

```
.nMaxFileTitle = 254 ' الحجم الأقصى لاسم الملف
```

```
.lpstrTitle = Titre ' عنوان صندوق الحوار
```

```
.flags = OFN_HIDEREADONLY ' خيارات صندوق الحوار
```

```
If ((IsNull(RepParDefaut)) Or (RepParDefaut = "")) Then
```

```
    RepParDefaut = CurrentDb.Name
```

```
    PathStripPath (RepParDefaut)
```

```
    .lpstrInitialDir = Left(CurrentDb.Name, Len(CurrentDb.Name) -
```

```
Len(Mid$(RepParDefaut, 1, InStr(1, RepParDefaut, vbNullChar) - 1)))
```

```
Else
```

```
    .lpstrInitialDir = RepParDefaut
```

```
End If
```

```

End With
If (GetOpenFileName(StructFile)) Then ' إذا تم اختيار ملف
Select Case TypeRetour
Case 1: OuvrirUnFichier = Trim$(Left(StructFile.lpstrFile, InStr(1, _
StructFile.lpstrFile, vbNullChar)-1))
Case 2: OuvrirUnFichier = Trim$(Left(StructFile.lpstrFileTitle, InStr(1, _
StructFile.lpstrFileTitle, vbNullChar)-1))
End Select
End If
End Function

```

قاعدة البيانات المطلوب فتحها بهذا النموذج ستكون عبر الكائن **VDB** من إجراء عند النقر للزر (افتح) ، هذه الواجهة المقترحة في هذه الفقرة ، تسمح من جهة بفتح قاعدة بيانات ، ومن جهة أخرى تسمح بتعريف المستخدم ضمن ملف مجموعة العمل ، وإدخال كلمة سر النظام لفتح الكائن **Database** اختيار ملف مجموعة العمل يتم بواسطة الكود التالي :

```

Set oDbc = New DBEngine
oDbc.SystemDB = strFichierMDW
Set VWK = oDbc.CreateWorkspace(Format(Now(), _
"yyyymmddhhnss"), TUtilisateur, TMDP, dbUseJet)

```

السطر الأول ينشئ نسخة من الكائن **DBEngine** ، والذي تعين خاصيته **SystemDB**. بقيمة الملف **MDW** بالسطر الثاني ، ثم ننشئ أخيراً مجموعة العمل باستخدام الطريقة **CreateWorkspace** أحد الأسباب الرئيسية التي تجعل محاولة إنشاء مجموعة العمل ترفع خطأ الزوج : مستخدم- الرقم السري يفشل تعريفه بالملف **mdw** المختار ، ويمكن أيضاً أن يفشل الإسناد إلى الخاصية **SystemDB** مما يدل أن البرنامج استعمل طرائق **DAO** قبل هذه الكتلة من أسطر الكود ، وللتحقق من ذلك ننفذ الكود التالي :

```

MsgBox oDbc.SystemDB
oDbc.SystemDB = FichierMDW
MsgBox _
oDbc.SystemDB

```

فإذا كانت النتيجة في صندوق الحوار متماثلة فهذا يعني أن البرنامج استعمل طرائق **DAO** من قبل

3.3 - الكائن Database

3.3.1 - معمومات

كما يمكن أن تكون قد لاحظت في نافذة قاعدة البيانات للأكسيس أن الملف `mdb` يمكن أن يقسم إلى قسمين :

- قسم إجرائي به النماذج والتقارير والوحدات النمطية ووحدات الماكرو
- وقسم للبيانات : ، وهذا القسم **SGBD** (نظام تدبير قاعدة البيانات) توجد به الجداول والاستعلامات ، وحيث إن أكسيس نظام علائقي **SGBDR** فهذا القسم أيضا توجد العلاقات النموذج الكائني **DAO** يسمح بالوصول إلى قسم (البيانات) بواسطة الكائن **Database** ، وهذا الكائن هو في الأصل يجمع بين سمة **DDL** (لغة تعريف البيانات) و **DML** (لغة معالجة البيانات) ، وبواسطة هذا الكائن أيضا يتم إنشاء الجداول والاستعلامات و كذا العلاقات ومعالجة البيانات (إضافة/حذف/تحديث..) و كما ذكر آنفا الكائن **Database** يمكن إنشاء نسخة عنه بواسطة محرك قاعدة البيانات (**DBEngine**) ومن خلال هذه بواسطة محرك العمل (**Workspace**) بمساعدة الطرائق **CreateDatabase** و **OpenDatabase** . وهذا يتم إما بالتصريح :

```
Dim oDb As DAO.Database
Set oDb=DBEngine.CreateDatabase (#)
```

أو بالتضمين :

```
Dim oDb As DAO.Database
Set oDb=CreateDatabase(#)
```

إلا أنه في غالب الأحيان ترغب في استعمال قاعدة البيانات الحالية المفتوحة في الأكسيس ، وهنا يكفي أن تستعمل الطرائق **Application.CurrentDb** أو **Application.CodeDb**

3.3.1.1 - الطريقة CurrentDb

هذه الطريقة المضمنة في تطبيق الأكسيس ، ترجع متغيرة من نوع **Database** مطابقة لقاعدة البيانات الحالية المفتوحة في تطبيق أكسيس .
والطريقة **Currentdb** تنشئ نسخة عن قاعدة البيانات عدد كل نداء ، وأيضا حينما تستعيد نتيجة المتغيرة من نوع **Database** يجب استغلالها بشكل جيد ، وإليك ما لا يجب فعله :

```
Dim Tbl As DAO.TableDef, Tbl2 As DAO.TableDef
Set Tbl = CurrentDb.TableDefs("Table1")
Set Tbl2 = CurrentDb.TableDefs("Table2")
```

في الواقع هذا الكود يستدعي الطريقة **Currentdb** مرتين مما يؤدي إلى إنشاء نسختين من قاعدة البيانات الحالية . والأفضل مرة واحدة فقط في مثل هذه الحالة :


```
Dim Tbl As DAO.TableDef, Tbl2 As DAO.TableDef
Dim Db as DAO.Database
Set Db=CurrentDb
Set Tbl = Db.TableDefs("Table1")
Set Tbl2 = Db.TableDefs("Table2")
```

الخاصية **DBEngine.Workspaces(0).Databases(0)** تمثل أي ضاقاعة البيانات المفتوحة ، غير أن هذه الطريقة غير منصوح بها ، لأنها لا تنشئ نسخة جديدة من الكائن **Database** وعلى العكس ، فنتيجتها يمكن اعتبارها مشيراً (pointer) إلى قاعدة البيانات ، أيضا وفي مجال عمل متعدد المستخدمين هذا يمكن أن يتسبب في تنازع يؤدي إلى فشل الوصول إلى البيانات ، ومن الأفضل إذن أن تستدعي الطريقة **CurrentDb**

3.3.1.2 - الطريقة **CodeDb**

وعكس الطريقة **CurrentDb** التي ترجع قاعدة البيانات الحالية ، فإن الطريقة **CodeDb** ترجع كائن **Database** متوافق مع قاعدة البيانات التي تتضمن الكود قيد الانجاز .
CodeDb تستعمل في حال إنشاء قاعدة بيانات ملحقة أو قاعدة بيانات مكتبة . والواقع أن القاعدة الملحقة **Add-in** لن تكون القاعدة الحالية . والطريقة **CurrentDb** لن تعيد الكائن **Database** المراد .
 كلا الطريقتين **CodeDb** و **CurrentDb** تنشئ نسخة عن الكائن **Database** وعليه يجب تخزين نتيجته في كائن **Database** ليتمكنك استغلاله .

```
Dim Tbl As DAO.TableDef, Tbl2 As DAO.TableDef
Dim Db as DAO.Database
Set Db=CodeDb
Set Tbl = Db.TableDefs("Table1") Set Tbl2 =
Db.TableDefs("Table2")
```

3.3.2 - قائمة الخواص :

3.3.2.1 - الخاصية **CollatingOrder**

Type : Long

إنها خاصية يستعملها المحرك للمقارنة بين البيانات من نوع (نص) ويمكن أن تأخذ القيم التالية :

الثابتة	دلالاتها
dbSortGeneral	عام (انجليزية ، فرنسية ... لغات غرب أوربية)
dbSortArabic	اللغة العربية
dbSortChineseSimplified	اللغة الصينية
dbSortChineseTraditional	اللغة الصينية التقليدية
dbSortCyrillic	اللغة الروسية
dbSortCzech	اللغة التشيكية
dbSortDutch	اللغة الهولندية

الثابتة	دلالتها
dbSortGreek	اللغة اليونانية
dbSortHebrew	اللغة العبرية
dbSortHungarian	اللغة الهنغارية
dbSortIcelandic	اللغة الأيسلاندية
dbSortJapanese	اللغة اليابانية
dbSortKorean	اللغة الكورية
dbSortNeutral	حيادية
dbSortNorwDan	اللغة النرويجية ، الدانماركية
dbSortPolish	اللغة البولندية
dbSortSlovenian	اللغة السلوفانية
dbSortSpanish	اللغة الإسبانية
dbSortSwedFin	اللغة السويدية ، الفنلندية
dbSortThai	اللغة التايلاندية
dbSortTurkish	اللغة التركية
dbSortUndefined	غير محددة

3.3.2.2 - الخاصية *Connect*

Lecture Seule.

Type : String

الخاصية **Connect** تمثل جملة الاتصال المستعملة عند فتح الكائن **Database** . عندما تستعمل هذه الخاصية في قاعدة من تنسيق أكسيس فإن قيمة هذه الخاصية تساوي "".

3.3.2.3 - الخاصية *Containers*

Type : Collection

تجمع في هذه المجموعة معلومات عن كائنات (الحاويات container) المسجلة بقاعدة البيانات ، فتجد بها مثلا بيانات عن النماذج والتقارير ... الخ

3.3.2.4 - الخاصية *Name*

Lecture Seule.

Type : String

الخاصية **Name** للكائن **Database** ترجع اسم قاعدة البيانات . و بالأكسيس مثلا، فإنها تأخذ قيمتها اسم ملف **mdb** ، وكذا مساره الكامل .

3.3.2.5 - الخاصية *Properties*

Type : Collection

مجموعة تتضمن كل الخصائص **Property** للقاعدة

3.3.2.6 - الخاصية QueryDefs

Type : Collection

هذه المجموعة تتضمن كل الاستعلامات المسجلة بالقاعدة

هذه الخاصية سيتوسع فيها في الفقرة (4.7 الكائن QueryDef)



3.3.2.7 - الخاصية RecordsAffected

Lecture Seule.

Type : Long

الخاصية **RecordsAffected** ، ترجع عدد السجلات التي تأثرت بأخر إجراء SQL في الطريقة **Execute** . وهذا ما يساعد على معرفة عدد السجلات المحذوفة من جراء استعلام حذف مثلا .

مثال لاستعمال هذه الخاصية :

```
Dim Requete As String
Dim DB As DAO.Database
Requete = "DELETE FROM MaTable WHERE ChampID>20"
Set DB = CurrentDb
DB.Execute Requete
MsgBox DB
```

3.3.2.8 - الخاصية Recordsets

للقراءة فقط

Type : Collection

هذه المجموعة تضم كل كائنات السجلات recordset المفتوحة بقاعدة البيانات .

هذا الجزء سوف يدرس بتوسع أكثر في القسم الخامس (الوصول للبيانات)



3.3.2.9 - الخاصية Relations

Type : Collection

الخاصية **Relations** هي مجموعة تضم كل الروابط الموجودة بقاعدة البيانات . هذه المجموعة ليست ذات فائدة إلا في قواعد البيانات العلائقية SGBDR

هذا الجزء سوف يدرس بتوسع أكثر في القسم الرابع (4-6 الكائن Relation)



3.3.2.10 - الخاصية *TableDefs*

Type : Collection

هذه المجموعة تضم كل الجداول المسجلة بقاعدة البيانات . وفي حالة الجداول المرتبطة ، يوجد كذلك كائن *TableDef* لكل جدول مرتبط .

هذا الجزء سوف يدرس بتوسع أكثر في القسم الرابع (3-4 الكائن *TableDef*)



3.3.2.11 - الخاصية *Transactions*

للقراءة فقط

Type : Boolean

الخاصية *Transactions* للكائن *Database* ، تبين إذا ما كانت قاعدة البيانات تدعم نظام الجلسات فإذا كان الأمر نعم ، فإنها تأخذ القيمة *True* . في حالة قاعدة البيانات أكسيس ، فإن قيمة هذه الخاصية ثابتة وتساوي *True* والجلسات متاحة دائما بواسطة كائنات السجلات *Recordset* و *Dynaset* و *Table* .

3.3.2.12 - الخاصية *Updatable*

للقراءة فقط

Type : Boolean

إذا ساوت قيمة هذه الخاصية *False* ، فإن قاعدة البيانات تكون محمية ضد الكتابة ، ولن تتمكن من تغيير لا بيانات ولا هيكل هذه القاعدة .

3.3.2.13 - الخاصية *Version*

للقراءة فقط

Type : String

هذه الخاصية ترجع رقم إصدار المحرك *Jet* المستعمل لبناء قاعدة البيانات . في حالة قاعدة بيانات أكسيس 2000 وما بعد فإن هذه الخاصية تساوي 4.0

3.3.3 - قائمة الطرائق

3.3.3.1 - الطريقة *Close* .

الطريقة *Close* تغلق قاعدة البيانات ، غير أن هذا لا يحرر الذاكرة المخصصة للكائن *database* ، ولفعل ذلك يجب إرسال القيمة *Nothing* إلى المتغيرة .

3.3.3.2 - الطريقة *CreateProperty*

الطريقة *CreateProperty* ، تنشئ وترجع كائنا من نوع *Property*

تركيب :

```
Set property = object.CreateProperty (name, type, value, DDL)
```

سإفصل القول في الكائن property في القسم الموالي



3.3.3.3 - الطريقة CreateQueryDef

الطريقة CreateQueryDef ، تنشئ استعلاماً جديداً وترجع كائناً من نوع QueryDef مطابقاً له . والكائن المنشأ يضاف ألياً إلى المجموعة Database.QueryDefs

تركيب :

```
Set querydef = Database.CreateQueryDef (name, sqltext)
```

المعلمة name : تتعلق باسم الاستعلام المراد إنشاؤه ، ويجب أن يكون اسماً فريداً لأن المجموعة QueryDefs لا تقبل التكرار . وعندما ترسل عبارة فارغة ("") لهذه المعلمة فإنك تنشئ كائن QueryDef مؤقت ولن يضاف إلى مجموعة الكائنات QueryDefs ، وفي هذه الحالة فإن الكائن المؤقت المنشأ سيندثر مع نهاية الأجراء قيد التنفيذ .
أما المعلمة sqltext : فهي عبارة نصية تتضمن أوامر SQL لإنشاء الاستعلام ، و يمكن أن يكون استعلام تحديد ، تحديث ، حذف ، ... الخ

اشتغال الكائن QueryDef موضح في الفقرة بعنوان (4.7 الكائن QueryDef)



3.3.3.4 - الطريقة CreateRelation

الطريقة CreateRelation ، تنشئ وترجع كائن Relation

تركيب :

```
Set relation = database.CreateRelation (name, table, foreigntable, _  
attributes)
```

3.3.3.5 - الطريقة CreateTableDef

الطريقة CreateTableDef ، تنشئ وترجع كائناً Tabledef ، الكائن فقط يندشأ ولا يندضم إلى المجموعة TableDefs بقاعدة البيانات ، و لإضافته لهذه المجموعة يجب أن تتوسل بالطريقة Append

تركيب :

```
Set tabledef = database.CreateTableDef (name, attributes, source, _connect)
```

إنشاء الجداول سيتم التوسع فيه بشكل مفصل في القسم الموالي (4.3 الكائن TableDef)



- 3.3.3.6 الطريقة Execute

هذه الطريقة تنفذ أمر SQL ، وهذا الأمر يجب أن يكون استعلاما إجرائيا (إلحاق ، تحديث ، حذف) . أما استعلام التحديد فإنه موكول لكائن السجلات (Recordset) وعند تمام إجراء أمر SQL ، فإنه من الممكن معرفة عدد السجلات التي شملها الإجراء بواسطة الإجراء RecordsAffected

تركيب :

```
Dim Db As DAO.Database
Set Db=CurrentDb
Db.Execute ("DELETE FROM MaTable")
```

- 3.3.3.7 الطريقة NewPassword

الطريقة NewPassword تمكن من تعديل (تغيير) كلمة سر النظام لقاعدة البيانات .

يجب عدم الخلط بين كلمة سر النظام ، وكلمة سر المستخدم التي تستعمل لفتح مجموعة العمل . فكل كلمة سر النظام هي تلك التي تحدد من خلال القائمة (أدوات / الأمان / تعيين كلمة مرور قاعدة البيانات)



مثال :

```
Dim Db As DAO.Database
Set Db=CurrentDb
Db.NewPassword "mdpold" , "mdpnew"
```

- 3.3.3.8 الطريقة OpenRecordset

الطريقة OpenRecordset ، تفتح وترجع كائن السجلات Recordset الذي يمكن من الوصول للبيانات والكائن المنشأ ينضم تلقائيا للمجموعة Recordsets

القسم الرابع :

- هيكلة قواعد البيانات في نموذج DAO

- 4.1.1.1 -

سنعرض في هذا القسم لنقطة مهمة في DAO : تعريف البيانات (DDL) . في الواقع الكائن Database يسمح بالوصول لكل الجداول ، الاستعلامات ، والروابط بقاعدة البيانات أكسيس . ليس فقط لتصفح و استعراض مختلف هذه الكائنات ، ولكن يمكن أيضا أن يسمح بالتعديل عليها بالإضافة أو الحذف و يمكنك إذن من خلال كود الـ VBA ، أن تشتغل على بنية ملفك وكأنك تشتغل على بيئة التطوير بالأوكسيس لديك . والفائدة طبعا هي إمكانية تغيير هذه البنية من غير أي تدخل للمستخدم (فتركك المستخدم يتفاعل مع الجداول مباشرة فيه خطر كبير على قاعدة البيانات) ، لهذا قواعد البيانات العلائقية الأخرى SGBD تستعمل عموما لغة SQL بأوامرها المعروفة (... Create, Alter, Drop) بينما أكسيس غير مهياً إلى درجة كبيرة للتعامل مع لغة SQL ، والاستعانة بلغة البرمجة VBA أصبح أمرا ملحا لإنجاز العديد من العمليات في أكسيس . وسنرى لاحقا في هذا القسم كيفية الاشتغال على الجداول والاستعلامات والروابط . لكن كخطوة أولى سنتعرض لموضوع الخصائص النوعية .

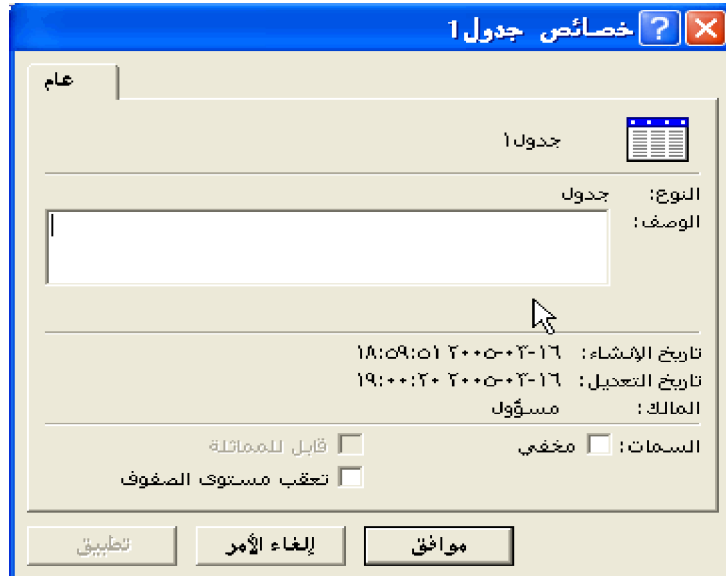
- 4.2 الخصائص النوعية

- 4.2.1 تعريفات

DAO هو نموذج كائني ، وأيضا كل كائن منه يمتلك قائمة من الخصائص بحيث كل واحدة منها تتعلق بكائن خاصية (Property) وقد يتبادر إلى ذهنك السؤال ، مافائدة أن تكون هناك مجموعة Properties ما دامت هذه الخصائص متاحة بكود VBA حسب التركيب (كائن خاصية Object.Property) ؟ بالتأكيد، الفيچوال بيزيك به قائمة من الخصائص ، لكنه لا يتيح عرضها جميعا ، فمثلا في قائمة خصائص الكائن TableDef خاصية Description التي تتعلق بوصف الجدول . وهذه الخاصية غير معروضة في إجمالي قائمة الخصائص لهذا الكائن ، والوسيلة الوحيدة للوصول إليه ، هي اسد تعامل المجموعة TableDef.Properties

```
Dim Db As DAO.Database
Dim Tbl As DAO.TableDef
Set Db = CurrentDb
Set Tbl = Db.TableDefs("MaTable")
Tbl.d
Tbl.d
DateCreated
Fields
Indexes
LastUpdated
Name
OpenRecordset
Properties
```

لماذا الكثير من الخصائص لا تعرض إلا عبر المجموعة properties ؟
 إن عددا لا بأس به من الخصائص ضمن المجموعة Properties لا تنشأ حتى ترسل إليها قيمة ،
 وبالتالي الخاصية description للكائن TableDef غير متاحة ، ولا حتى موجودة إذا كذت لم تدخل وصفا
 للجدول بصندوق الوصف في مربع حوار خصائص الأداة



بالإضافة إلى أنه يمكنك إنشاء خصائص جديدة وإضافتها إلى مجموعة Properties لكائن من اختيارك

ويظل مستحسبا تحقيق لائحة مرهقة لمجموعة Properties لكائن من DAO

4.2.2 - قائمة الخصائص لكائن الخاصية Property

كل كائن خاصية Property يمتلك مجموعة من الخواص

4.2.2.1 - الخاصية *Inherited* .

للقراءة فقط

عندما تساوي قيمة هذه الخاصية True فهذا يعني أن Property لا تخص الكائن نفسه ولكنها مورثة له عن كائن آخر .
تصور أن كائن QueryDef Q1 أضفنا له خاصية مخصصة (منشأة) P1
فإن P1 خاصية أنشئت لـ Q1 ، إذن الخاصية P1.Inherited تساوي False ، وإذا بعد ذلك
فتخذنا سجلات R1 recordset صدرها Q1 ، فإن الخاصية P1 تتضمن إلى مجموعة الخصائص
Properties لكائن السجلات R1 .
و إذن P1 تصبح مورثة : أي أن قيمة الخاصية P1.Inherited = True ، وفي هذه الحالة
العبارتان التاليتان يمكن التحقق منهما

```
Q1.Properties("P1").Inherited=False
```

```
R1.Properties("P1").Inherited=True
```

و

4.2.2.2 - الخاصية *Name*

هذه الخاصية تتعلق باسم الخاصية Property ، وهذا الإسم يجب أن يكون فريدا في مجموعة الخصائص
Properties للكائن المعني ، وإلا فإن خطأ سيعلم عنه كلما حاولت إضافة كائن جديد Property يحمل اسما
لكائن آخر Property موجود بالمجموعة Properties .

استخدم دائما عند تسمية الكائنات أسماء دقيقة من حيث علاقتها بالخاصية التي تعنيها ، وبالمقابل
تحاشى استعمال الأسماء المحجوزة للفيجوال بيزيك . فمثلا إن كانت الخاصية تمثل تاريخ المراجعة
سمها *DateVerification* وليس *Date* هكذا



4.2.2.3 - الخاصية *Type*

يتعلق الأمر بنوع الخاصية ، وقيمة هذه الخاصية يجب أن تكون واحدة من الثوابت التالية :

النوع	قيمته العددية	الثابتة
رقمي	16	dbBigInt
ثنائي بطول ثابت (255 بايت كحد أقصى)	9	dbBinary
Booléen	1	dbBoolean
أثمونة	2	dbByte
سلسلة أحرف (عبارة) بطول ثابت	18	dbChar
عملة	5	dbCurrency
تاريخ	8	dbDate
عشري	20	dbDecimal
مزوج	7	dbDouble
حقيقي	21	dbFloat
معرف النسخة المماثلة	15	dbGUID

النوع	قيمه العدديه	الثابته
عدد صحيح	3	dbInteger
عدد صحيح طويل	4	dbLong
ثنائي بطول متغير (OLE)	11	dbLongBinary
مذكرة	12	dbMemo
رقمي	19	dbNumeric
مفرد	6	dbSingle
نص بطول ثابت	10	dbText
توقيت	22	dbTime
TimeStamp	23	dbTimeStamp
ثنائي بطول متغير (255بايت كحد أقصى)	17	dbVarBinary

4.2.2.4 - الخاصية Value

هذه الخاصية تتعلق بقيمة الكائن Property وهذه القيمة يجب أن تكون متوافقة مع النوع المعين في الخاصية Type

4.2.3 - الوصول إلى خاصية

المجموعة Properties تشتغل ككل المجموعات في DAO غير أنه عندما لانعرف محتوى مجموعة يبدو من الصعب الوصول إلى أحد عناصرها عن طريق الفهرس ، ويجب عندها استخدام اسم العنصر المبحوث عنه . و يواجهنا مشكل آخر حين لانعلم بالضبط هل هذا الكائن المبحوث عنه موجود بالفعل ضمن المجموعة . وفي الواقع و كما سبق و ألمحنا إليه أعلاه بخصوص الخاصية description للكائن TableDef ، فهذه غير موجودة أصلاً حتى يتم تهيئتها ، وبالتالي فإنه يجب التحقق من وجود الخاصية قبل الوصول إليها .
طريقته ان فقط يمكن اسد تعامل إحداهما للتحقق من وجود الكائن Property ضمن المجموعة
: Properties

- الأولى تتطلب تمرير كائنات Property الموجهة بالمجموعة وإرجاع قيمة (صح/خطأ) عن الكائن المبحوث عنه

```
Public Function ExistProperty(ColTProperties As DAO.Properties, _
    strTNom As String) As Boolean
Dim oTmpPrp As DAO.Property
For Each oTmpPrp In ColTProperties
    If oTmpPrp.Name = strTNom Then
        ExistProperty = True
        Exit For
    End If
Next oTmpPrp
End Function
```

مثال للاستعمال :

```
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
MsgBox ExistProperty(oTbl.Properties, "description")
```

- الثانية : ونداول فيها الوصول إلى الكائن عبر استخدام اسمه مع معالجة الخطأ المرجع في حال عدم وجود الكائن :

```
Public Function ExistProperty(ColTProperties As DAO.Properties, _
    strTNom As String) As Boolean
On Error GoTo err
Dim oTmpPrp As DAO.Property
Set oTmpPrp = ColTProperties(strTNom)
ExistProperty = True
err:
End Function
```

استخدام دالة المثال الثاني تظل مثل دالة المثال الأول ، غير أن دالة المثال الثاني أسرع في التنفيذ لأنها لا تستخدم الحلقة التكرارية .
في حالة عدم استعمالك لأي من الدالتين السابقتين ، وكون الخاصية غير موجودة ضمن المجموعة فإن الخطأ 3270 (خاصية غير موجودة) سيعلنه النظام .
وفي حال تحقق وجود الخاصية ضمن المجموعة ، فإن قيمتها تسترد بالخاصية value

```
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
If ExistProperty(oTbl.Properties, "description") Then
    MsgBox oTbl.Properties("description").Value
Else
    MsgBox "هذه الخاصية غير متوفرة"
End If
oDb.Close
Set oDb=Nothing
Set oTbl=Nothing
```

4.2.4 - التعديل في قيمة خاصية

للتعديل في قيمة كائن Property يجب أولا التحقق من وجود الخاصية ضمن المجموعة ، ويمكن أن نستعين بواحدة من الدالتين أعلاه ، ثم بعد ذلك وبكل بساطة نرسل القيمة الجديدة إلى الخاصية Value للكائن Property .
يجب أن ننتبه إلى أن العديد من الخصائص هي للقراءة فقط ، مما يجعل معالجة الأخطاء أمرا ضروريا .
وهذا مثال عن دالة استعملت للتعديل على قيم خصائص نوعية لكائن في DAO:

```
Public Function ModifProperty(colTProperties As DAO.Properties, _
    strTNom As String, strTValeur As Variant) As Boolean
On Error GoTo err
Dim oTmpPrp As DAO.Property
' يبحث عن العنصر
Set oTmpPrp = colTProperties(strTNom)
' يعدل القيمة
oTmpPrp.Value = strTValeur
' ينعش المجموعة
```

```


colTProperties.Refresh
' إن العملية تمت بنجاح يرجع القيمة True
ModifProperty = True
Exit Function
' يعالج الأخطاء
err:
Select Case err.Number
Case 3270
    MsgBox "الخاصية غير موجودة"
Case 3421
    MsgBox "نوع الخاصية غير متوافق"
Case Else
    MsgBox "الخاصية للقراءة فقط"
End Select
End Function
    
```

طريقة الاستعمال :

```

Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
ModifProperty oTbl.Properties, "description", "essai"
    
```

بعض الملاحظات حول هذه الدالة :
 Tvalueur تتعلق بالقيمة الجديدة للكائن Property . ونوع هذه المعلمة متغيرة ؛ للتكيف مع كل الأتواع الممكنة ، غير أنه في حال رغبتك مثلا ألا تعدل سوى الخواص من نوع String يمكنك تعويض نوع المعلمة Variant ، String .
 الطريقة Refresh ، المطبقة على المجموعة Properties ، تمكن من التحديث الآني للخاصية ، وإذا لم تستخدم هذه الطريقة ، فإن التعديلات المجراة على قيم الخصائص لن تصبح فعالة إلا بعد إقلاع التطبيق من جديد .
 الدالة ترجع القيمة False ، في حال فشل التعديل على قيمة الخاصية .

إنه من غير المجدي استعمال هذه الدالة في حال الخصائص غير النوعية ، ففي الواقع الخصائص المقترحة بواسطة مستكشف الكائنات (يستدعى من محرر VBA / بيئة تطوير الأكسيس / والضغط على F2) هي كلها خصائص مندمجة ، ويمكن إجراء التعديل على قيمها بالتركيب المعهود . Object. Property	
---	---

مثال :

```

Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
oTbl.name= "essai"
    
```

4.2.5 - إنشاء خاصية جديدة

لإنشاء خاصية جديدة ، يجب التأكد أولا من عدم وجودها ضمن المجموعة ، ثم يكفي تطبيق الطريقة CreateProperty للكائن المعني ، لتهيئ كائن Property . وعدد هذه الخطوة فإن الخاصية الجديدة تكون قد أنشئت ولن تنضم إلى المجموعة إلا بتطبيق الطريقة Append ، للمجموعة Properties .

وهذا مثال عن إضافة خاصية نسميها "test" من نوع Integer إلى الجدول commande

```
Sub ajout()
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Dim oPrp As DAO.Property
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
' يختبر عدم وجود الخاصية
If Not ExistProperty(oTbl.Properties, "test") Then
' ينشئ الخاصية الجديدة
Set oPrp = oTbl.CreateProperty("test", dbInteger, 0)
' يضم (يضيف) الخاصية الجديدة
oTbl.Properties.Append oPrp
' ينعش قائمة الخصائص
oTbl.Properties.Refresh
End If
' محرر الكائنات
oDb.Close
Set oTbl=Nothing
Set oDb=Nothing
MsgBox "انضمت الخاصية المنشأة"
End Sub
```

الطريقة CreateProperty ترجع كائنا من نوع Property و الكود أعلاه يمكن اختزاله بما يلي :

```
Sub ajout()
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = Db.TableDefs("Commande")
' يختبر عدم وجود الخاصية
If Not ExistProperty(oTbl.Properties, "Test") Then
' ينشئ ويضم الخاصية الجديدة في نفس الوقت
oTbl.Properties.Append oTbl.CreateProperty("test", dbInteger, 0)
' ينعش قائمة الخصائص
oTbl.Properties.Refresh
End If
' محرر الكائنات
oDb.Close
Set oTbl=Nothing
Set oDb=Nothing
MsgBox "انضمت الخاصية المنشأة"
End Sub
```

وبالمثل ، كما يمكن التحقق من وجود الخاصية ، يمكن محاولة إضافتها مع معالجة الأخطاء طبعاً :

```
Sub ajout()
On Error GoTo err
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Commande")
' ينشئ ويضم الخاصية الجديدة في نفس الوقت
oTbl.Properties.Append oTbl.CreateProperty("test", dbText, 0)
```

```
ينعش قائمة الخصائص '  
oTbl.Properties.Refresh  
MsgBox "انضمت الخاصية المنشأة"  
Exit Sub  
يعالج الأخطاء '  
err:  
Select Case err.Number  
    Case 3367: MsgBox "الخاصية موجودة مسبقا"  
    Case Else: MsgBox "وقع خطأ غير معروف"  
End Select  
End Sub
```

4.3 - الكائن TableDef

4.3.1 - تعريفات

المجموعة TableDefs للكائن Database ، تضم كل الجداول بقاعدة البيانات ، و هذه المجموعة تضم كذلك جداول النظام والجداول المرتبطة . وكل كائن TableDef من المجموعة ، يمتلك مجموعة من كائنات الحقول Field وكائن فهرس Index ، وهذان النوعان من الكائنات سيتم دراستهما لاحقا .

تماما ، كما لو أنك في بنية التطوير بالأكسيس ، وفي عرض التصميم ، يستحيل أن تعدل في بنية جدول مرتبط ، فإن ذلك مستحيل أيضا عن طريق DAO .



4.3.2 - قائمة الخصائص

فيما يلي نستعرض قائمة الخصائص المطبقة على الكائن TableDef

بعض من هذه الخصائص ، ليست متاحة للكتابة ، فقط عدا للكائنات التي لم تضم بعد للمجموعة Database.TableDefs



4.3.2.1 - الخاصية Attributes

Type : Long

الخاصية Attributes ، تميز الجدول المعني ، ويتعلق الأمر بمجموعة من الثوابت كمايلي :

الثابتة	دلالتها
dbAttachExclusive	تستعمل فقط على الجداول المرتبطة . وهذه القيمة تشير إلى أن الجدول المرتبط مفتوح في وضع خاص <i>exclusif</i>
dbAttachSavePWD	تستعمل فقط مع الجداول المرتبطة . ومع هذه الثابتة تحفظ معلومات التسجيل المستعملة لربط الجداول
dbSystemObject	تشير إلى جدول نظام <i>systeme</i>
dbHiddenObject	تشير إلى جدول مخفي <i>cachée</i>
dbAttachedTable	تشير إلى جدول مرتبط عبر المحرك Jet
dbAttachedODBC	تشير إلى جدول مرتبط عبر المحرك ODBC

4.3.2.2 - الخاصية Connect

Type : String

يتعلق الأمر بعبارة الاتصال (سلسلة نصية) المستعملة لربط جدول ، قيمة هذه العبارة = Null ، عندما لا يكون الجدول مرتبطا . والعبارة هذه تأخذ في الحسبان عدة معلمات ؛ مثل اسم المستخدم ، كلمة مروره ، ... إلخ إنه بإمكانك استعراض خصائص الجدول في عرض التصميم

هذه الخاصية هي للقراءة فقط بالنسبة لكل كائنات TableDef الموجودة بالمجموعة Database.TableDefs لقاعدة البيانات



4.3.2.3 - الخاصية *DataCreated*

للقراءة فقط

Type : Date

الخاصية *DataCreated* ترجع تاريخ إنشاء الجدول

4.3.2.4 - الخاصية *Fields*

Type : Collection

هذه المجموعة تضم كل الحقول بالجدول ، ويمكن التعديل عليها بالإضافة أو الحذف

4.3.2.5 - الخاصية *Indexes*

Type : Collection

مجموعة تضم كل الفهارس الموجودة بالجدول ، ويمكن التعديل عليها بالإضافة أو الحذف

4.3.2.6 - الخاصية *LastUpdated*

للقراءة فقط

Type : Date

الخاصية *LastUpdated* ترجع تاريخ وساعة آخر تعديل حصل على هيكل الجدول المعني

4.3.2.7 - الخاصية *Name*

Type : String

الخاصية *Name* تتلقى اسم الجدول ، وهذا الاسم يجب أن يكون فريدا ضمن المجموعة *TableDefs* لقاعدة البيانات .

ستجد فيما يلي في الفقرة (4.3.4.2) مثلا يوضح كيفية تعديل اسم لجدول في قاعدة بيانات .



4.3.2.8 - الخاصية *Properties*

Type : Collection

الخاصية *Properties* هي مجموعة تضم كل الخصائص للكائن *TableDef*

4.3.2.9 - الخاصية RecordCount

للقراءة فقط

Type : Long

الخاصية RecordCount ترجع عدد السجلات بالجدول

4.3.2.10 - الخاصية SourceTableName

Type : String

يتعلق الأمر بالاسم الأصلي للجدول المرتبط . وإذا كانت الجدول غير مرتبط ، فإن هذه الخاصية ستكون للقراءة فقط وقيمتها عبارة فارغة ("").

4.3.2.11 - الخاصية Updatable

للقراءة فقط

Type : Boolean

الخاصية Updatable تحدد ما إذا كانت البيانات المرسله من الجدول قابلة التعديل عليها أم لا

4.3.2.12 - الخاصية ValidationRule

Type : String

الخاصية ValidationRule ، تحدد قاعدة التحقق من الصحة الذي سوف يراعي عدد محاولة التحديث (بما في ذلك الإدراج) .
وكذلك لإجبار المستخدم على ملء الحقل (MyChamp) مثلاً عندما يكون صدق الاختيار MyCase مؤشر بعلامة فلدينا القاعدة التالية :

([MyChamp] IS NOT NULL AND [MyCase]) OR NOT ([MyCase])

4.3.2.13 - الخاصية ValidationText

Type : String

الخاصية ValidationText تتعلق بالرسالة التي ستعلن عند التحديث إذا كانت البيانات لا تحقق قاعدة التصديق المحدد في ValidationRule

4.3.3 - قائمة الطرائق

4.3.3.1 - الخاصية *CreateField*

الطريقة *CreateField* تنشئ وترجع كائن *Field* جديد ، سينضم تلقائياً إلى المجموعة *Database.TableDefs*

تركيب :

```
Set field = TableDef.CreateField (name, type, size)
```

هذه الطريقة سيفصل الحديث عنها في الفقرات المخصصة للكائن *Field*



4.3.3.2 - الخاصية *CreateIndex*

الطريقة *CreateIndex* تنشئ فهرساً للجدول . هذا الفهرس يجب أن ينضم إلى المجموعة *Indexes* للكائن *TableDef* حتى يمكن تفعيله .

تركيب :

```
Set index = tabledef.CreateIndex (name)
```

4.3.3.3 - الخاصية *CreateProperty*

الطريقة *CreateProperty* تنشئ وترجع كائناً جديداً *Property* ، الذي يجب أن ينضم إلى المجموعة *Properties* للكائن *TableDef*

التركيب المقترح لهذه الطريقة سبق عرضه في الفقرة (4.2 : الخصائص النوعية)



4.3.3.4 - الخاصية *OpenRecordset*

الطريقة *OpenRecordset* تفتح كائن سجلات *recordset* جديد ، يتضمن جميع السجلات المخزنة بالجدول المصدر .

تركيب :

```
Set recordset = object.OpenRecordset (type, options, lockedits)
```

4.3.3.5 - الخاصية RefreshLink

الطريقة RefreshLink لا تستعمل إلا مع الجداول المرتبطة . وهي تسمح بعد التعديل على الخاصية Connect بإنعاش بيانات الاتصال بين الجدول المرتبط ومصدر بياناته .

تركيب :

```
tabledef.RefreshLink
```

4.3.4 - إجراءات على جداول قاعدة بيانات

4.3.4.1 - استعراض جداول قاعدة بيانات

سأقترح عليك فيما يلي بعض الأكواد التي تتيح لك استعراض كل الجداول بقاعدة بيانات . وسنميز بداية بين جداول النظام والجداول المرتبطة . ففي الواقع ، الخاصية Attributes هي قناع ثنائي binaire . والعامل AND الموجود بين ثابتة (X) مبحوث عنها وقيمة الخاصية ، ينجز إجراء (ثنائيا binaire) يرسل عددا يساوي 0 ، إذا كانت الثابتة غير مستعملة في الخاصية .
ولذلك فإنه لمعرفة هل الجدول جدول نظام يمكن استعمال التركيب :

```
If monObjet.Attributes AND dbSystemObjet then .....
```

لاستعراض كل جداول النظام يكفي إذن تمرير جداول المجموعة TableDefs واحدا واحدا واختبار التعبير البولي boolean فهي متحققة أم لا .

```
Sub Lister()  
Dim oDb As DAO.Database  
Dim oTbl As DAO.TableDef  
Set oDb = CurrentDb  
' الحلقة التكرارية لكل جدول  
For Each oTbl In oDb.TableDefs  
    ' إن كان الجدول نظام إذن أظهر الاسم  
    If oTbl.Attributes And dbSystemObject Then MsgBox oTbl.Name  
Next oTbl  
End Sub
```

و لاستعراض كل الجداول المرتبطة لدينا الكود :

```
Sub Lister()  
Dim oDb As DAO.Database  
Dim oTbl As DAO.TableDef  
Set oDb = CurrentDb  
' الحلقة التكرارية لكل جدول  
For Each oTbl In oDb.TableDefs  
    ' إن كان الجدول مرتبطا إذن أظهر الاسم  
    If oTbl.Attributes And dbAttachedTable Then MsgBox oTbl.Name  
Next oTbl  
End Sub
```

ولاستعراض الجداول معا ، يكفي أن نستعمل العامل المنطقي (Or) والكود التالي يحقق ذلك :

```
Sub Lister()
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
' الحلقة التكرارية لكل جدول
For Each oTbl In oDb.TableDefs
' إن كان الجدول مرتبطا أو نظام إذن أظهر الإسم
If oTbl.Attributes And (dbAttachedTable Or dbSystemObject) Then
MsgBox oTbl.Name
Next oTbl
End Sub
```

هذا الكود لا يتعارض والتالي :

```
Sub Lister()
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
' الحلقة التكرارية لكل جدول
For Each oTbl In oDb.TableDefs
' إن كان الجدول مرتبطا أو نظام إذن أظهر الإسم
If oTbl.Attributes And (dbAttachedTable And dbSystemObject) Then
MsgBox oTbl.Name
Next oTbl
End Sub
```

والواقع أن هذا الكود يستعرض في الوقت نفسه جداول النظام والجداول المرتبطة .

4.3.4.2 - إعادة تسمية جدول

إعادة تسمية (تغيير اسم) جدول هو إجراء سهل للغاية . غير أنه يجب ألا ننسى بأن اسم الجدول يجب أن يكون اسما فريدا في المجموعة TableDefs ، وبالتالي يجب وضع كود لمعالجة الأخطاء المحتملة عن مثل هذا الإجراء .

الكائنات Relations المنضمة للجدول سوف يتم تحديثها أليا بواسطة المحرك Jet



ولهذا سنستخدم الدالة RenommerTable التالية :

```
Private Function RenommerTable(oBaseDeDonnees As DAO.Database,
strAncienNom As String, _
strNouveauNom As String) As Boolean
On Error GoTo err
Dim oTbl As DAO.TableDef
' يبحث عن الجدول لتعديل اسمه
Set oTbl = oBaseDeDonnees.TableDefs(strAncienNom)
' إعادة التسمية
oTbl.Name = strNouveauNom
' إنعاش المجموعة
oBaseDeDonnees.TableDefs.Refresh
```

```

    ' يرجع true إن نجح الإجراء '
    RenommerTable = True
Exit Function
err:
Select Case err.Number
    Case 3265: MsgBox " لم يتمكن من إيجاد الجدول لإعادة التسمية "
    Case 3010: MsgBox " الجدول : " & strNouveauNom & " موجود مسبقا "
    Case Else: MsgBox " حدث خطأ غير معروف "
End Select
End Function
    
```

طريقة استدعاء الدالة :

```

Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
If RenommerTable(oDb, "Commande", "Commande_Archive") Then
    MsgBox " أعيديت تسمية الجدول بنجاح "
End if
    
```

4.3.4.3 - حذف جدول

إن حذف جدول يستدعي بعض الاحتياطات ، فيجب أولاً أن يكون الجدول موجوداً حتى يمكن حذفه ، كما يجب ألا يغيب عن أذهاننا أن حذف جدول يكسر كل العلاقات المرتبطة به ، وأيضاً فإن التسرع باتخاذ قرار الحذف قد يعرض كائنات Relations للحذف إجمالاً ، لذا أنصحك بإرسال رسالة تنبيه للمستخدم تطلب منه تأكيد قرار الحذف قبل إنجاز الإجراء .
سنحتاج إذن لمعرفة ما إذا كان الجدول المراد حذفه مرتبطاً بكائنات علاقة ، ولهذا سنستخدم الدالة **ExistRelation** التالية :

```

Private Function ExistRelation(oBaseDeDonnees As DAO.Database, _
    strNomTable As String) As Integer
Dim oRlt As DAO.Relation
' الحلقة التكرارية لكل جدول
For Each oRlt In oBaseDeDonnees.Relations
' يختبر إن كان الجدول يدخل في علاقة
    If oRlt.Table = strNomTable Or oRlt.ForeignTable = strNomTable Then
        ExistRelation = ExistRelation + 1
    End If
Next oRlt
End Function
    
```

هذه الدالة ترجع العلاقات المنضمة للجدول المرسل للمعلمة strNomTable
نستطيع إذن كتابة دالة حذف الجدول التالية :

```

Private Function SupprimerTable(oBaseDeDonnee As DAO.Database, _
    strNomTable As String)
On Error GoTo err
Dim intNbRlt As Integer
' بعد العلاقات المرتبطة
intNbRlt = ExistRelation(oBaseDeDonnee, strNomTable)
' إذا كان الجدول على الأقل يرتبط بعلاقة
If NbRlt <> 0 Then
    
```

```

        ' إذا لم يرغب المستعمل في المتابعة / خروج '
        If MsgBox( _
            & " علاقة . " & intNbRlt & " مرتبط بـ: " & strNomTable & " الجدول : " & _
            & vbCrLf & " إن حذف هذا الجدول سيتسبب في كسر هذه العلاقات " & _
            & vbCrLf & " ("حذف جدول " & vbCrLf & " هل ترغب بالمتابعة ؟ " , vbQuestion + vbYesNo, _
            = vbNo Then
            Exit Function
        End If
        ' يحذف الجدول
        oBaseDeDonnee.TableDefs.Delete (strNomTable)
        ' ينعش المجموعة
        oBaseDeDonnee.TableDefs.Refresh
        ' يرسل true في حال تمام الإجراء
        SupprimerTable = True
        Exit Function
err:
        ' يعالج الإخطاء المحتملة
        Select Case err.Number
            Case 3265: MsgBox " لم يتم العثور على الجدول "
            Case Else: MsgBox " حدث خطأ غير معروف "
        End Select
    End Function

```

طريقة استدعاء الدالة :

```

Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Set oDb = CurrentDb
If SupprimerTable(oDb, "Commande") Then
    MsgBox " تم حذف الجدول بنجاح "
End If

```

4.3.4.4 - إنشاء جدول.

إنشاء جدول ليس ذا جدوى إن لم تنشأ به الحقول ، وأقترح عليك قبل الاستمرار في متابعة هذه الفقرة ، الانتقال للاطلاع على الفقرات التي تتعرض للحقول والفهارس ، ثم العودة بعد ذلك إلى المثال التالي :

في هذا المثال ، سننشئ جدولاً نسميه *Client* به ثلاث حقول :

- IDClient : مفتاح أساس , Type : Long. ترقيم تلقائي
- NomClient : حقل نص 15 حرفاً
- PrenomClient : حقل نص 25 حرفاً

```

Sub CreerTable()
Dim oDb As DAO.Database
Dim oNouvelleTable As DAO.TableDef
Dim oChamp As DAO.Field
Dim oIndex As DAO.Index
    ' يهيء قاعدة البيانات
    Set oDb = CurrentDb
    ' ينشئ الجدول الجديد
    Set oNouvelleTable = oDb.CreateTableDef("Clients")
    ' ينشئ الحقل : IDClient

```

```

Set oChamp = oNouvelleTable.CreateField("IDClient", dbLong)
' تخصيص الحقل : (ترقيم تلقائي)
oChamp.Attributes = dbAutoIncrField
' يضم الحقل إلى الجدول
oNouvelleTable.Fields.Append oChamp
' ينشئ الحقل : nomClient ثم يضيفه إلى الجدول
oNouvelleTable.Fields.Append
oNouvelleTable.CreateField("NomClient", dbText, 15)
' ينشئ الحقل : PrenomClient ثم يضيفه إلى الجدول
oNouvelleTable.Fields.Append
oNouvelleTable.CreateField("PrenomClient", dbText, 25)
' يخصص الحقل IDClient مفتاح أساس
Set oIndex = oNouvelleTable.CreateIndex("PK_IDClient")
oIndex.Primary = True
oIndex.Fields.Append oIndex.CreateField("IdClient")
' يضم الفهرس إلى الجدول
oNouvelleTable.Indexes.Append oIndex
' يضم الجدول إلى قاعدة البيانات
oDb.TableDefs.Append oNouvelleTable

' يمرر الكائنات
oDb.Close
Set oIndex = Nothing
Set oChamp = Nothing
Set oNouvelleTable = Nothing
Set oDb = Nothing
End Sub
    
```


4.4 - الكائن Field

4.4.1 - تعريف

الكائن Field يمثل حقلا بجدول أ و استعمال في قاعدة البيانات . و النموذج DAO ، يقدم إمكانيات عدة متعلقة بالحقول ، فبالإمكان إنشاء حقل ، أو حذفه أو تعديله بسلسلة ربما أكبر مما يحدث في وضع التصميم للأكسيس ، فهناك تشابه إلى أبعد حد بين قائمة الخصائص للكائن Field و نافذة التعديل على بنية (هيكل) الجدول . فكل حقل نص في هذه النافذة يوافق خاصية من قائمة الخواص للكائن Field و أقترح عليك في البداية أن تقوم بجولة استطلاع على كافة الخواص والطرائق المتعلقة بهذا الكائن قبل الخوض في غمار الأمثلة الأكثر تخصصا .

4.4.2 - قائمة الخصائص :

سأستعرض فيما يلي كافة الخصائص للكائن Field ضمن القائمة التالية :

بعض من هذه الخصائص ، ليست متاحة للكتابة ، عدا للكائنات التي لم تضم بعد للمجموعة TableDef.Fields	
---	---

4.4.2.1 - الخاصية AllowZeroLength

Type : Boolean
تحدد هذه الخاصية إذا كان الحقل يقبل عبارة (سلسلة) نصية فارغة ، فإن كان يقبل بها ساوت قيمة هاته الخاصية True ، وإلا فإن رسالة خطأ ستعلن عند محاولة التحديث للبيانات .

4.4.2.2 - الخاصية Attributes

Type : Long
الخاصية Attributes تحدد خيارات الحقل ، والقيم الممكن إرسالها لهذه الخاصية هي الثوابت التالية :

الثابتة	دلالتها
dbAutoIncrField	الحقل عدد من نوع ترقيم تلقائي ، ولا تستعمل هذه القيمة إلا مع الحقول الرقمية
dbDescending	في حالة ما إذا كان الحقل فهرسا فإن هذه القيمة لخاصية Attributes ، تقرض الترتيب العكسي للفرز
dbFixedField	قيمة افتراضية للحقول الرقمية ، فهذه الحقول ذات سعة ثابتة
dbHyperlinkField	تدبير الارتباط التشعبي في حقل من نوع مذكرة
dbSystemField	هذه القيمة للخاصية Attributes تجعل الحقل يخزن معلومات عن الاستنساخ حين وجود نسخة ثانية duplicatas ، و مثل هذا الحقل لا يمكن حذفه ..
dbUpdatableField	قيمة الحقل يمكن تغييرها
dbVariableField	هذا الحقل قابل للتعديل ، ولا تستعمل هذه القيمة إلا مع الحقول من نوع نص

4.4.2.3 - الخاصية *CollatingOrder*

Type : Long

إنها خاصية تتعلق بالطريقة التي يستعملها المحرك للمقارنة بين البيانات من نوع (نص) ويمكن أن تأخذ القيم التالية :

الثابتة	دلالاتها
dbSortGeneral	عام (انجليزية ، فرنسية ... لغات غرب أوربية)
dbSortArabic	اللغة العربية
dbSortChineseSimplified	اللغة الصينية
dbSortChineseTraditional	اللغة الصينية التقليدية
dbSortCyrillic	اللغة الروسية
dbSortCzech	اللغة التشيكية
dbSortDutch	اللغة الهولندية
dbSortGreek	اللغة اليونانية
dbSortHebrew	اللغة العبرية
dbSortHungarian	اللغة الهنغارية
dbSortIcelandic	اللغة الإيسلاندية
dbSortJapanese	اللغة اليابانية
dbSortKorean	اللغة الكورية
dbSortNeutral	حيادية
dbSortNorwDan	اللغة النرويجية ، الدانماركية
dbSortPolish	اللغة البولونية
dbSortSlovenian	اللغة السلوفانية
dbSortSpanish	اللغة الإسبانية
dbSortSwedFin	اللغة السويدية ، الفنلندية
dbSortThai	اللغة التايلاندية
dbSortTurkish	اللغة التركية
dbSortUndefined	غير محددة

4.4.2.4 - الخاصية *DataUpdatable*

للقراءة فقط


Type : Boolean

إذا كانت قيمة هذه الخاصية تساوي False فإنه يصبح من المستحيل التعديل على قيم الحقل ، إنها بالضبط القيمة التي تأخذها هذه الخاصية بالنسبة للحقل من نوع ترقيم تلقائي

4.4.2.4 - الخاصية *DefaultValue*

Type : Variant

هذه الخاصية تحدد القيمة الافتراضية للحقل ، وهي خاصة للكتابة والقراءة بالنسبة لحقول الجدول الأول *TableDef* ، وللقراءة فقط بالنسبة لحقول الاستعلامات *QueryDef* وكائن السجلات *Recordset* . ولا تؤخذ بالاعتبار بالنسبة لكائنات *Fields* المنضمة للمجموعات *Database.Indexes* و *Database.Relations* .

بالنسبة لحقل من نوع عدد صحيح طويل *Long* ، يمكنك أن ترسل لهذه الخاصية القيمة : *GenUniqueID()* ، وهكذا سيعمل المحرك *Jet* على إرسال قيمة وحيدة عشوائية لكل سجل جديد ، وهذه القيمة تنتج محرك ترقيم تلقائي عشوائي . 

4.4.2.5 - الخاصية *FieldSize*

للقراءة فقط

Type : Long

الخاصية *FieldSize* ، تقدر حقلًا من نوع ثنائي أو مذكرة . وهذه القيمة يعبر عنها بعدد الأحرف لحقل مذكرة ، وعدد الأثمنونات *octets* بالنسبة لحقل ثنائي

4.4.2.6 - الخاصية *ForeignName*

Type : String

الخاصية *ForeignName* : لا تستعمل إلا مع حقول لكائن من نوع علاقة *Relation* ، إنها تحدد اسم الحقل في الجدول الخارجي للعلاقة (مفتاح أجنبي) . وهذه الخاصية للقراءة والكتابة في حالة العلاقات غير المنضمة إلى المجموعة *Database.Relations* ، وللقراءة فقط عندما تنضم إليها .

4.4.2.7 - الخاصية *Name*

Type : String

اسم الكائن (حقل) ، ويجب أن يكون اسمًا فريدًا في الجدول

4.4.2.8 - الخاصية *OrdinalPosition*

Type : Integer

الخاصية *OrdinalPosition* ، تمكن من تحديد موقع الحقل داخل المجموعة *TableDef.Fields* ، وطبعا الحقل الأول يأخذ الموقع 0، والحقل الأخير يحل في الموقع (n-1) . هذا الموقع مشروط بالترتيب الذي تظهر عليه الحقول في استعلام *SELECT * FROM ...* . ولنسجل أنه إذا ما ساوت خاصية *OrdinalPosition* في حقلين نفس القيمة ، فهذا يعني أن الحقول مرتبة أبجديًا . هذه الخاصية للقراءة فقط بالنسبة للكائن استعلام *QueryDef* .

4.4.2.9 - الخاصية *Properties*

Type : Collection
هذه المجموعة تضم كافة كائنات (خاصية Property) للكائن Field

4.4.2.10 - الخاصية *Required*

Type : Boolean
الخاصية Required ، تحدد إن كان الحقل يقبل القيم الخالية null ، أن ساوت قيمة هذه الخاصية True ، فإن خطأ سيعلن حين محاولة إرسال القيمة : null إلى هذا الحقل .

4.4.2.11 - الخاصية *Size*

Type : Integer
الخاصية Size ، تحدد حجم الحقل تماما كما تفعل في وضع التصميم للجدول ، وهذه الخاصية للكتابة والقراءة ، فقط للحقول التي لم تنضم بعد للمجموعة Fields

4.4.2.12 - الخاصية *SourceField*

للقراءة فقط
Type : String
هذه الخاصية للقراءة فقط ، وتشير في الاستعلام إلى الاسم الأصلي وليس إلى الاسم المقروض alias للكائن (حقل)

ليكن لدينا استعلام "SELECT NomClient as LeNom FROM Client" ، ولنسمه R01 :

الكود التالي سيعرض اسم الحقل :

```
Dim oDb As DAO.Database
Dim oQR As DAO.QueryDef
Set oDb = CurrentDb
Set oQR = oDb.QueryDefs("R01")
MsgBox oQR.Fields(0).Name
```

الكود التالي سيعرض الاسم : *NomClient* :

```
Dim oDb As DAO.Database
Dim oQR As DAO.QueryDef
Set oDb = CurrentDb
Set oQR = oDb.QueryDefs("R01")
MsgBox oQR.Fields(0).SourceField
```

يمكننا كتابة الدالة التالية التي ترجع اسم الحقل استنادا إلى اسمه المقروض :

```
Function NomChamp(oBaseDeDonnee As DAO.Database, strRequete As String & _
String, strAlias As String) As String
On Error GoTo err
Dim Qdf As DAO.QueryDef
Set Qdf = oBaseDeDonnee.QueryDefs(strRequete)
NomChamp = Qdf.Fields(strAlias).SourceField
Exit Function
' يعالج الأخطاء المحتملة
err:
Select Case err.Number
Case 3265
' إذا كان qdf فارغة إذن خروج
If Qdf Is Nothing Then
MsgBox " لم يتم العثور على الاستعلام " & strRequete
Else
MsgBox " بالاستعلام " & strAlias & " لم يتم العثور على الحقل " & strRequete
End If
Case Else
MsgBox " حدث خطأ غير معروف "
End Select
End Function
```

الكود التالي لاستدعاء الدالة أعلاه :

```
Dim oDb As DAO.Database
Set oDb = CurrentDb
MsgBox NomChamp(oDb, "R01", "LeClient")
```

4.4.2.13 - الخاصية SourceTable

للقراءة فقط

Type : String

هذه الخاصية تركز إلى نفس المبدأ الذي ترتبته إليه الخاصية السابقة ، فهي تتيح معرفة الجدول الذي ينحدر منه حقل في استعلام وهكذا انطلاقاً من المثال السابق يمكن معرفة الجدول المصدر للحقل *NomClient* :

```
Dim oDb As DAO.Database
Dim oQR As DAO.QueryDef
Set oDb = CurrentDb
Set oQR = oDb.QueryDefs("R01")
MsgBox oQR.Fields(0).SourceTable
```

4.4.2.14 - الخاصية Type

Type : Integer

هذه الخاصية تحدد نوع الحقل ، وهي تأخذ قيمة ثابتة مما يلي :

النوع	قيمتها	الثابتة
رقمية (دقة عالية)	16	dbBigInt
ثنائي بطول ثابت (255 حد أقصى)	9	dbBinary
بولي Booléen	1	dbBoolean

النوع	قيمتها	الثابتة
بايت	2	dbByte
سلسلة نصية بطول ثابت	18	dbChar
عملة	5	dbCurrency
تاريخ	8	dbDate
عشري	20	dbDecimal
مزدوج	7	dbDouble
بفاصلة عائمة	21	dbFloat
معرف النسخة المماثلة	15	dbGUID
عدد صحيح	3	dbInteger
عدد صحيح طويل	4	dbLong
ثنائي بطول متغير (OLE) .	11	dbLongBinary
مذكرة	12	dbMemo
رقمي	19	dbNumeric
مفرد	6	dbSingle
عبارة نصية بطول ثابت	10	dbText
وقت	22	dbTime
TimeStamp	23	dbTimeStamp
ثنائي بطول متغير (255 أتمونة كحد أقصى)	17	dbVarBinary

4.4.2.15 - الخاصية ValidationRule

Type : String
الخاصية ValidationRule ، تحدد قاعدة التحقق من الصحة التي سوف تراعي عند محاولة التحديث (ويؤخذ الإلحاق بعين الاعتبار)

4.4.2.16 - الخاصية ValidationText

Type : String
الخاصية ValidationText تتعلق بالرسالة التي ستعلن عند التحديث إذا كانت البيانات لا تحقق قاعدة التصديق المحددة في ValidationRule

4.4.2.17 - الخاصية Value

Type : Variant
بالنسبة لكائن السجلات Recordset ، الخاصية Value ترجع قيمة الحقل في السجل الحالي .

4.4.2.18 - الخاصية VisibleValue

Type : Variant
هذه الخاصية تستعمل فقط في مجال ODBCdirect ، وترسل القيمة الحالية لحقل عند تتم المعالجة

بالكتل وهذه القيمة أهدت من تلك التي ترجعها الخاصية OriginalValue

4.4.3 - قائمة الطرائق

4.4.3.1 الطريقة AppendChunk

الطريقة AppendChunk تمكن من إضافة البيانات في نهاية الحقل من نوع مذكرة أو نوع الثنائي binaire (بكاتنات السجلات Recordsets حصريا). إنها تمنع طبعا القواء من بداية الحقل، أو دمج أو إرسال القيمة الجديدة. واسدعمال الخاصية AppendChunk تحد إذن من كمية تدفق البيانات بين القاعدة والمحرك Jet

التركيب :

```
field.AppendChunk source.
```

المعلمة : source من نوع Variant حتى تحتل كل أنواع البيانات

4.4.3.2 - الطريقة CreateProperty

الطريقة CreateProperty تنشئ وترجع كائنا جديدا من نوع خاصية Property التي ستتضم إلى مجموعة الخصائص Properties للكائن Field

تركيب :

```
Set property = object.CreateProperty (name, type, value, DDL)
```

4.4.3.3 - الطريقة GetChunk

الطريقة GetChunk ، تطبق على حقول كائنات الاسجلات Recordset ، وهي ترجع مضمون (أو جزءا من مضمون) حقل مذكرة ، أو حقل ثنائي binaire ، ويرسل لمتغيرة من نوع نص .

التركيب :

```
Set variable = field.GetChunk (offset, numbytes)
```


المعلمة offset هي من نوع عدد طويل Long ، وتتعلق بموقع الحرف (أو البايت) حيث يجب أن تبدأ القراءة .
المعلمة numbytes تحدد عدد الحروف التي ستقرأ .

4.4.4 - إجراءات على حقول كائن جدول TableDef

فقط الكائن TableDef يسمح بتغيير (تعديل) عناصره بالمجموعة Fields، وأقترح عليك مجموعة من الأمثلة التطبيقية لبيان الإمكانيات الهائلة التي يتيحها النموذج DAO

4.4.4.1 - تغيير اسم حقل

كما مر معنا في الفقرة (- 4.3.4.2 إعادة تسمية جدول) ، فهذه العملية تستدعي أن يكون الاسم الجديد غير مسبق بنظيره في المجموعة Fields للكائن TableDef

الكائنات (علاقة Relation و فهرس Index) باستعمالها لهذا الحقل الذي تم تغيير اسمه يتم تحديثها أليا	
--	---

دالة **RenommerChamp** لتغيير اسم الكائن حقل :

```
Private Function RenommerChamp (oBaseDeDonnees As DAO.Database, _
strNomTable As String, strAncienNomChamp As String, strNouveauNomChamp _
As String) As Boolean

On Error GoTo err
Dim Tbl As DAO.TableDef
' استرداد الجدول
Set Tbl = oBaseDeDonnees.TableDefs(strNomTable)
' يغير اسم الحقل
Tbl.Fields(strAncienNomChamp).Name = strNouveauNomChamp
' ارجاع النتيجة True
RenommerChamp = True

Exit Function
' معالجة الأخطاء
err:
Select Case err.Number
' إذا لم يكن العثور على الكائن ضمن المجموعة
Case 3265
' إذا لم يكن العثور على الجدول tbl=nothing إذن الجدول مصدر الخطأ
If Tbl Is Nothing Then
MsgBox " : " & strNomTable
Else
MsgBox " : " & strAncienNomChamp
End If
Case 3010, 3191: MsgBox " الحقل " & strNouveauNomChamp & " وجود " _
& " مسبقا "
Case Else: MsgBox " خطأ غير معروف "
End Select
End Function
```

طريقة استدعاء الدالة :

```
Dim oDb As DAO.Database
Set oDb = CurrentDb
If RenommerChamp (oDb, "Commande", "Reference", & _
"Nouvelle_Reference") then
    MsgBox "تمت العملية بنجاح"
End if
```

4.4.4.2 - حذف حقل

عملية حذف حقل عملية جد حساسة ، ففي الواقع كثير من الأخطاء يمكن أن تقع من جراء هذه العملية

- 3265 : الحقل غير موجود
- 3280 : الحقل مستعمل من قبل فهرس
- 3303 : الحقل داخل في علاقة أو عدة علاقات

الخطأ 3303 من الصعب التحكم فيه ، لأنه وإن تم إعلان الخطأ فإن الكائن سيحذف بينما تظل العلاقات قائمة ، وإذا أعدت إضافة الحقل من جديد فإن العلاقات ستنشأ من جديد مع إضافته ، مما يثير مواقف غير منتظرة إنه يلزمنا بداية التأكد من عدم دخول الحقل في علاقات .

```
Private Function ChampDansRelation(oBaseDeDonnees As DAO.Database, _
strNomTable As String, strNomChamp As String) As Integer
On Error GoTo err
Dim bolUtilise As Boolean
Dim oRlt As DAO.Relation
Dim oFld As DAO.Field
    'تمرير العلاقات
For Each oRlt In oBaseDeDonnees.Relations
    bolUtilise = False
    If oRlt.Table = strNomTable Then
        'التأكد من أسماء كل الحقول
        For Each oFld In oRlt.Fields
            If oFld.Name = strNomChamp Then bolUtilise = True
        Next oFld
    ElseIf oRlt.ForeignTable = strNomTable Then
        'بالنسبة لكل حقل، يراقب الاسم الخارجي
        For Each oFld In oRlt.Fields
            If oFld.ForeignName = strNomChamp Then bolUtilise = True
        Next oFld
    End If
    'إذا كانت العلاقة تستخدم الحقل ، ترفع النتيجة +1
    If bolUtilise Then ChampDansRelation = ChampDansRelation + 1
Next oRlt
Exit Function
err:
ChampDansRelation = -1
End Function
```

هذه الدالة ترجع عدد العلاقات التي تستعمل الحقل المرسل إليها بالمعلمة .

طريقة استدعاء الدالة :

```
Dim oDb as DAO.Database
Set oDb=CurrentDB
MsgBox ChampDansRelation(oDb, "Commande", "Reference")
```

هذا المثال يظهر عدد العلاقات المستعملة لحقل Reference بالجدول Commande.

الدالة ترجع القيمة -1 عند حدوث خطأ



يمكن إذن الآن كتابة دالة حذف الحقل التالية :

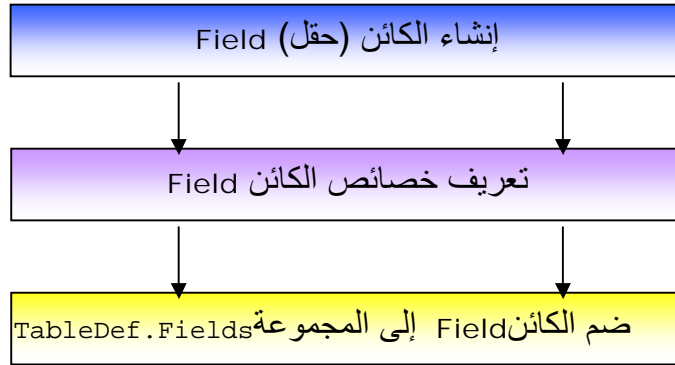
```
Private Function SupprimerChamp(oBaseDeDonnees As DAO.Database, _
strNomTable As String, strNomChampASupprimer As String) As Boolean
On Error GoTo err
Dim oTbl As DAO.TableDef
'التحقق من كون الحقل لا يدخل في علاقات
If ChampDansRelation(oBaseDeDonnees, strNomTable, _
strNomChampASupprimer) = 0 Then
'استرجاع الجدول
Set oTbl = oBaseDeDonnees.TableDefs(strNomTable)
'حذف الحقل
oTbl.Fields.Delete (strNomChampASupprimer)
'ارجاع النتيجة True
SupprimerChamp = True
Else
MsgBox "يستحيل حذف الحقل لاستعماله من طرف علاقة أو عدة علاقات"
End If
Exit Function
Select Case err.Number
'إذا لم يكن العثور على الكائن ضمن المجموعة
Case 3265
'إذا لم يكن العثور على الجدول tbl=nothing إذن الجدول مصدر الخطأ
If Tbl Is Nothing Then
MsgBox "لم يكن العثور على الجدول " & strNomTable
Else
MsgBox "لم يكن العثور على الحقل " & strNomChampASupprimer
End If
Case 3280: MsgBox "يستحيل حذف الحقل لاستعماله من طرف فهرس "
Case Else: MsgBox "خطأ غير معروف "
End Select
End Function
```

طريقة استدعاء الدالة :

```
SupprimerChamp oDb, "Commande", "Reference"
```

4.4.4.3 - إنشاء حقل

عملية إنشاء حقل تمر من ثلاث مراحل :



ولعل المثال يكون أكثر بلاغة من فقرات شرح طويلة ، وأقترح عليك إنشاء حقل AdresseClient في الجدول Client، وهذا الحقل له السمات التالية :

- النوع : نص
- الطول : 120 حرفا
- عبارة فارغة : لا
- منع الحقل الفارغ : نعم

```

Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Dim oFld As DAO.Field
Set oDb = CurrentDb
Set oTbl = oDb.TableDefs("Client")
' الخطوة 1 إنشاء حقل
Set oFld = oTbl.CreateField("AdresseClient", dbText, 120)
' الخطوة 2 تحديد خصائص الحقل
oFld.AllowZeroLength = False ' لا : عبارة فارغة
oFld.Required = True ' نعم : منع الحقل الفارغ
' الخطوة 3 إضافة الحقل إلى الجدول
oTbl.Fields.Append Fld
' إنعاش المجموعة
oTbl.Fields.Refresh
  
```

إنه من المهم جدا ، احترام ترتيب الخطوات كما قدمت في المثال ، فإذا ضمنت الحقل إلى المجموعة Fields قبل أن تحدد خصائصه ، فإنك ستصدم بمجموعة من خصائص غير قابلة للكتابة .

لم أتعرض هنا لمعالجة الأخطاء ، والملاحظات عن الأخطاء في هذه العملية هي نفس الملاحظات في فقرة إعادة تسمية حقل ، وفي الواقع أكبر مجازفة في هذا الكود أن تستعمل للكائن الجديد اسما موجودا ضمن المجموعة .

4.4.4.4 - استنساخ حقل حقل

أقترح عليك فيما يلي كودا تبدو قوته حين ترغب بإنشاء حقل نظير لحقل آخر . إن المبدأ الذي يستند إليه هذا الكود بسيط ، ننطلق من حقلين ، الأول نجعله مصدرا للاستنساخ ، لذلك سنمرر كل خاصية لهذا الحقل عدا

خاصية الإسم Name طبعا ، ونكتب قيمها لخواص الحقل الثاني (المستنسخ).

بعض من تلك الخواص يحتمل أن تكون للقراءة فقط ، فسنحتاج إلى انتداب عملية القراءة والكتابة للخواص إلى إجراء آخر خارج إجراء الاستنساخ ، لمنع توقف البرنامج عند أول خطأ يصادفه .



```
Public Sub CloneChamp(oFldSource As DAO.Field, oTableDestination As _
DAO.TableDef, strNouveauNom As String)
On Error GoTo err
Dim oFld As DAO.Field
Dim oPrp As DAO.Property
' إنشاء الحقل
Set oFld = oTableDestination.CreateField(strNouveauNom)
' تمرير الخصائص
For Each oPrp In oFldSource.Properties
' ينقل الخاصية إن لم تكن الإسم
If oPrp.Name <> "Name" Then _
EcrirePropriete oPrp, oFld
Next oPrp
' يضيف الحقل
oTableDestination.Fields.Append oFld
Exit Sub
err:
Select Case err.Number
Case 3010, 3191: MsgBox " الحقل " & strNouveauNom & " موجود مسبقا "
Case Else: MsgBox " خطأ غير معروف المصدر "
End Select
End Sub
```

```
Private Sub EcrirePropriete(oPropriete As DAO.Property, _
oChampDestination As DAO.Field)
On Error GoTo err
' ينقل الخاصية
With oChampDestination.Properties
.Item(oPropriete.Name) = oPropriete.Value
.Refresh
End With
err:
' هنا معالجة الأخطاء
End Sub
```

لاستنساخ الحقل AdresseClient من الجدول Client بالجدول SauvegardeClient نستعمل

```
Dim oDb As DAO.Database
Dim oTblSource As DAO.TableDef, oTblDestination As DAO.TableDef
Dim oFld As DAO.Field
Set oDb = CurrentDb
' تخزين الجدولين
Set oTblSource = oDb.TableDefs("Client")
Set oTblDestination = oDb.TableDefs("SauvegardeClient")
' تخزين الحقل
Set oFld = oTblSource.Fields("AdresseClient")
' استنساخ الحقل
CloneChamp oFld, oTblDestination, "AdresseClient2"
```

4.5 - الكائن Index

4.5.1 - تعريفه

لن نخوض في تعريفات الفهرس ، غير أنه يجب التنكير بأن الفهرس يساعد على يسر الوصول للبيانات حين القيام ببحث في بيانات جدول ، كما يسمح أيضا بفرض نسق معين على البيانات (مفهوم المفتاح الأولي ، السماح بالتكرار أو بدون تكرار ...)
الفهرس يمكن أن يتكون من حقل واحد أو عدة حقول ، إنه يمتلك إذن مجموعة Fields تحصي هذه الحقول .

4.5.2 - قائمة الخصائص

4.5.2.1- الخاصية *DistinctCount*

للقراءة فقط

Type : Long

الخاصية *DistinctCount* ترجع عدد القيم الوحيدة المخزنة في جدول .
وفي حال الفهرس ذي الحقل الوحيد ، فإنها ترجع عدد القيم المفارقة بهذا الحقل (إذ يتعلق الأمر هنا بما يشبه الاستعلام `SELECT DISTINCT MonChamp...`)
وفي حال الفهرس المركب من عدة حقول ، فإنها ترجع عدد النظمات المفارقة بهذه الحقول (إذ يتعلق الأمر هنا بما يشبه الاستعلام `SELECT DISTINCT MonChamp1, MonChamp2 ..`)

4.5.2.2 - الخاصية *Fields*

Type : Collection

هذه المجموعة تضم جميع الحقول المكونة للفهرس ، وكل كائن Field في هذه المجموعة يجب أن يمتلك نفس الاسم للكائن Field بجدول الضم .

4.5.2.3 - الخاصية *Foreign*

للقراءة فقط

Type : Boolean


عندما تساوي الخاصية *Foreign* لفهرس القيمة True، فهذا يعني أن الفهرس مفتاح أجنبي ، إنها الحال تماما للفهرس التي ينشئها الأكسيس أليا حينما تقوم بإنشاء علاقات بين الجداول .

4.5.2.4 - الخاصية IgnoreNulls

Type : Boolean
إذا أرسلت القيمة True للخاصية IgnoreNulls ، فإن القيم الخالية لا تفهرس ، وهذا يكون نافعا عندما تقرر تسجيل عدد كبير من القيم الفارغة

4.5.2.5 - الخاصية Name

Type : String
هذه الخاصية تتعلق باسم الفهرس وال الذي يتوجب أن يكون اسما وحيدا في المجموعة Indexes للكائن TableDef المناسب

<p>اختر لفهارسك أسماء واضحة ومعبرة ما أمكن. كأن تقيد هذه الفهارس باختصارات تقرب معناها أو وظيفتها ، مثلا PK_Myfield لحقل مفتاح أساس</p>	
---	---

4.5.2.6 - الخاصية Properties

Type : Collection
مجموعة تضم كل الكائنات من نوع خاصية Property للكائن فهرس Index

4.5.2.7 - الخاصية Required

Type : Boolean
إذا كانت قيمة الخاصية Required تساوي True ، فهذا يعني أن حقول الفهرس تستوجب جميعا إدخال قيم إليها ، وإلا فإن خطأ سيعلن حين محاولة التحديث .

4.5.2.8 - الخاصية Unique

Type : Boolean
إذا كانت قيمة الخاصية Unique تساوي True ، فهذا يعني أن حقول الفهرس لا تقبل التكرار

<p>في فهرس متعدد الحقول ، القيم أ أو ب و ب أ ، ليست قيما مكررة</p>	
--	---

4.5.3 - قائمة الطرائق

كما ستلاحظ ، الكائن فهرس لا يتيح إلا طرائق قليلة ، وهي طرائق سبق التعرض لها أعلاه .

4.5.3.1 - الطريقة CreateField

هذه الطريقة هي نفسها تلك المطبقة على كائنات TableDef مع فرق وحيد أنها هنا ذات معلمة وحيدة (اسم الحقل) وهذا الاسم يجب أن يكون مطابقا لاسم الحقل بالجدول الذي يتكئ عليه الفهرس .

التركيب :

```
Set field = index.CreateField (name)
```

4.5.3.2 - الطريقة CreateProperty

هذه الطريقة لن أعرضها هنا لأنك ستجد حديثا عنها تقريبا في كل أقسام هذا الكتاب ، وأحيلك على الفقرة 4.2.5 الخصائص النوعية .

4.5.4 - إجراءات على الفهارس

سأقترح عليك مثالين تطبيقيين ، وأنا أعلم أنهما كافيين لتعميق فهمك لهذا الجزء من الدرس .

4.5.4.1 - إنشاء فهرس

ليكن لدينا الجدول (Client) به العديد من الحقول بينها واحد (ترقيم تلقائي) و سننشئ مفتاحا أساسا على هذا الحقل .

```
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
Dim oFld As DAO.Field
Dim oInd As DAO.Index
' يهيء قاعدة بيانات
Set oDb = CurrentDb
' يهيء الجدول
Set oTbl = oDb.TableDefs("Client")
' إنشاء الفهرس وتسميته : PK_Client
Set oInd = oTbl.CreateIndex("PK_Client")
' إنشاء الحقل في الفهرس
Set oFld = oInd.CreateField("NumeroClient")
' ضم الحقل إلى المجموعة
oInd.Fields.Append oFld
' تحديد المفتاح الأساس
oInd.Primary = True
' ضم الفهرس للجدول
oTbl.Indexes.Append oInd
' إنعاش المجموعة
oTbl.Indexes.Refresh
```

4.5.4.2 - حذف مفتاح أساس

لتوضيح إجراء حذف فهرس ، أقترح عليك المثال التالي : حيث سندرس حالة حذف المفتاح الأساس من الجدول
هذا الإجراء يستدعي تمرير كل الفهارس بالكائن TableDef ، وحذف كل فهرس تكون قيمة خاصيته Primary تساوي True

```
Private Sub SupprimerClePrimaire(oTbl As DAO.TableDef)
On Error GoTo err
Dim oInd As DAO.Index
' تمرير الفهارس
For Each oInd In oTbl.Indexes
' إن كان مفتاحا أساسا ، إذن يحذفه
If oInd.Primary Then
oTbl.Indexes.Delete (oInd.Name)
End If
Next oInd
Exit Sub
err:
MsgBox "Une erreur inattendue est survenue"
End Sub
```

طريقة استدعاء الدالة : **SupprimerClePrimaire**

```
Dim oDb As DAO.Database
Dim oTbl As DAO.TableDef
'Instancie la base
Set oDb = CurrentDb
'Instancie la table
Set oTbl = oDb.TableDefs("Client")
'Supprime la clé primaire
SupprimerClePrimaire oTbl
```

4.6 - الكائن Relation

4.6.1 - تعريف

إن التطبيقات في العادة تقاس أهميتها انطلاقاً من كفاءتها ونجاعة إجراءاتها ، وعندما يتعلق الأمر بقاعدة بيانات ، فإن شروط التقييم تختلف ، فالدور الأساسي لقاعدة تعريف البيانات SGBD هو تخزين البيانات ، وهذا التخزين يجب أن يتم في ظروف أمان جيدة ، فإنه سيكون كارثياً إذا ما ضاعت بعض البيانات . فتطبيق أكسيس يجب أن يضمن التام وتكامل البيانات . تصور أن جدولاً (Commande) يضم قائمة من الطلبات وجدولاً آخر (Client) يضم قائمة من العملاء ، فإنه سيكون من غير المنطقي أن ترسل طلبية إلى عميل غير وارد في القائمة بالجدول Client ، ومن أجل ألا يقع مثل هذا المحذور ، فإن المحرك Jet يسمح بفرض التكامل المرجعي في العلاقات بين الجداول ، هذا التكامل الذي لا يسمح بإدخال بيانات طلبية في الجدول Commande إلا إذا كان العميل المناسب موجوداً بالجدول Client

4.6.2 - قائمة الخصائص

4.6.2.1 - الخاصية Attributes

Type : Long

الخاصية Attributes تحدد نوع الإلجبار الذي تفرضه العلاقة Relation. والقيمة التي ترسل لهذه الخاصية هي واحدة أو تركيب من قيم الثوابت بالجدول التالي :

الثابتة	دالتها
dbRelationUnique	علاقة رأس برأس
dbRelationDontEnforce	التكامل المرجعي غير مطبق
dbRelationUpdateCascade	تتالي تحديث الحقول المرتبطة
dbRelationDeleteCascade	تتالي حذف السجلات المرتبطة
dbRelationLeft	يكون الرابط من نوع Left
dbRelationRight	يكون الرابط من نوع Righth

4.6.2.2 - الخاصية Fields

Type : Collection

هذه المجموعة تضم كل الحقول لم ستعمله بالعلاقة Relation ، ويمكن استعمال الطريقة Relation.CreateField لإدخال حقول جديدة داخل العلاقة ، ثم يتعين بعد ذلك ضمها إلى المجموعة Relation.Fields بالطريقة Append

كل كائن Field في العلاقة ، يجب أن يمتلك اسماً هو الاسم نفسه للحقل المعني بالجدول المخصص بالخاصية Table. وهذا الحقل سيكون له خاصية ForeignName ، إذا كان يمتلك اسماً هو الاسم نفسه للحقل المعني بالجدول المخصص بالخاصية ForeignTable



- 4.6.2.3 الخاصية ForeignTable

Type : String
الخاصية ForeignTable تتعلق باسم الجدول الخارج في العلاقة . وهذا الاسم يجب أن يكون هو نفسه للجدول TableDef المائل في المجموعة TableDefs للقاعدة Database

- 4.6.2.4 الخاصية Name

Type : String
الخاصية Name تمثل اسم العلاقة . ويجب أن يكون اسم ما وجد في المجموعة Database.Relations

استعمال أسماء دالة ، تيسر إجراءات الصيانة ، كما تسهل قراءة الكود لتطويره ، فعلاقة بين جدول الطلبات *commande* و جدول العملاء *client* تسمى مثلا : *Rlt_CommandeClient*



- 4.6.2.6 الخاصية Properties

Type : Collection
المجموعة Properties تضم كل الخصائص للكائن Relation

- 4.6.2.7 الخاصية Table

Type : String
الخاصية Table تتعلق باسم الجدول الرئيسي الذي تعتمد عليه العلاقة ، إنه طبعا الجدول الذي يتضمن المفتاح الأساس في العلاقة . وعلى العكس ، فالخاصية الإسم الخارجي ForeignName يتعلق بالجدول الخارجي الذي يتضمن مفتاحا خارجيا في العلاقة .

- 4.6.3 قائمة الطرائق

الكائنات من نوع علاقة Relation لا تمتلك إلا طريقة واحدة CreateField ، وهذه الطريقة تسمح بإنشاء حقل لإضافته إلى العلاقة .

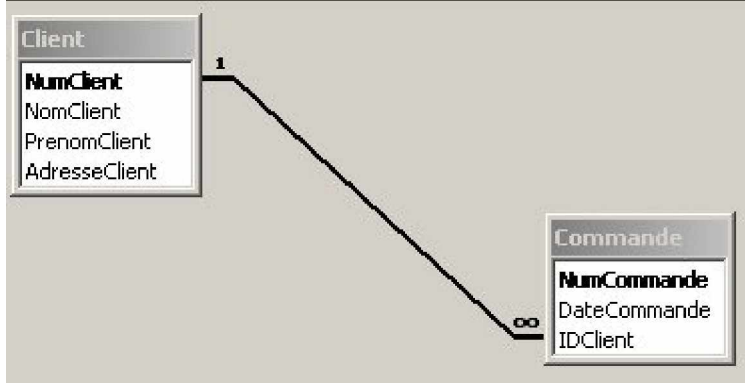
التركيب :

```
Set field = relation.CreateField (name)
```

Name يتعلق باسم حقل المفتاح الأساس في العلاقة .

4.6.4 - مثال تطبيقي عن تعريف الخصائص

لنفترض جدولين وعلاقة :



تعريف معلمات العلاقة :



وخصائص العلاقة أيضا تعرف كالتالي :

- dbRelationUpdateCascade = Attributes
- Fields : كائن Field يسمى "NumClient" ومن من خصائصه :

"IDClient" = ForeignNam

- "Commande" = ForeignTable
- "Rlt_CommandeClient" = Name
- False = PartialReplica (لا يستعمل)
- "Client" = Table

4.6.5 - إجراءات على الخصائص

سأقتراح عليك متتالية من الأمثلة لتتمكن من بلورة تصور عن معالجة كائنات علاقة بالكود VBA

4.6.5.1 - إنشاء علاقة

لنعد إلى المثال السابق ، سأقدم لك كودين ينجزان مثل تلك العلاقة :
الطريقة الأولى تركز على مبدأ إنشاء العلاقة و إرسال القيمة المناسبة لكل خاصية من خواصها .

```
Dim oDb As DAO.Database
Dim oRlt As DAO.Relation
Dim oFld As DAO.Field
Set oDb = CurrentDb
' ينشئ العلاقة
Set oRlt = oDb.CreateRelation
' يعرف الخصائص
oRlt.Attributes = dbRelationUpdateCascade
oRlt.ForeignTable = "Commande"
oRlt.Name = "Rlt_CommandeClient"
oRlt.Table = "Client"
' ينشئ الحقل في العلاقة
Set oFld = oRlt.CreateField("NumClient")
' يعرف اسم المفتاح الخارجي
oFld.ForeignName = "IDClient"
' يضم الحقل
oRlt.Fields.Append oFld
' ينعش المجموعة Relations
oDb.Relations.Refresh
' يضم العلاقة
oDb.Relations.Append oRlt
' ينعش المجموعة Relations
oDb.Relations.Refresh
' يغلق قاعدة البيانات
oDb.Close
Set oDb = Nothing
```

الطريقة الثانية تركز إلى استعمال كل معلمات الطريقة CreateProperty وهي تأخذ شكل التركيب

التالي :

```
Set relation = database.CreateRelation (name, table, foreigntable, _
attributes)
```

والكود إذن :

```
Dim oDb As DAO.Database
Dim oRlt As DAO.Relation
Dim oFld As DAO.Field
Set oDb = CurrentDb
' ينشئ العلاقة
Set oRlt = oDb.CreateRelation("Rlt_CommandeClient", "Client", _
"Commande", dbRelationUpdateCascade)
' ينشئ الحقل
Set oFld = oRlt.CreateField("NumClient")
' يعرف اسم المفتاح الخارجي
oFld.ForeignName = "IDClient"
' يضم الحقل
oRlt.Fields.Append oFld
' ينعش المجموعة Relations
```

```

oDb.Relations.Refresh
' يضم العلاقة
oDb.Relations.Append oRlt
' ينعش المجموعة Relations
oDb.Relations.Refresh
' يغلق قاعدة البيانات
oDb.Close
Set oDb = Nothing
    
```

- 4.6.5.2 إعطاء العلاقات التي تستخدم حقلًا معطى

هذه الدالة سبق استعمالها عندما حاولنا حذف حقل من جدول :
والمبدأ الذي يرتكز إليه حقل مستعمل من طرف علاقة هو :

- الجدول حيث يوجد هذا الحقل ، و يرسل كقيمة للخاصية Table في العلاقة ، واسم حقل من هذا الجدول مطابق لاسم الحقل في العلاقة .
- الجدول حيث يوجد هذا الحقل ، و يرسل كقيمة للخاصية ForeignTable في العلاقة ، وعلى الأقل حقل واحد من حقول هذه العلاقة يمتلك خاصية ForeignName قيمتها مطابقة لاسم هذا الحقل .

و سديفينا إذن تمرير كل كائنات علاقة في المجموعة لقاعدة البيانات عبر حلقة تكرارية في الدالة

: **ChampDansRelation**

```

Private Function ChampDansRelation(oBaseDeDonnees As DAO.Database, _
    strNomTable As String, strNomChamp As String) As Integer
On Error GoTo err
Dim bolUtilise As Boolean
Dim oRlt As DAO.Relation
Dim oFld As DAO.Field
' تمرير العلاقات
For Each oRlt In oBaseDeDonnees.Relations
    bolUtilise = False
    If oRlt.Table = strNomTable Then
        ' لكل حقل يتم اختيار الاسم
        For Each oFld In oRlt.Fields
            If oFld.Name = strNomChamp Then bolUtilise = True
        Next oFld
    ElseIf oRlt.ForeignTable = strNomTable Then
        ' لكل حقل يتم اختيار الاسم الخارجي
        For Each oFld In oRlt.Fields
            If oFld.ForeignName = strNomChamp Then bolUtilise = True
        Next oFld
    End If
    ' كلما استعمل الحقل من علاقة تنضاف القيمة +1 إلى النتيجة
    If bolUtilise Then ChampDansRelation = ChampDansRelation + 1
Next oRlt
Exit Function
err:
ChampDansRelation = -1
End Function
    
```

إنه إذن من الممكن معرفة عدد العلاقات التي تستعمل الحقل NumClient من الجدول Client
طريقة استدعاء هذه الدالة :

```
Dim oDb as DAO.Database
Set oDb=CurrentDB
MsgBox ChampDansRelation(oDb, "Client", "NumClient")
```

4.6.5.3 - حذف العلاقات المحيطة بجدول

قد يحدث مرة أن يكون من الضروري ، بل والنافع أيضا ، حذف العلاقات المحمولة من جدول . لتمكين التعديل على بنيتها . وه ذا الإجراء من المؤكد ذاته مفضل للغاية ، وأذا اقتدر عليك الدالة : **SupprimerRelationTable** التي ستنتج هذه المهمة عنك .

```
Public Function SupprimerRelationTable(oBaseDeDonnees As _
    DAO.Database, strNomTable As String) As Integer
Dim oRlt As DAO.Relation
' لكل علاقة
For Each oRlt In oBaseDeDonnees.Relations
' إن كان الجدول يستعملها
If oRlt.Table = strNomTable Or oRlt.ForeignTable = strNomTable Then
' حذف هذه العلاقة
oBaseDeDonnees.Relations.Delete oRlt.Name
oBaseDeDonnees.Relations.Refresh
' يرفع من قيمة العداد
SupprimerRelationTable = SupprimerRelationTable + 1
End If
Next oRlt
End Function
```

الدالة **SupprimerRelationTable** ترجع عدد العلاقات المحذوفة

التركيب :

```
Dim oDb As DAO.Database
Dim oRlt As DAO.Relation
Set oDb = CurrentDb
'Supprime les relations de la table Client
MsgBox SupprimerRelationTable(oDb, "Client")
```

4.7 - الكائن QueryDef

نحن الآن بصدد دراسة آخر عنصر من كائنات القسم (تعريف البيانات DDL في النموذج DAO) هذا القسم الذي بعجالة نتبعنا فيه قوائم الخصائص والطرائق لهذه الكائنات ثم قدمنا أمثلة تطبيقية عنها بالـVBA

4.7.1 - تعريفات

أكسيس من أندر قواعد البيانات العلائقية SGBDR التي تمكن من تخزين استعلامات SQL داخل القاعدة من غير أن تكون عرضا view . بالتأكيد العرض هو ناتج الاستعلام ، ولكن محدثا ثابت ، ولاستعراض التعديلات التي طرأت على مجموعة من السجلات بين لحظتين ، من الضروري إعادة إنشاء العرض . بينما الاستعلام في قاعدة بيانات أكسيس تبقى حيوية ، فهي في كل مرة تفتح تعيد تنفيذ تعليمات SQL بها . وهذا المبدأ روعي عند نمذجة QueryDef في DAO . وهذه الاستعلامات يمكن أن تكون : إما للتحديد ، للإنشاء ، للإلحاق ، للتحديث ، للحذف أو تكون استعلاما جدوليا .

الاستعلامات المعيارية :

يمكن النموذج DAO من تدبير المعلمات في الاستعلامات عن طريق المجموعة Parameters للكائن QueryDef ، هذه الاستعلامات تقريبا أهمها في أكسيس ، واسد تعويض عنها باستعلامات SQL المشفرة (المكودة) مباشرة في الـVBA. رغم أن معالجة تلك الاستعلامات سهلة للغاية :

```
{ PARAMETERS Param_Num Text ( 10 ); } 1
{ SELECT *                               } 2
{ FROM Client                             }
{ WHERE NumClient=Param_Num;              }
```

الجزء الأول معيار PARAMETERS ، وهو مجموعة تضم قائمة من المعلمات (متغيرات مع أنواعها) (واختياريا مع أطوالها))

الجزء الثاني : الاستعلام والذي ستضمنه المعلمات

وهكذا لفتح كائن السجلات Recordset على سجل العميل الذي يحمل رقم AF36. سنكتب الكود التالي

```
Dim oDb As DAO.Database
Dim oQdf As DAO.QueryDef
Dim oRst As DAO.Recordset
Set oDb = CurrentDb
Set oQdf = oDb.QueryDefs("R01")
oQdf.Parameters("Num").Value = "AF36"
Set oRst = Qdf.OpenRecordset
```

وللمقارنة هاهو الكود نفسه ، ولكن لم يستعمل فيه الاستعلام المعياري

```
Dim oDb As DAO.Database
Dim oSQL As String
Dim oRst As DAO.Recordset
```

```
Dim strSQL As String
Set oDb = CurrentDb
strSQL = "SELECT * FROM Client " & _
        "WHERE NumClient=" & Chr(34) & "AF36" & Chr(34)
Set oRst = oDb.OpenRecordset(strSQL)
```

استخدام الاستعلامات المعيارية يبدو أ كثر فعالية ، ويريح من استعمال دالات تحويل السلاسل ، وأكثر من ذلك إذا ما رغبتنا في المستقبل تعديل الاستعلام لزيادة ارتباط أو غيره ، فإنه يكفي تعديل كود SQL المناسب في الاستعلام المعياري ، ولو لم نستعمل مثل هذا الاستعلام ، لكان علينا تعديل كود VBA في كل مكان يشار به إلى الاستعلام .

إذا فتحت كائن قاعدة بيانات Db2 من القاعدة الحالية Db1 ، ونفذت كائن استعلام QueryDef منضم للقاعدة Db2 ، فإن البيانات ، ستستدعى للاستعلام من القاعدة Db2 ، وليس من القاعدة الحالية Db1.(كل كائن استعلام QueryDef يشتغل على بيانات القاعدة Database التي تحتويه)



4.7.2 - قائمة الخصائص

سنستعرض تاليه قائمة الخصائص المتوفرة للكائن QueryDef

الخاصية Prepare غير موثقة في هذا الدرس .



4.7.2.1 - الخاصية Connect

للقراءة فقط

Type : String

الخاصية Connect ترجع عبارة(السلسلة النصية) للاتصال المستعملة من لدن الاستعلام عندما ينشأ هذا من قاعدة بيانات أخرى .

4.7.2.2 - الخاصية DateCreated

للقراءة فقط

Type : Date

هذه الخاصية ترجع تاريخ إنشاء الاستعلام

4.7.2.3 - الخاصية Fields

للقراءة فقط

Type : Collection

مرة أخرى نصادف هذه المجموعة التي تضم كل الحقول المرجعة باستعلام .

تذكر أنه بإمكانك التمييز بين اسم الحقل وبين مقروضه alias بأستعمال الخاصية

SourceField للكائن Field



4.7.2.4 - *LastUpdated* الخاصة

للقراءة فقط

Type : Date

هذه الخاصية ترجع تاريخ آخر تعديل على الاستعلام

4.7.2.5 - *MaxRecords* الخاصة

Type : Long

تحدد هذه الخاصية عدد السجلات التي سيرجعها الاستعلام كحد أقصى ، وهذه الخاصية تكون ذات فائدة حين يكون الجهاز الحاسب المستعمل ، لا يتوفر على مصادر ذاكرة كافية لمعالجة كافة البيانات .

هذه الخاصية متوفرة فقط مع الاستعلامات التي تستعمل مصدر البيانات ODBC



4.7.2.6 - *Name* الخاصة

Type : String

الخاصية Name تمثل اسم الاستعلام ، ويجب أن يكون اسما فريدا في المجموعة QueryDefs لقاعدة

البيانات

إذا استخدمت الطريقة CreateQueryDef ولم تحدد معلمة الاسم Name ، فإن الاستعلام لا يحفظ في قاعدة البيانات حتى نهاية المعالجة .



4.7.2.7 - *Parameters* الخاصة

Type : Collection

هذه المجموعة تضم كل معلمات المعيار للكائن استعلام QueryDef. وإنه من المستحيل حذف أو إضافة عنصر إلى هذه المجموعة .

مثال تطبيقي عن استعمال الاستعلام المعياري سبق التعرض له في الأسطر أعلاه .



4.7.2.8 - *Properties* الخاصة

Type : Collection

مجموعة Properties تضم كل الكائنات من نوع خاصية (Property) للكائن QueryDef

4.7.2.9 - *RecordsAffected* الخاصة

للقراءة فقط

Type : Long


في حالة استعلام تدخل (إلحاق، تحديث أو حذف) فإن الخاصية RecordsAffected ترجع عدد

السجلات التي طالها الحدث

4.7.2.10 - الخاصية ReturnsRecords

Type : Boolean


هذه الخاصية تحدد أو ترجع قيمة تشير إلى حال الاستعلام هل يرجع سجلات أم لا .

هذه الخاصية متوفرة فقط مع اسعلامات SQL المباشرة	
---	---

4.7.2.11 - الخاصية SQL

Type : String

يتعلق الأمر بكود SQL للاستعلام .

هذه الخاصية هي قيمة المعلمة SQL في الطريقة Database.CreateQueryDef	
--	---

4.7.2.12 - الخاصية Type

Type : Long

هذه الخاصية ترجع نوع الاستعلام ، وهي تحدد ألياً عند إنشائك الكائن QueryDef ، وبذلك هي تمكنك من إحصاء أنواع الاستعلامات لديك بالقاعدة .

القيم الممكنة لهذه الخاصية هي :

الثابتة	دلالاتها
dbQAction	تدخل
dbQAppend	إضافة
dbQCompound	غير موثقة
dbQCrosstab	جدولي
dbQDDL	تعريف البيانات
dbQDelete	حذف
dbQMakeTable	إنشاء
dbQProcedure	إجراءات مخزنة
dbQSelect	تحديد
dbQSetOperation	انضمام
dbQSPTBulk	استعلام لايرجع بيانات
dbQSQLPassThrough	استعلام SQL مباشر
dbQUpdate	تحديث

4.7.2.13 - الخاصية *Updatable*

للقراءة فقط

Type : Boolean

هذه الخاصية ترجع True إذا كانت بيانات الاستعلام لا يمكن التعديل عليها .

4.7.3 - قائمة الطرائق

4.7.3.1 - الطريقة *CreateProperty*

الطريقة *CreateProperty* تنشئ كائنا جديدا *Property* (ولمعلومات أكثر تفصيلا أحيلك على الفقرة 4.2.5 الخصائص النوعية)

4.7.3.2 - الطريقة *Execute*

الطريقة *Execute* تنفذ استعلام تدخل ، وإن خطأ سيعلن إن حاولت بها تنفيذ استعلام تحديد

التركيب :

```
querydef.Execute options
```

المعلمة *options* يجب أن تكون واحدة أو تركيبا من الثوابت التالية :

الثابتة	دالاتها
dbDenyWrite	منع الوصول للكتابة بالنسبة للمستخدمين
dbInconsistent	(قيمة افتراضية) تنفذ التحديثات دون مراعاة التوافق (التكامل المرجعي)
dbConsistent	تنفذ التحديثات مع مراعاة التوافق (التكامل المرجعي)
dbSQLPassThrough	تنفذ استعلام SQL مباشر
dbFailOnError	في حال الخطأ ، تلغى جميع التحديثات
dbSeeChanges	تعلن خطأ إن حاول مستخدم آخر تحديث البيانات التي أنت في طور تحديثها

يمكنك استخدام هذه الخاصية لمعرفة عدد التسجيلات التي طالها التعديل



4.7.3.3 - الطريقة *OpenRecordset*

الطريقة *OpenRecordset* تفتح مجموعة سجلات كنتيجة للاستعلام .

التركيب :

```
Set recordset = object.OpenRecordset (type, options, lockedits)
```

هذه الطريقة سنقف عندها طويلا في القسم الموالي :



4.7.4 - إجراءات على الاستعلامات

4.7.3.1 - إنشاء استعلام

في مثالنا هذا سنحاول إنشاء استعلام تحديد كامل للجدول clients ، إنه استعلام بسيط ، لكن الطريقة المطبقة تبقى هي نفسها مع كل الاستعلامات مهما اختلفت أنواعها .

```
Sub CreerRequete()
On Error GoTo Err
Dim oDb As DAO.Database
Dim strCodeSql As String
' يهيء القاعدة
Set oDb = CurrentDb
' في متغيرة نصية SQL يخزن كود
strCodeSql = "SELECT * FROM Client"
' ينشئ الاستعلام
oDb.CreateQueryDef "R01_SelectionClient", strCodeSql

Finally:
oDb.Close
Set oDb = Nothing

Err:
' يعالج الأخطاء
Select Case Err.Number
Case 3012: MsgBox "الاستعلام موجود مسبقا بالقاعدة"
Case Else: MsgBox "حدث خطأ غير معروف"
End Select
Resume Finally
End Sub
```

4.7.4.2 - تعديل الكود استعلام

- يمكنك التعديل على كود SQL باستعلام ، يجب الامتثال لخطوتين اثنتين :
- استرجاع الكائن QueryDef المطابق (إن كان موجودا)
 - تعديل الخاصية SQL لهذا للكائن .

```

Public Function ModifierSQL(oBaseDeDonnees As DAO.Database, _
    strNomRequete As String, strCodeSQL As String) As Boolean
On Error GoTo Err
Dim oQdf As DAO.QueryDef
'Instancie l'objet QueryDef
Set oQdf = oBaseDeDonnees.QueryDefs(strNomRequete)
'Modifie le code SQL
oQdf.SQL = strCodeSQL
'Renvoie True
ModifierSQL = True
Exit Function
Err:
'Gere les erreurs
Select Case Err.Number
    Case 3265: MsgBox "Impossible de trouver la requête à modifier"
    Case Else: MsgBox "Une erreur inattendue est survenue"
End Select
End Function
    
```

طريقة استدعاء الدالة :

```

Dim oDb As DAO.Database
Set oDb = CurrentDb
If ModifierSQL(oDb, "R01_SelectionClient", "SELECT NumClient FROM
Client") Then
    MsgBox " تم تعديل كود الاستعلام SQL بنجاح "
End If
    
```

4.7.4.3 - عرض الاستعلامات

الخاصية Type ترجع نوع الكائن استعلام ، إنه إذن ممكن استعراض استعلامات التحديد

```

Dim oDb As DAO.Database
Dim oQdf As DAO.QueryDef
'تهيئة قاعدة البيانات
Set oDb = CurrentDb
'مكرر الاستعلامات
For Each oQdf In oDb.QueryDefs
    'إن كان استعلام تحديدي ، إذن يظهر الاسم
    If oQdf.Type = dbQSelect Then MsgBox oQdf.Name, , "Sélection"
Next oQdf
    
```

القسم الخامس :

- الوصول إلى البيانات

5.1 - تعريفات

في الفيجوال بيزيك ، التطبيق قادر على الوصول إلى البيانات عبر كائن السجلات Recordset ، هذا الكائن متعلق بمجموعة سجلات ، وعادة ما يدعى بالإنجليزي CURSOR، وهو اختصار لـ (مجموعة الصفوف الحالية CURrent Set Of Rows).

إذا كانت هذه خطواتك الأولى في عالم السجلات ، يمكنك أن تتصور هذا مثلا : جدول استعلام في طريقة العرض (ورقة بيانات) فكائن السجلات Recordset إذن مكون من عدة صفوف (سجلات) و عدة أعمدة (حقول) وعندما تصل إلى قيمة حقل ما ، فأنت تفعل ذلك بالنسبة للسجل الحالي بكائن السجلات .

وعلى جدول استعلام في طريقة العرض (ورقة بيانات) فإن السجل الحالي مجهز بمثلث يظهر في محددات السجلات أعلى أقصى يمين السطر أو يساره حسب اللغة المستعملة (، وينقلب إلى شكل قلم عند إدخال البيانات بالحقل .

السجل الحالي هنا

رقم معرف	الاسم	الرقم الوظيفي	الترتبه	الا
2	منصور	10101	الاولى	ال
3	كمال	11111	الاولى	
4	أحمد	21212	الاولى	
5	سفيان	22222	الاولى	


كائن السجلات Recordset يعطي إذن فرصة استرجاع نتائج استعلام ، وإرسال ذلك إلى متغيرة ، أو فرصة تحليلها أو تعديلها... إلخ (تماما كما تفعل ذلك يدويا على ورقة بيانات أو نموذج) .

5.1 - أنواع كائنات السجلات Recordset

هناك خمس أنواع منها :

1. كائن سجلات من نوع جدول :

كائن سجلات Recordset من نوع جدول يسمح بالوصول مباشرة إلى محتوى جدول ، ومحتوى هذا الكائن حيوي (دينامي) ، وهذا يعني أن تحيين البيانات مباشرة يؤثر على السجلات . ومصدر بيانات هذا النوع من كائنات السجلات هو اسم لجدول .

انتبه ، فهذا النوع من كائنات السجلات غير متوافق مع الجداول المرتبطة (الجداول الموجودة بقاعدة بيانات خارجية) 

2. كائن سجلات من نوع مجموعة حيوية :

كائن سجلات من نوع مجموعة حيوية Dynaset ، يطابق نتائج تعليمة SQL ترجع سجلات ، محتواها حيوي ومصدره إما اسم الجدول أو تعليمة SQL متحققة .

3. كائن سجلات من نوع لقطة :

كائن سجلات من نوع لقطة Snapshot يطابق لقطة فوتوغرافية للبيانات في لحظة معينة (لحظة فتح الكائن) ، والتغييرات الحادثة على قاعدة البيانات بعد فتح مثل هذا الكائن لا تتأثر بها السجلات الماثلة فيه ، فمحتواه إذن ثابت (استاتيكي) ، ومصدره إما اسم الجدول أو تعليمة SQL متحققة .

4. كائن سجلات من نوع تقديمي فقط :

كائن سجلات من نوع تقديمي فقط ForwardOnly هو نفسه ذلك من نوع لقطة مع فارق بسيط ، أن هذا لايسمح بالتنقل بين السجلات بحرية ، بحيث يمتنع عن التراجع إلى سجل سابق عن الحالي .

5. كائن سجلات من نوع حيوي :

هذا النوع من كائن السجلات مشابه للأول Dynaset مع فرق وحيد أن هذا يطبق مع مصدر البيانات المباشر ODBC

5.2 - فتح كائن السجلات *Recordset*

من الأكسيس ، ومن VBA هناك أكثر من طريقة لفتح مجموعة السجلات

5.2.1 - انطلاقا من كائن Database

الكائن Database (وبطريقة غير مباشرة CurrentDB) مزود بالطريقة OpenRecordset التي تنشئ نسخة لكائن سجلات جديد وتضيفه إلى المجموعة Database.Recordsets : وتركيب هذه الطريقة هو التالي :

```
Set recordset = Database.OpenRecordset (source, [type, options, _ Lockedits])
```

Source : معلمة تتعلق بمصدر السجلات للكائن Recordset . و يمكن أن يكون اسم جدول ، أو استعلام ، أو تعليمة SQL متحققة (SELECT) ، ترجع مجموعة من السجلات . وفي حالة كائن السجلات من نوع جدول ، فهذه المعلمة تقتصر على اسم الجدول فقط . لتكون تعليمة SQL متحققة يجب أن تأخذ بعين الاعتبار الأنواع المتعددة للحقول في تعبيرات المعيار ، وهكذا إن كان الحقل NomClient من نوع نص ، فإن المتغيرة المصدر التي تنتقي العميل " محمد العلوي " تكون كالتالي :

```
Source="SELECT * FROM Client WHERE NomClient=" & chr(34) & _ "محمد العلوي" & chr(34)
```

الدالة Chr(34) تسمح بتأطير المعلمة من نوع نص بالمزدوجتين ، لأنه في الواقع سيعتبر خطأ إملائيا كتابتنا للكود بالشكل التالي :

```
Source="SELECT * FROM Client WHERE NomClient=" محمد العلوي "
```

ففي هذه الحالة المزدوجة (" قبل (محمد العلوي) هي لإغلاق المزدوجة قبل (SELECT) . خطأ إذن سيعلنه المجمع compilateur ، والحل الوحيد في هذه الحالة ، هو استعمال كود المحارف Ascii ، و إرساله إلى معلمة (متغيرة) الدالة Chr ، لترجعه حرفاً يقرأ بعد السلسلة في مجمع الكود كالتالي :

```
SELECT * FROM Client WHERE NomClient=" محمد العلوي "
```

غير أنه تبقى هناك مشكلة أخرى ؛ فإذا كانت إحدى القيم المبحوت عنها تتضمن مزدوجة ، فإن المحرك Jet سيؤولها هي أيضا كنهاية للسلسلة النصية ، والحل أن نضاعف المزدوجة في عبارة المعيار كالتالي

```
Source="SELECT * FROM Client WHERE NomClient=" & chr(34) & _ Replace(MaVariable,chr(34),chr(34) & chr(34)) & chr(34)
```

و هكذا إذا كانت المتغيرة تساوي السيد "X" ، إذن الدالة ترجع : السيد ""X"" ، وتصيح العبارة المصدر لكائن السجلات كما يلي :

```
SELECT * FROM Client WHERE NomClient=""X"" السيد "
```

أما النوع تاريخ فإنه الأسهل للمعالجة ، إذ يجب مراعاة تأطير قيمة المعلمة بالرمز # ، كما يسجل أن عبارة التاريخ تأخذ التنسيق الأمريكي : mm/jj/yyyy
فكائن السجلات الذي سينتقي الطلبات ليوم : 10 ديسمبر 2003 ستكون عبارة مصدره كالتالي :

Source="SELECT * FROM Commande WHERE DateCommande=#12/10/2003#"

Type : هذه المعلمة ترتبط بنوع كائن السجلات والذي يأخذ قيمه من جدول الثوابت التالي :

نوع السجلات	الثابتة
جدول	dbOpenTable
حيوي (ODBC مباشر)	dbOpenDynamic
ورقة بيانات حيوية	dbOpenDynaset
لحظي	dbOpenSnapshot
فقط تقدمي	dbOpenForwardOnly

Options : أما هذه المعلمة فتحدد المعايير الاختيارية لكائن السجلات ، ويتعلق الأمر بوحدة أو تركيب من الثوابت التالية :

مفعولها	الثابتة
تمكن المستخدمين من إضافة سجلات جديدة للكائن Recordset ، وتمنعهم من التعديل أو الحذف على السجلات الحالية (كائن السجلات من نوع ورقة بيانات حيوية) في المحرك Jet ، (Dynaset) فقط .	dbAppendOnly
ترسل تعليمة SQL إلى مصدر بيانات مباشرة ODBC ، متصلة بالمحرك Jet من زاوية المعالجة (كائن السجلات من نوع لحظة) في المحرك Jet ، (Snapshot) فقط .	dbSQLPassThrough
يعلن خطأ التنفيذ إذا ما حاول مستخدم تعديل بيانات في طور التعديل من مستخدم آخر (كائن السجلات من نوع ورقة بيانات حيوية) في المحرك Jet ، (Dynaset) فقط .	dbSeeChanges
يمنع على المستخدمين الإضافة أو التعديل على السجلات	dbDenyWrite
يمنع على المستخدمين قراءة البيانات من جدول .	dbDenyRead
يسمح بالتحديث المفكك (لا يراعي فرض التكامل المرجعي)	dbInconsistent
يسمح فقط بالتحديث الذي يراعي فرض التكامل المرجعي .	dbConsistent

بعض الثوابت أغفلت عن عمد في هذه اللائحة ، على اعتبار أنها لا تستعمل في تطبيقات أكسيس ، أو لأنها غير منصوص باستعمالها .

و المعلمة LockEdits : تتعلق بأسلوب إغلاق السجلات الذي يستعمله كائن Recordset . والثوابت المتوفرة هي :

أسلوب الإغلاق	الثابتة
منع المستخدم من التعديل على الكائن Recordset	dbReadOnly
تستخدم أسلوب الإغلاق المتحفظ ، من أجل تحديد نمط التعديلات التي يتطلبها الكائن Recordset في مجال متعدد المستخدمين . الصفحة الحاوية للسجلات التي أذت بصدد التعديل عليها تغلق بمجرد تطبيق الطريقة Edit(وهو خيار افتراضي في مجالات العمل للمحرك Microsoft Jet)	dbPessimistic
تستخدم أسلوب الإغلاق المتفائل ، من أجل تحديد نمط التعديلات التي يتطلبها الكائن Recordset في مجال متعدد المستخدمين . الصفحة الحاوية للسجلات التي أذت بصدد التعديل عليها لا تغلق طالما لم يستدعى تطبيق الطريقة Update .	dbOptimistic

بعض الثوابت أغفلت عن عمد في هذه اللائحة ، على اعتبار أنها لا تطبق في تطبيقات أكسيس



مثال :

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset ("SELECT * FROM Client WHERE NumClient=2",dbOpenDynaset)
```

هذا المثال يفتح كائن يرجع العميل الذي رقمه 2

5.2.2 - انطلاقاً من كائن QueryDef

إذا رغبت الوصول إلى بيانات استعلام مسجل بقاءة البيانات ، فإنه بإمكانك استعمال الطريقة OpenRecordset للكائن QueryDef المناسب ، وكذا استعمال تلك الطريقة للكائن Database ، غير أنها تتطلب منك أن ترسل اسم الاستعلام كمعلمة .
وتركيب هذه الطريقة للكائن QueryDef كالتالي :

```
Set recordset = QueryDef.OpenRecordset ([type, options, lockedits])
```

معلومات المعيار المستعملة مثل تلك المستعملة في الطريقة Database.OpenRecordset ، مع ملاحظة أنه لا يمكن فتح سجلات من نوع جدول على كائن استعلام .

في حالة استعلام معياري ، فإنه من الضروري تمرير معلومات المعيار قبل فتح السجلات ، ومن المستحيل فتح السجلات على كائن استعلام يطالب المستخدم بالمعيار عبر علبة الإدخال (inputbox) ، وسيعلن الخطأ التالي (خطأ في التنفيذ... × معايير منتظرة) ، في كل مرة



5.2.3 - انطلاقاً من كائن نموذج

دون أن تعلم ، فأنت دائماً تعالج كائن السجلات بواسطة نموذج ، ففي الواقع كل النماذج أو التقارير بالأكسيس تتضمن مصدراً للبيانات (خاصية RecordSource / Source) تستعمل (مجموعة الصفوف الحالية) (curseur) لمعالجة البيانات التي تحتويها ، كما تستطيع التأثير على كائن السجلات فإذا استعملت

الطريقة Movenext لسجلات النموذج Recordset ، فستلاحظ أن النموذج يرتبط بالسجل التالي مباشرة ولنسجل أن أي تنقل في سجلات نموذج ، ينشط استدعاء حدث Form_Current لتنشيط النموذج .

التركيب :

```
Set recordset=Me.Recordset
```

والطريقة RecordsetClone للنموذج ، تسمح باستنساخ سجلات النموذج ، وبذلك يمكن معالجتها دون أن تنعكس نتائج المعالجة على النموذج ، ومن غير استثارة أي حدث . وليأخذ النموذج التغييرات بعين الاعتبار ، يجب استدعاء الطريقة Requery لهذا النموذج .

التركيب :

```
Set recordset=Me.RecordsetClone
```

5.3 - المواقع والتنقلات

5.3.1 - التنقل داخل السجلات Recordset

عندما تهيء كائن سجلات ، فإن السجل الأول هو السجل الحالي إن لم يكن كائن السجلات فارغا ، وعلى العموم هناك موقعان متميزان :

- قبل أول سجل هناك موقع تساوي عنده قيمة الخاصية BOF للسجلات True
 - وبعد آخر سجل موقع آخر تساوي عنده قيمة الخاصية EOF للسجلات True
- وفي حالة كون كائن السجلات فارغ ، فإن موقع السجل الحالي = قبل أول سجل وبعد آخر سجل في نفس الوقت ، وهكذا لاختبار هل كائن السجلات فارغ أم لا ، يكفي استعمال الشرط التالي :

```
If Not oRecordset.EOF Then
    MsgBox "كائن السجلات ليس فارغا"
else
    MsgBox "كائن السجلات فارغ"
End if
```

خمس طرق تسمح بالتنقل داخل السجلات :

- MovePrevious للوصول إلى السجل السابق
- MoveNext للوصول إلى السجل الموالي
- MoveFirst للوصول لأول سجل
- MoveLast للوصول لآخر سجل
- Move للانتقال إلى سجل معين اعتبارا من الموقع الحالي

أمثلة :

```
' يتموقع على أول سجل
oRecordset.MoveFirst
' يتموقع على آخر سجل
oRecordset.MoveLast
' يتموقع على السجل السابق
oRecordset.MovePrevious
' يتموقع على السجل التالي
oRecordset.MoveNext
' يتراجع عن السجل الحالي بثلاث مواقع
oRecordset.Move -3
' يتقدم على السجل الحالي بثلاث مواقع
oRecordset.Move 3
```

عندما تصل لآخر سجل ، وتستخدم الطريقة MoveNext ، فإن قيمة الخاصية EOF للسجلات ستساوي True ، والموقع الحالي ليس سجلا ، و بالتالي فإن خطأ في التنفيذ سيعلن .. والشيء نفسه إن تراجع لما قبل السجل الأول فإن قيمة الخاصية BOF للسجلات ستساوي True ، والموقع الحالي ليس سجلا ، و بالتالي فإن خطأ في التنفيذ سيعلن .



لتمرير كل سجلات الكائن ، استخدم الإجراء التالي :

```

ما دام لم يطل آخر سجل '
While Not oRecordset.EOF
    انتقل إلى التالي '
    oRecordset.MoveNext
Wend

```

- 5.3.2 البحث عن سجل

الطرائق التي يبدأ اسمها بـ (*Find*) تسمح بالوصول لسجل يحقق شرط المعلمات المرسله للمعيار . وهذا المعيار يماثل الاشتراط WHERE في استعلام SQL والاختلاف بين طرائق البحث يكمن في اتجاه البحث :

- FindNext تبحث في السجلات الموالية للسجل الحالي عما يحقق معلمات المعيار
- FindPrevious تنفذ نفس البحث ولكن للخلف
- FindFirst تنفذ البحث ابتداء من أول سجل
- FindLast تنفذ البحث ابتداء من آخر سجل

وهكذا للوصول إلى أول سجل في النموذج يبدأ فيه اسم العميل (NomClient) بحرف الجيم نكتب الكود التالي :

```

Dim strCritere as string
strCritere="NomClient LIKE " & chr(34) & "ج*" & chr(34)
Me.Recordset.FindFirst strCritere

```

إذا لم يتحقق اشتراط المعيار في أي من سجلات النموذج فإن الخاصية NoMatch للسجلات ستساوي True

مثال :

```

Dim strCritere as string
strCritere="NomClient LIKE " & chr(34) & "ج*" & chr(34)
Me.Recordset.FindFirst strCritere
If Me.Recordset.NoMatch Then MsgBox " لا يوجد أي اسم مطابق للمعيار "

```

- 5.3.3 الاحتفاظ بموقع

في بعض الحالات تحتاج إلى التنقل بين السجلات ثم العودة إلى السجل الحالي ، والخاصية Bookmark للسجلات ترجع مؤشرا نحو السجل المراد ، وهذا المؤشر عبارة عن جدول بياناته من نوع Byte

مثال :

```

Dim BytPosition() as Byte
BytPosition=oRecordset.Bookmark

```

وإذا بعد ذلك رغبت في العودة إلى هذا الموقع المؤشر إليه يكفي أن ترسل القيمة المحفوظة إلى الخاصية Bookmark

```

oRecordset.Bookmark=BytPosition

```

5.4 - معالجة البيانات

5.4.1 - قراءة سجل

من الممكن قراءة بيانات السجل الحالي بمساعدة المجموعة Fields ، هذه المجموعة ككل المجموعات في DAO ، تسمح باستدعاء كل عنصر فيها باسمه ، أو بموقعه (فهرسه)

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset("SELECT NumClient,NomClient FROM Client
WHERE NumClient=2",dbOpenDynaset)
Msgbox " : اسم العميل هو " & Rst.Fields("NomClient").Value

' يمرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

أو

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset("SELECT NumClient,NomClient FROM Client
WHERE NumClient=2",dbOpenDynaset)
Msgbox " : اسم العميل هو " & Rst.Fields(1).Value

' يمرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

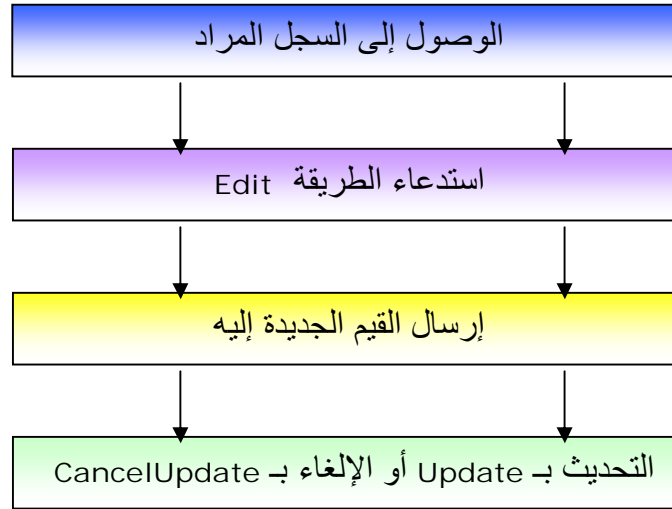
الخاصية Value هي الخاصية الافتراضية للكائن Field ، ويمكنك أيضا التعبير عنها دون ذكرها

مثال :

```
Msgbox Rst.Fields(1)& " : اسم العميل هو " :
```

5.4.2 - تعديل سجل

هناك أربع خطوات لتعديل قيم سجل :



المثال التالي يضبط تاريخ كل الطلبات على تاريخ اليوم :

```

Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset ("TblCommande",dbOpenTable)
While Not oRst.EOF
  ' استدعاء خاصية التعديل
  oRst.Edit
  ' إرسال التاريخ الحالي
  oRst.Fields("DateCommande").Value=Now()
  ' تحديث السجلات
  oRst.Update
  ' الانتقال إلى التالي
  oRst.MoveNext
Wend

' محرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
  
```

مثالنا يحقق تحديثًا بالكتلة على كل سجلات الجدول غير أنه يجب ألا يغيب عنك أنه في كثير من الحالات تكون فيها تعليمات SQL ، أفضل و أقوى من أي إجراء عبر حلقة تكرارية .



```

CurrentDb.Execute "UPDATE TblCommande SET DateCommande=#" & Now() _
& "#"
  
```

5.4.3 - إضافة سجل

الطريقة AddNew للسجلات ، تسمح بإضافة سجل جديد ، وبعد تعيين القيم لمختلف الحقول ، يلزمك استدعاء الطريقة Update ، لتحديث البيانات ، أو CancelUpdate لإلغاء الإضافة . والسجل المنشأ حديثًا

يوضع في آخر موقع بكائن السجلات من نوع جدول ، حيث توضع السجلات في ترتيب يراعي فهارس الجدول ، وعلى كل حال فإن كائن السجلات يرجع إلى موقعه الأصلي قبل استدعاء الطريقة Update

مثال :

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset ("TblProduit",dbOpenTable)
' الانتقال إلى وضع الإضافة
oRst.AddNew
' تعيين القيم لمختلف الحقول
oRst.Fields("Reference").Value="SRS200"
oRst.Fields("Libelle").Value="Souris optique sans fil"
' تحديث
oRst.Update

' تحرير الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

ويلزم لتعيين قيم الحقول أن تكون الخاصية Required ، تساوي True، وإلا فإن خطأ سوف يعلن ، كما يلزم احترام إرغامات الفهرس الذي سبق لك تعيينه على الجدول ، وكذا فإن المفتاح الأساس لا يحتمل التكرار . وفي الحالة التي يكون فيها أحد الحقول من نوع ترقيم تلقائي ، فكثيرا ما و لأسباب متعددة ، تحتاج لمعرفة القيمة المنشأة لهذا الحقل، وهذه القيمة يوفرها المحرك Jet قبل استدعاء الطريقة Update .
ليكن لدينا جدول العملاء " client " حيث الحقل " NumClient " ترقيم تلقائي :

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Dim lngNouvelleValeur as Long
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset ("TblClient",dbOpenTable)
' الانتقال إلى وضع الإضافة
oRst.AddNew
' تعيين القيم لمختلف الحقول
oRst.Fields("NomClient").Value="العلوي"
oRst.Fields("PrenomClient").Value="حمد"
' استرداد المعرف الجديد
lngNouvelleValeur=oRst.Fields("NumClient").Value
' تحديث
oRst.Update
' إعلان رقم العميل في آخر السجل منشأ
Msgbox NouvelleValeur & " : " "قد أنشئ سجل العميل"
```

تحرير الكائنات

```
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

في حالة إضافة عدد مهم من السجلات ، فإن أمر SQL : (تعليمة INSERT) ترسل إلى الطريقة Execute للكائن Database تكون أكثر نجاعة من تحميل الذاكرة بمجموعة صفوف (curseur) تتضمن كل السجلات .



لدينا إذن :

```
Dim StrChaineSQL As String
StrChaineSQL="INSERT INTO TblProduit (Reference,Libelle) VALUES _
 (" & chr(34) & "SRS200" & chr(34) & _
 ", " & chr(34) & "شاشات 17 بومة فلاترون" & chr(34) & " )"
CurrentDb.Execute StrChaineSQL
```

ومع ذلك ، فإنه مع هذا التركيب ، يستحيل الحصول على قيمة حقل ترقيم تلقائي عند إضافة سجل ومن المؤكد أنه يكتفى باسترداد القيمة القصوى للحقل المعني حتى لا يتم اختيار إلا السجل الأخير في الجدول ، ومع ذلك فهذا لا يضمن بأنه السجل الذي أنت بصدد إضافته ، ففي الواقع ، إن مستخدماً آخر يمكن أن يضيف سجلاً مباشرة بعد إضافتك .



considéré

5.4.4 - حذف سجل

الطريقة Delete للكائن Recordset من نوع جدول أو ورقة بيانات حيوية ، يسمح بحذف السجل الحالي ، وبعد الحذف ، يكون الموقع الحالي لكائن السجلات غير معرف ولئلا يعلن خطأ إذا حاولت الدخول ، يلزمك إذن تعديل موقع المؤشر بواسطة طرائق التنقل بين السجلات .

المثال المرفق يحذف جميع طلبات العميل رقم : 13

```
Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' يفتح مجموعة الصفوف
Set oRst=oDb.OpenRecordset("SELECT * FROM TblCommandes WHERE
NumClient=13")
' يمرر السجلات
While Not oRst.EOF
' يحذف السجل
oRst.Delete
' ينتقل إلى التالي
oRst.MoveNext
Wend

' يمرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

5.4.5 - تعداد السجلات

هناك طريقتان يمكن أن تتبعاً لتعداد سجلات كائن Recordset.

- الأولى ترتكب إلى استعمال استعلام SQL ، به التعليمة Count ثم قراءة القيمة المرجعة بها .
وهكذا فإن عدد العملاء الساكنين بمدينة الدار البيضاء ، يمكن تحديده بالكود التالي :

```
Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
Dim lngNbEnregistrement As Long
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' فتح مجموعة صفوف CURSEUR تتضمن عدد العملاء ساكني الدار البيضاء
Set oRst=oDb.OpenRecordset("SELECT Count(*) FROM TblCLients WHERE
Ville=" & Chr(34) & "الدار البيضاء" & Chr(34))
' يقرأ النتيجة
lngNbEnregistrement=oRst.Fields(0).Value

' يحرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

لاحظ أن الاستعلام الذي يستخدم دالة التجميع يرجع على الأقل سجلا واحدا ، وبناء عليه فإنه من غير الضروري اختبار إذا ما كان خاليا ، قبل قراءة السجل الأول .



- الطريقة الثانية ، ترتكب إلى الخاصية RecordCount للكائن Recordset . وهذه الخاصية ترجع عدد السجلات للكائن Recordset من نوع جدول ، أو عدد السجلات التي تم الوصول إليها في باقي الأنواع .
وهكذا فإنه لحساب عدد السجلات في كائن recordset من نوع جدول ، نستعمل :

```
Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
Dim lngNbEnregistrement As Long
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' فتح مجموعة صفوف CURSEUR تتضمن كل العملاء
Set oRst=oDb.OpenRecordset("TblCLients",dbOpenTable)
' بحسب عدد السجلات
lngNbEnregistrement=oRst.RecordCount

' يحرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

وفي حالة نوع آخر ، سنجبر على استعمال الطريقة MoveLast من أجل الوصول إلى آخر سجل في الكائن Recordset ، وإذ ذلك فإن الخاصية RecordCount ترجع عدد السجلات في هذا الكائن :

```
Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
Dim lngNbEnregistrement As Long
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' فتح مجموعة صفوف CURSEUR تتضمن عدد العملاء ساكني الدار البيضاء
```

```

Set oRst=oDb.OpenRecordset("SELECT * FROM TblClients WHERE Ville=" &
Chr(34) & "الدار البيضاء" & Chr(34),dbOpenDynaset)
' يصل إلى آخر سجل
oRst.MoveLast
' بحسب عدد السجلات الموصول إليها (مجموع السجلات)
LngNbEnregistrement=oRst.RecordCount

' يمرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
    
```

5.4.6 - تصفية السجلات

إنه من الممكن فتح كائن سجلات جديد من كائن آخر مفتوح ، نضمنه بيانات مختارة بمساعدة مصفوفة . ولهذا يجب تخصيص مصفوفة للخاصية Filter لكائن السجلات الأول ثم استعمال طريقته OpenRecordset لفتح كائن السجلات الثاني ذي البيانات المصفوفة . المصفوفة المطبقة هي سلسلة نصية تماثل الاشتراط WHERE في الاستعلام . خذ مثالا الجدول TblClients حيث سنختار العملاء ساكني مدينة الدار البيضاء .

```

Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
Dim oRstFiltre as DAO.Recordset
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' فتح مجموعة صفوف curseur تتضمن كل العملاء
Set oRst=oDb.OpenRecordset("SELECT * FROM TblClients",dbOpenDynaset)
' يطبق المصفوفة لتصفية العملاء ساكني الدار البيضاء
oRst.Filter="Ville=" & Chr(34) & "Paris" & Chr(34)
' فتح مجموعة الصفوف curseur من جديد
Set oRstFiltre=oRst.OpenRecordset
*****
' هنا معالجة البيانات المصفوفة
*****
    
```

```

' يمرر الكائنات
oRst.Close
oRstFiltre.Close
oDb.Close
Set oRst=Nothing
Set oRstFiltre=Nothing
Set oDb=Nothing
    
```

لاحظ مع ذلك ، أنه في غالبية الأحيان ، سيكون أكثر سرعة وفعالية فتح كائن سجلات جديد مباشرة من كائن Database مخصصين استعمال SQL كامل للمعلمة source

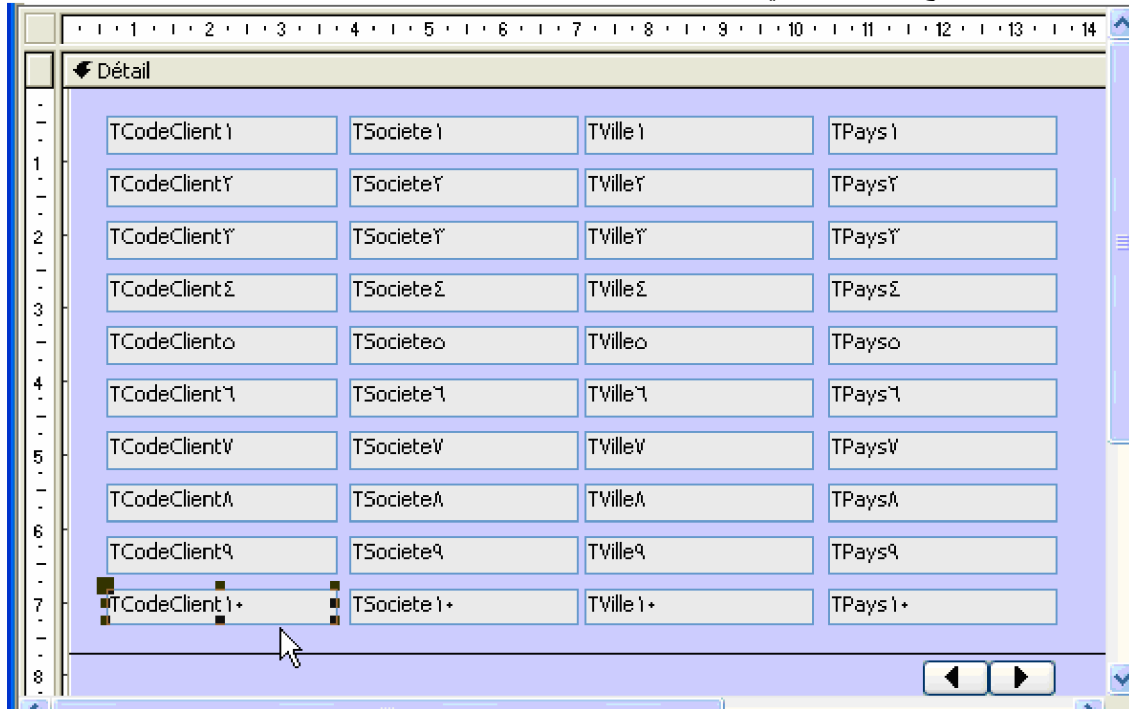
```

Dim oDb as DAO.Database
Dim oRstFiltre as DAO.Recordset
' يهيء قاعدة البيانات
Set oDb=CurrentDb
' فتح مجموعة صفوف curseur تتضمن عدد العملاء ساكني الدار البيضاء
    
```

```
Set oRstFiltre=oDb.OpenRecordset("SELECT * FROM TblClients WHERE  
Ville=" & Chr(34) & "الدار البيضاء" & Chr(34)",dbOpenDynaset)
```

```
*****  
' هنا معالجة البيانات المصفاة  
*****  
  
' مقرر الكائنات  
oRstFiltre.Close  
oDb.Close  
Set oRstFiltre=Nothing  
Set oDb=Nothing
```


أما النموذج فيتكون مما يلي :



كل مربع نص هو بالمرّة غير منضم ، ويحمل اسما يوافق اسم العمود الذي ينتمي إليه مضاف إليه رقم يمثل فهرس الصف في المصفوفة .

نستعمل متغيرتين قابلتين للوصول على مستوى وحدة النموذج :

```
Dim oRstClient As DAO.Recordset ' يجزن مجموعة الصفوف curseur
Dim intnbLus As Integer ' يجزن عدد السجلات المقروءة آخر مرة
```

المبدأ الذي يرتكز إليه أسلوب القراءة هذا بسيط للغاية ؛ عند فتح النموذج نهيء كائن السجلات ، ويظهر النموذج السجلات العشر الأولى .

```
Private Sub Form_Load()
Dim oDb As DAO.Database
' يهيء قاعدة البيانات
Set oDb = CurrentDb
' يفتح كائن السجلات
Set oRstClient = oDb.OpenRecordset("Clients", dbOpenTable)
' يقرأ العشر سجلات الأولى
LectureVersLAvant
End Sub
Private Sub LectureVersLAvant()
On Error GoTo err
RemplirZoneTexte oRstClient.GetRows(10)
Exit Sub
err:
' إن وصلنا آخر سجل لا يفعل شيء وإلا يصدر تنبيهه
If err.Number <> 3021 Then
MsgBox "خطأ وقع أثناء القراءة", vbCritical, "تنبيهه"
End If
End Sub
```

الدالة **RemplirZonetexte** تنفذ دارة تكرارية لكل سطر من المصفوفة ، وتظهر القيم الموافقة لكل حقل في مربع النص الموافق له على السطر المحدد بالنموذج وتخفي كل السطور الموالية .

```

Sub RemplirZoneTexte(Tableau As Variant)
Dim I As Integer
' يسترجع عدد السجلات المقروءة
intnbLus = UBound(Tableau, 2) + 1
' يرسل القيم إلى مربعات النص الموافقة
For I = 0 To intnbLus - 1
' يظهر رمز العميل
Controls("TCodeClient" & I + 1) = Tableau(0, I)
Controls("TCodeClient" & I + 1).Visible = True
' يظهر الشركة
Controls("TSociete" & I + 1) = Tableau(1, I)
Controls("TSociete" & I + 1).Visible = True
' يظهر المدينة
Controls("TVille" & I + 1) = Tableau(2, I)
Controls("TVille" & I + 1).Visible = True
' يظهر الدولة
Controls("TPays" & I + 1) = Tableau(3, I)
Controls("TPays" & I + 1).Visible = True
Next I
' يخفي بقية مربعات النص
For I = intnbLus + 1 To 10
' يخفي رمز العميل
Controls("TCodeClient" & I).Visible = False
' يخفي الشركة
Controls("TSociete" & I).Visible = False
' يخفي المدينة
Controls("TVille" & I).Visible = False
' يخفي الدولة
Controls("TPays" & I).Visible = False
Next I
End Sub
    
```

التنقلات نحو الخلف ، تكون أكثر تعقيدا ، فإذا كان قد قرئ 20 سجلا من بداية كائن السجلات ، فإن الموقع الحالي يحدد برقم الصف 21 ، والصفوف 11 إلى 20 معروضة بالنموذج ، والحالة هذه ، إذا رغبتنا بالتنقل نحو الخلف ، والكتلة المرغوب بها هي السجلات من 1 إلى 10 ، إذن يتوجب علينا التراجع بعدد السجلات المقروءة آخر مرة ، ثم نتوقع قبل 10 سجلات إلى الخلف لبدء القراءة ؛ غير أنه يمكن أن يحدث ؛ أن عدد السجلات المرغوب التراجع بها يفوق عدد لمتوفر إلى الخلف ، وأسهل طريقة معالجة الخطأ (3021) حتى لا يتوقف البرنامج :

```

Private Sub LectureVersLArriere()
On Error GoTo err
' التراجع بعدد السجلات المطلوب
oRstClient.Move -1 * intnbLus - 10
RemplirZoneTexte oRstClient.GetRows(10)
Exit Sub
err:
Select Case err.Number
' إذا تراجعنا أكثر من اللازم ، إذن يتموقع على أول سجل
Case 3021: oRstClient.MoveFirst
' إن كان غير ذلك ينبه
Case Else: MsgBox "خطأ غير معروف ، قد حدث أثناء القراءة " & vbCrLf, "تنبيه"
End Select
End Sub
    
```

يمكن إذن الحصول على واجهة تشبه هذه :



ID	Name	Location	Country
ALFKI	Alfreds Futterkiste	Berlin	Allemagne
ANATR	Emparedados y helados	México D.F.	Mexique
ANTON	Antonio Moreno Taquería	México D.F.	Mexique
AROUT	Around the Horn	London	Royaume-Uni
BERGS	Berglunds snabbköp	Luleå	Suède
BLAUS	Blauer See Delikatessen	Mannheim	Allemagne
BLONP	Blondel père et fils	Strasbourg	France
BOLID	Boldo Comidas preparadas	Madrid	Espagne
BONAP	Bon app'	Marseille	France
BOTTM	Bottom-Dollar Markets	Tsawassen	Canada

- 5.6 قائمة الخصائص

هذا الجزء سيستعرض كل الخصائص التي تطبق على كائن السجلات في النموذج DAO

- 5.6.1 الخاصية AbsolutePosition

Type : Long

الخاصية AbsolutePosition توافق رقما يمثل السجل الحالي بكائن السجلات . وأول سجل يمثلته الرقم : 0 ، وبذاء عليه يمكن أن نرسل لهذه الخاصية رقما بين الأصف و عدد السجلات -1 ، وسديتم الانتقال للوصول إلى السجل المحدد . غير أنه من المحبذ استعمال الخاصية Bookmark لحفظ موقع ، ثم العودة إليه . وفي حال كون كائن السجلات خاليا ، فإن هذه الخاصية سترجع القيمة -1 .
لا تطبق على كائن السجلات من نوع جدول (dbOpenTable) ، ولا من نوع تقديمي فقط (dbOpenForwardOnly)

- 5.6.2 الخاصية BOF

للقراءة فقط


Type : Boolean

إذا كانت الخاصية BOF لكائن سجلات تساوي True ، فهذا يعني أن الموقع الحالي يوجد قبل أول سجل هذه الخاصية تطبق على جميع أنواع كائن السجلات .

- 5.6.3 الخاصية Bookmark

Type : Variant

والخاصية Bookmark تناظر علامة (مؤشرا) تعين بكيفية فريدة سجلا ضمن كائن السجلات ، هذه الخاصية تستعمل لحفظ موقع أو للتنقل ، وهناك مثال عن هذه الخاصية في الفقرة : 5.3.3 حفظ موقع . هذه الخاصية لا تطبق على كائن السجلات من نوع تقديمي فقط (dbOpenForwardOnly)

العلامات (المؤشرات) لا يمكن استعمالها إن كانت الخاصية Bookmarkable تساوي	False	
--	-------	---

- 5.6.4 الخاصية Bookmarkable

للقراءة فقط

Type : Variant

الخاصية Bookmarkable ترجع True إذا كانت مجموعة الصفوف curseur تقبل هذه العلامات (وهي حالة تلقائية لقاعدة بيانات أكسيس) ، وإن كنت تشتغل باتصال مع قواعد بيانات أخرى فإنه من الضروري اختبار هذه الخاصية قبل استعمال الخاصية Bookmark

هذه الخاصية لا تطبق على كائن السجلات من نوع تقديمي فقط (dbOpenForwardOnly)

5.6.5 - الخاصية CacheSize


Type : Long

الخاصية CacheSize تتعلق بعدد السجلات المخزنة بالذاكرة البينية . وقيمتها تتراوح بين 5 و 1200 من غير أن تتجاوز مقدار الذاكرة المتوفر ، والقيمة 0 توقف استغلال الذاكرة البينية . هذه الخاصية تطبق فقط مع كائن السجلات من الأنواع الحيوية dynamique أو اللحظي ، وأيضا عندما يستقبل بيانات من مصدر ODBC.

5.6.6 - الخاصية CacheStart

Type : String

قيمة الخاصية CacheStart توافق علامة (مؤشرا) على أول سجل في الكائن Recordset وضع في الذاكرة البينية . ويمكن استعمال العلامة لكل السجلات ، لتحديد الخاصية CacheStart هذه الخاصية تطبق فقط مع كائن السجلات من الأنواع الحيوية dynamique أو اللحظي ، وأيضا عندما يستقبل بيانات من مصدر ODBC.

هذه الخاصية قد فصل القول فيها بشكل مستفيض في هف المساعدة للأكسيس ، ويمكن الرجوع إليه للتوسع . 

5.6.7 - الخاصية Connection

للقراءة فقط

Type : Connection

في مجال عمل ODBC مباشر ، فإن الخاصية Connection ترجع كائن الاتصال الذي من خلاله تم فتح كائن السجلات . هذه الخاصية غير متوفرة بالنسبة لكائن السجلات من نوع جدول .

5.6.8 - الخاصية DataCreated

للقراءة فقط

Type : Date

لكائن سجلات من نوع جدول هذه الخاصية ترجع تاريخ إنشاء الجدول

5.6.9 - الخاصية EditMode

للقراءة فقط

Type : Long

الخاصية EditMode ترجع قيمة تشير إلى حالة التعديل بالنسبة للسجل الحالي ، والقيم الممكنة هي عبارة عن ثابتة من الثوابت التالية :

الثابتة	دلالتها
dbEditNone	لا إجراء تعديل على السجل الحالي
DbEditInProgress	لقد أثيرت الخاصية Edit، والسجل الحالي الموجود في الذاكرة البينية للنسخ هو جاهز للتحديث (الخاصية Update لم تستدعى بعد)
dbEditAdd	أثيرت الخاصية AddNew، والسجل الحالي الموجود في الذاكرة البينية للنسخ هو سجل جديد (لم يسجل بعد بقاعدة البيانات)

هذه الخاصية تطبق على جميع أنواع كائنات السجلات .

وبهذا في سنن (كود) معقد ، من الممكن معرفة حالة كائن السجلات ، وهكذا لمعرفة هل نحن على وشك التصديق على إضافة سجل ، يمكن أن نختبر القيمة : EditMode

```
If oRecordset.EditMode=dbEditAdd then MsgBox " السجل الحالي سجل جديد لم يسجل بعد"
```

ومع ذلك ، إذا كان الكود مهيكلاً بشكل جيد للقراءة ، يمكنك معرفة حالة كائن السجلات عند كل سطر من الكود ، واستعمال هذه الخاصية يجب ألا يتم إلا في الحالات القصوى ، أو حين يراد تتبع الشوائب بالكود المكتوب .



- 5.6.10 الخاصية EOF

للقراءة فقط

Type : Boolean

عندما تساوي الخاصية EOF لكائن سجلات القيمة True ، فهذا يعني أن الموقع الحالي هو موقع بعد آخر سجل في الكائن . هذه الخاصية تطبق على جميع أنواع كائنات السجلات .

- 5.6.11 الخاصية Fields

للقراءة فقط

Type : Collection

هذه مجموعة تضم جميع الحقول لكائن سجلات ، ولمعرفة موسعة حول الكائن Field ، أحيك على الفقرة : 4.4 الكائن Field هذه الخاصية تطبق على جميع أنواع كائنات السجلات .

- 5.6.12 الخاصية Filter

Type : String

الخاصية Filter تسمح بتخصيص مصفاة يمكن تطبيقها عند استدعاء الطريقة OpenRecordset . يتعلق الأمر إذن بالاشتراط WHERE لاستعلام SQL (من غير ذكر الكلمة المحجوزة WHERE) ، ولمعرفة أوسع بهذا الكائن أحيك على الفقرة : 5.4.6 تصفية البيانات .

- 5.6.13 الخاصية Index

Type : String
الخاصية Index تتعلق باسم الفهرس المستعمل في كائن سجلات ، للبحث عن سجلات في هذا الكائن على أن يكون من نوع جدول دون سواه .

- 5.6.14 الخاصية LastModified

للقراءة فقط
Type : Variant
هذه الخاصية ترجع علامة (مؤشرا) نحو آخر سجل تم تعديله بكائن السجلات ، ويمكن بعد ذلك الوصول إلى هذا السجل بإرسال هذه العلامة إلى الخاصية Bookmark

مثال :

```
With oRecordset
.Bookmark=.LastModified
End With
```

هذه الخاصية تطبق مع كل كائنات السجلات التي من نوع تقديمي فقط (dbOpenForwardOnly)

- 5.6.15 الخاصية LastUpdated

للقراءة فقط
Type : Date
هذه الخاصية ترجع تاريخ آخر تحديث لجدول كائن سجلات .
لا تطبق إلا مع كائنات السجلات من نوع جدول فقط

- 5.6.16 الخاصية LockEdits

Type : Boolean
هذه الخاصية تحدد أسلوب إغلاق السجلات المستعمل من كائن سجلات Recordset . فإذا كانت قيمة هذه الخاصية تساوي True (وهي القيمة الافتراضية لها في أكسيس) ، فإن الصفحة الحالية تغلق مباشرة عند استدعاء الطريقة Edit ، وفي الحالة الأخرى عندما تساوي القيمة False ، فإن الصفحة الحالية تغلق فقط حين استدعاء الطريقة Update

- 5.6.17 الخاصية Name

Type : String
الخاصية Name تمثل اسم الكائن . وهذه الخاصية غير مفيدة إلى حد ما عند استعمال كائن السجلات ، حيث تأخذ اسما لها الـ256 حرفا الأولى من معلمة المصدر المستعملة من الطريقة :
Database.OpenRecordset

- 5.6.18 الخاصية NoMatch

للقراءة فقط

Type : Boolean

الخاصية NoMatch تساوي قيمتها True عندما واحدة من طرائق البحث Find.... لا تصل إلى أي سجل يحقق اشتراط البحث .

- 5.6.19 الخاصية PercentPosition

Type : Single

الخاصية PercentPosition تتراوح قيمتها بين 0 و 100 وتشير إلى موقع تقريبي داخل كائن السجلات ، ويمكن استعماله للوصول إلى موقع سجل ، غير أنه ليس ذا أهمية لأن قيمته تقريبية . لا تطبق إلا في كائنات السجلات من نوع جدول ، وورقة بيانات حيوية ، ولحظي

يفترض عند القيمة 100 الوصول إلى آخر سجل لتكون قيمة هذه الخاصية ذات معني في علاقة مع كامل كائن السجلات ، وإلا فإن هذه الخاصية لا تركز إلى العدد الإجمالي للسجلات ، ويتوجب الوصول إلى آخر سجل باستعمال الطريقة MoveLast.



- 5.6.20 الخاصية Properties

للقراءة فقط

Type : Collection

هذه المجموعة تضم كل الخصائص المتوفرة لكائن سجلات كما سبق تفصيله في الفقرة : 4.2 الخصائص النوعية

- 5.6.21 الخاصية RecordCount

للقراءة فقط

Type : Long

الخاصية RecordCount ترجع رقما يساوي عدد السجلات في كائن سجلات من نوع جدول ، أو عدد السجلات التي تم الوصول إليها في كائن سجلات من أي نوع آخر . لا تطبق على كائن السجلات من نوع تقديمي فقط (dbOpenForwardOnly)

لمعلومات أكثر حول هذه الخاصية ، يمكن التوجه إلى الفقرة : 5.4.5 تعداد السجلات



5.6.22 - الخاصية Restartable

للقراءة فقط

Type : Long

عندما تساوي قيمة هذه الخاصية True ، فهذا يعني أنه من الممكن تحديث البيانات ، بإعادة تنفيذ الاستعلام بمساعدة الطريقة Requery . وبعبارة أخرى فهي تبين إن كانت الطريقة Requery مدعومة من الكائن سجلات ، وفي الحالة الأخرى عندما تساوي القيمة False فإنك تضطر لفتح مجموعة الصفوف curseur بمساعدة الطريقة OpenRecordset للكائن Database أو الكائن QueryDef المستعمل لاستمداد البيانات . يستعمل مع كل أنواع كائنات السجلات ، غير أنه يرجع القيمة False مع كائن السجلات من نوع جدول .

5.6.23 - الخاصية Sort

Type : String

الخاصية Sort تسمح بتخصيص أمر الفرز الذي سيطبق عند استدعاء الطريقة OpenRecordset ، إنه يماثل الاشتراط ORDER BY لاستعلام SQL (من غير ذكر هذا الاشتراط) وهكذا من الممكن إنشاء نسخة كائن جديد للسجلات مفروز انطلاقا من كائن سجلات آخر .

```
Dim odb as DAO.Database
Dim orst as DAO.Recordset
Dim orstTrie as DAO.Recordset
' إنشاء نسخة من قاعدة البيانات
Set odb=CurrentDb
' فتح مجموعة الصفوف curseur ترجع كافة العملاء من الجدول Clients
Set orst=odb.OpenRecordset("SELECT Ville,NomClient,PrenomClient " _
& "FROM TblClients" ,dbOpenDynaset)
' تفرز المدينة واسم العميل
orst.Sort="Ville,NomClient"
' إعادة فتح مجموعة الصفوف curseur
Set orstTrie=orst.OpenRecordset
' *****
' هنا معالجة البيانات المفروزة
' *****

' محرر الكائنات
orst.Close
orstTrie.Close
odb.Close
Set orst=Nothing
Set orstTrie=Nothing
Set odb=Nothing
```

هذه الخاصية لا تطبق إلا مع كائن السجلات من نوع ورقة بيانات حيوية ، وألحظي . نسجل مع ذلك ، أنه في غالبية الأحوال يكون أسرع فتح كائن سجلات مباشرة من قاعدة البيانات Database ، مخصص باستعلام SQL كامل في المعلمة source .

5.6.24 - الخاصية *StillExecuting*

للقراءة فقط

Type : Boolean

في مجال عمل ODBC مباشر، الخاصية *StillExecuting* ترجع القيمة True إذا كان استعلام لفتح كائن السجلات ما يزال في طور التنفيذ .

5.6.25 - الخاصية *Transactions*

للقراءة فقط

Type : Boolean

عندما تساوي الخاصية *Transactions* لكائن السجلات القيمة True، فهذا يعني أنه يمكنك استعمال الجلسات مع كائن السجلات هذا ، ولانطلاق جلسة ، استعمل الطريقة *BeginTrans* للكائن ، لتنشئ مجموعة الصفوف *curseur* ، ثم بعد ذلك توسل بالطريقة *CommitTrans* لتحديث البيانات ، أو *Rollback* للإلغاء

مثال :

```
Dim oRst as DAO.Recordset
Dim oDb as DAO.Database
Set oDb=CurrentDb
Set oRst=oDb.OpenRecordset ("TblCommande",dbOpenTable)
' انطلاق الجلسة
DBEngine.Workspaces(0).BeginTrans
While Not oRst.EOF
' انتقل إلى وضع الكتبي
oRst.Edit
' يرسل التاريخ الحالي
oRst.Fields("DateCommande").Value=Now()
' تحديث
oRst.Update
' ننتقل للموالي
oRst.MoveNext
Wend
' يطلب التصديق
If MsgBox("Êtes vous sûr de vouloir continuer")=VbYes
' إن الجواب بالإيجاب ، يتم التحديث
DBEngine.Workspaces(0).CommitTrans
Else
' و إلا يتم إلغء الإجراء
DBEngine.Workspaces(0).Rollback
End if

' يمرر الكائنات
oRst.Close
oDb.Close
Set oRst=Nothing
Set oDb=Nothing
```

- 5.6.26 الخاصية Type

للقراءة فقط

Type : Long

الخاصية Type ترجع نوع مجموعة الصفوف cursor المستعمل طبقا لمعلومات التخصيص المرسله ، راجع الفقرة : - 5.1 م أنواع كائنات السجلات

- 5.6.27 الخاصية Updatable

للقراءة فقط

Type : Boolean

عندما تكون قيمة الخاصية Updatable تساوي False فإنه من المستحيل تحديث بيانات كائن السجلات .

- 5.6.28 الخاصية UpdateOptions

Type : Long

الخاصية UpdateOptions توافق أسلوب إنشاء اشتراط WHERE باستعلام تحديث ، حين يتم التحديث بالكتل في مجال عمل ODBC مباشر . وفي بعض الحالات ، إنه كذلك ممكن الالتفاف ، بتفجير تحديث لقاعدة بيانات عن بعد ، وذلك بإجبار محرك قاعدة البيانات على تعويض استعلام التحديث باستعلام حذف متبوع باستعلام إلحاق ، غير أن هذه الخاصية لن يتوسع فيها هنا ، وأحيلك على ملفات التعليمات لميكروسوفت أكسيس.

- 5.6.29 الخاصية ValidationRule

للقراءة فقط

Type : String

الخاصية ValidationRule تحدد قاعدة التصديق التي ستختبر عند استدعاء الطريقة Update

- 5.6.30 الخاصية ValidationText

للقراءة فقط

Type : String

الخاصية ValidationText توافق الرسالة التي سيتم إعلانها عند التحديث إذا كانت البيانات لا تحقق قاعدة التصديق المخصصة في الخاصية ValidationRule

5.7 - قائمة الطرائق

وهذا ملخص الطرائق المتوفرة لكائن السجلات

5.7.1 - الطريقة AddNew

الطريقة AddNew تسمح بإضافة سجل جديد إلى كائن السجلات ، ستجد شرحا وافيا في الفقرة :
5.4.3 إضافة سجل

مثال :

```
With oRecordset
.AddNew
.Fields("Nom")="Toto"
.Update
End With
```

5.7.2 - الخاصية CancelUpdate

الطريقة CancelUpdate تثار بعد الطريقة Edit أو الطريقة Addnew من أجل إلغاء التحديث للسجل ، وفي حالة الإضافة ، فإن موقع السجل الحالي بعد تطبيق هذه الطريقة ، هو موقع سجل قبل استدعاء الطريقة AddNew

مثال :

```
With oRecordset
.AddNew
.Fields("Nom")="Toto"
'Annule l'insertion
.CancelUpdate
End With
```

5.7.3 - الطريقة Clone

الطريقة ترجع كائن سجلات مطابق تماما للكائن الذي طبقت عليه هذه الطريقة ، وكائن السجلات المستنسخ لا يتضمن أي سجل حالي ، ويتوجب إذن أن تتموقع على سجل من اختيارك ، ولتسجل فقط أن العلامات (المؤشرات) غير متطابقة بين كائني السجلات ، وهكذا فإنه من الممكن التموقع على سجل بكائن السجلات المنسوخ يطابق الموقع الحالي بالكود التالي :


```
Dim oRst as DAO.Recordset
Dim oRstClone as DAO.Recordset
'*****
' هنا الكود لفتح الكائن Rst
'*****
' يستنسخ كائن السجلات
Set oRstClone=oRst.Clone
' مجدد نفس السجل الحالي
oRstClone.Bookmark=oRst.Bookmark
```

عندما يتم استنساخ كائن السجلات oRstClone ، يمكنك إغلاق oRst من غير أن يؤثر ذلك على البيانات في oRstClone 

هذه الطريقة غير متوافقة مع كائن السجلات من نوع تقديمي فقط (dbOpenForwardOnly)

5.7.4 - الطريقة Close

الطريقة Close تغلق كائن سجلات مفتوح . غير أن هذا لا يحرر الذاكرة المخصصة لكائن السجلات الذي تم غلقه ، ولتحرير تلك الذاكرة ، عليك إما الانتظار إلى حين انتهاء الإجراء ، وتدمير متغيرة الكائن المغلق أو إرسال القيمة Nothing إلى هذه المتغيرة .

مثال :

```
oRecordset.Close
Set oRecordset=Nothing
```

5.7.5 - الطريقة CreateQueryDef

الطريقة CreateQueryDef تنسخ كائن الاستعلام QueryDef المستعمل لفتح كائن سجلات ، وهذا الأخير يجب حتما أن يكون قد فتح بمساعدة الطريقة OpenRecordset لكائن استعلام QueryDef

مثال :

```
Set querydef = oRecordset.CopyQueryDef
```

هذه الطريقة طبعا لن تطبق على كائن السجلات من نوع جدول ، ما دام لا يفتح انطلاقا من استعلام .

5.7.6 - الطريقة Delete

الطريقة تحذف سجلا من كائن السجلات ، وهي طريقة تم عرضها بالفقرة : 5.4.4 حذف سجل . عندما يتم حذف سجل فإن موقع هذا السجل غير قابل للولوج ، وتحتاج الوصول إلى موقع سجل آخر بإحدى طرائق التنقل .

مثال :

```
With oRecordset
.Delete
Msgbox "تم الحذف بنجاح"
.MoveNext
End With
```

- 5.7.7 الطريقة Edit

الطريقة Edit تتيح نسخ السجل إلى الذاكرة البينية والتنقل كائنات السجلات إلى الواجهة مع acEditInProgress لتمكينك من تعديل البيانات .
مثال تطبيقي لهذه الطريقة معروض بالفقرة : 5.4.2 تعديل سجل

- 5.7.8 الطرق FindFirst, FindLast, FindNext, FindPrevious

هذه الطرق تستعمل للوصول إلى سجل ضمن كائن السجلات ، وقد خصصت الفقرة : (5.3.2 البحث عن سجل) لعرض هذه الطريقة
هذه الطرق ليست متوفرة إلا لكائنات السجلات من نوع ورقة بيانات حيوية ، أو اللحظي

- 5.7.9 الطريقة GetRows

الطريقة GetRows تقرأ وترجع في مصفوفة ذات بعدين عددا معطى (X) من السجلات تستخلصها من كائن السجلات الحالي (على أن X عدد صحيح طبيعي موجب ، يرسل للمعلمة الوحيدة للطريقة GetRows) . مثال تطبيقي يمكن أن ترجع إليه في القسم : 5.5 القراءة بالكتل

- 5.7.10 الطرق Move, MoveFirst, MoveLast, MoveNext, MovePrevious

هذه الطرق تستعمل للتنقل والوصول إلى سجل ضمن كائن السجلات ، وقد خصصت الفقرة : (5.3.1 التنقل داخل السجلات Recordset) لعرض هذه الطريقة
هذه الطرق متوفرة لكل أنواع كائنات السجلات ، عدا الطريقتين MovePrevious و MoveFirst اللتين لا تطبقان على الكائنات من نوع تقديمي فقط (dbOpenForwardOnly) لأنها تتعارض معها .

- 5.7.11 الطريقة OpenRecordset

الطريقة تنشئ نسخة جديدة من كائن السجلات ، تمتلك مصدر البيانات نفسه لكن مع الأخذ بالاعتبار استثناءات التصفية والفرز الممكنة .
وهذا مثال لاستعمال الطريقة مع تطبيق مصفاة :

```


Dim oDb as DAO.Database
Dim oRst as DAO.Recordset
Dim oRstFiltre as DAO.Recordset
' تنشئ نسخة عن قاعدة البيانات (تهيء القاعدة)
Set oDb=CurrentDb
' فتح مجموعة صفوف CURSEUR تتضمن كل العملاء
Set oRst=oDb.OpenRecordset("SELECT * FROM TblClients",dbOpenDynaset)
' يطبق المصفاة لتصفية العملاء ساكني الدار البيضاء
oRst.Filter="Ville=" & Chr(34) & "Paris" & Chr(34)
' فتح مجموعة الصفوف CURSEUR من جديد
Set oRstFiltre=oRst.OpenRecordset
'*****
' هنا معالجة البيانات المصفاة
'*****

```

هذه الطريقة غير متوفرة بالنسبة لكائن السجلات من نوع تقديمي فقط (*dbOpenForwardOnly*)

- 5.7.12 الطريقة Requery

الطريقة Requery تحدث البيانات المضمنة في كائن سجلات بإعادة التنفيذ من جديد للاستعلام الذي استند إليه هذا الكائن في الإنشاء .

<p>لا حظ أن العديد من قو اعد البيانات لا تدعم استعمال الطريقة Requery، ولمعرفة إمكانية تجاوب قاعدة البيانات مع هذه الطريقة، اختبر الخاصية Restartable لهذه القاعدة .</p>	
--	---

- 5.7.13 الطريقة Update

الطريقة Update تسمح بتسجيل (حفظ) البيانات الموجودة في الذاكرة البينية للنسخ بقاعدة البيانات . هذه الطريقة تستدعي بعد استدعاء الطريقة Edit أو AddNew، و بعد إرسال مختلف البيانات إلى مختلف الحقول . ولإلغاء التسجيل (الحفظ) استعمل الطريقة CancelUpdate .
ولمعلومات أوسع في هذا المجال أحيك على الفقرة : 5.4.2 تعديل سجل .

القسم السادس :

- خلاصة

عبر فقرات هذا المؤلف استطعت أن تلامس النموذج DAO ، وتتعرف إلى إمكانياته المختلفة والقوية لمعالجة قواعد البيانات . إلا أن كل حدث يجب التفكير فيه بشكل جدي ، وألا يغيب عن تفكيرك في أي حال أن الترابط المنطقي و تكامل البيانات من الأمور الأساسية . وفي أي لحظة لا يمكن المغامرة بفقدان السجلات . منذ بضع سنوات كانت هناك محاولات لتعويض النموذج DAO بالنموذج (ActiveX Data Object) ، وفي هذه الأثناء ظل النموذج DAO وسيلة فعالة وبسيطة للوصول إلى قواعد البيانات عبر كود الـ VBA . إن تراتبية هذا النموذج جعلته الأقرب إلى نافذة قاعدة البيانات بالأكسيس . كما أن مقارنته للبيانات لا تتطلب معرفة معمقة في عالم البرمجة الكائنية التوجيه وهذا من الأشياء المحمودة والمشجعة .