إنشاء قواعد البيانات
وكتابة الاستعلامات بلغة SQL

بقلم:

م. محمد حمدي غانم

ملحوظة:

هذا الكتاب جزء من كتاب:

من الصفر إلى الاحتراف برمجة قواعد البيانات في سي شارب ٢٠١٠.. وسأنشر الجـــزء الثاني الخاص بتقنية ADO.NET قريبا بإذن الله، فتابعوني على صفحتي البرمجة:

https://www.facebook.com/vbandcsharp

هذا الكتاب صدقة جارية على روح والدي:

أ. حمدي كامل الحديدي غانم

رحمه الله وغفر له وجعل مثواه الجنة

لهذا أرجو من كل من يستفيد به أن يتذكر أن أبي هو الـــذي ربـــاني وعلمنـــي ولولاه بعد توفيق الله ما خرج إلى الوجود هذا الكتاب وغيره من الكتب.

فادعوا له بالرحمة والمغفرة

ومن كان منكم في الحرمين الشريفين وكان قادرا على عمل عمرة له، فجزاه الله خيرا.

أدعو الله أن يكون هذا الكتاب وباقي كتبي من العلم الذي ينتفع به، وأن يجعل الله لأبي نصيبا من ثوابه، فيكون من عمله الذي لا ينقطع بموته.

اللهم ارحم أبي واغفر له وكفر عنه سيئاته وقه من عذاب القبر وقه من عذاب النار،
وأدخله الجنة وأعلِ منزلته فيها
واحفظ والدتي وبارك في عمرها
اللهم ارحم والديّ كما ربياني صغيرا
آمين يا رب العالمين

للتواصل مع الكاتب:

- بريدي الالكتروني:

msvbnet@hotmail.com

- مدونتي:

http://mhmdhmdy.blogspot.com

- قناتي على يوتيوب (تحتوي على إلقاء أكثر من ٦٠ قصيدة بصوتي):

http://www.youtube.com/user/mhmdhmdy

- صفحتي الأدبية على فيسبوك:

https://www.facebook.com/Poet.Mhmd.Hmdy

- كتبي في مجال البرمجة بلغتي فيجوال بيزيك وسي شارب:

https://drive.google.com/drive/folders/1J21xi8Aw15BFSv-GUgVOElLuYM6zoNct

- صفحة فيجوال بيزيك وسي شارب على فيسبوك:

https://www.facebook.com/vbandcsharp

كتب مجانية للكاتب للتنزيل:

١ - كتاب: "خرافة داروين، حينما تتحول الصدفة إلى علم":

http://mhmdhmdy.blogspot.com/2013/11/blog-post_29.html

٢ - كتب أدبية (أشعار وقصص وروايات):

https://mhmdhmdy.blogspot.com/2018/10/blog-post_23.html

٣- المبرمج الصغير:

http://mhmdhmdy.blogspot.com.eg/2016/10/blog-post_9.html

٤- الرسم والتلوين والصور والمجسمات لمبرمجي فيجوال بيزيك دوت نت:

http://mhmdhmdy.blogspot.com.eg/2014/05/blog-post_26.html

٥- الرسم والتلوين والصور والمجسمات لمبرمجي سي شارب:

http://mhmdhmdy.blogspot.com.eg/2014/08/c.html

٦- من الصفر إلى الاحتراف: برمجة قواعد البيانات في فيجوال بيزيك دوت نت:

http://mhmdhmdy.blogspot.com.eg/2016/02/blog-post_28.html

كتب مطبوعة للكاتب:

١. **فيجيوال بيزيك وسي شارب:** طريقك المختصر للانتقال من إحدى اللغتين إلى الأخرى.
٢. من الصفر إلى الاحتراف: **فيجيوال بيزيك دوت نت ٢٠١٧.**
٣. من الصفر إلى الاحتراف: **سي شارب ٢٠١٧.**
٤. من الصفر إلى الاحتراف: **برمجة إطار العمل لمبرمجي فيجيوال بيزيك دوت نت وسي شارب.**
٥. من الصفر إلى الاحتراف: **برمجة نماذج الويندوز لمبرمجي فيجيوال بيزيك دوت نت وسي شارب.**
٦. المدخل العملي السريع إلى **فيجيوال بيزيك دوت نت ٢٠١٧.**
٧. المدخل العملي السريع إلى **سي شارب ٢٠١٧.**
٨. أساسيات WPF لمبرمجي **فيجيوال بيزيك دوت نت.**
٩. أساسيات WPF لمبرمجي **سي شارب.**

لقراءة مقدمة وفهرس كل كتاب:

https://drive.google.com/drive/folders/1J21xi8Aw15BFSv-GUgVOElLuYM6zoNct

لشراء هذه الكتب، يتم تحويل الثمن بحوالة بريدية داخل مصر، أو بويسترن يونيون من خارج مصر، ويتم إرسال الكتب بطرد بالبريد السريع.. لمزيد من التفاصيل أرسل رسالة بالكتب المطلوبة إلى:

msvbnet@hotmail.com

كتب أجهز لكتابتها في المرحلة القادمة بإذن الله:

- برمجة قواعد البيانات بـــ Entity Framework.
- إنشاء تقارير Report Viewer و Crystal Reports وعرضها وطباعتها.
- برمجة مواقع الويب بـــ ASP.NET MVC Core.
- المواضيع المتقدمة في برمجة إطار العمل.
- الوسائط المتعددة في WPF.
- برمجة مشاريع Windows Universal Applications.
- برمجة الاندرويد بـــ Xamarin.
- برمجة الشبكات بدوت نت.

سجلوا إعجابكم بصفحتي البرمجية لمتابعة صـــدور هـــذه الكتـــب بـــإذن الله، والاســـتفادة بالملاحظات البرمجية العملية التي أنشرها على الصفحة:

https://www.facebook.com/vbandcsharp

للتواصل مع باقي المبرمجين وتبادل الأسئلة والخبرات، يمكنكم الانضمام إلى هذه المجموعة:

https://www.facebook.com/groups/123809374886424/

محتويات الكتاب

- مقدمة ٩

-١-

تركيب قواعد البيانات Database Structure

قاعدة البيانات ١٠

الجداول Tables

قواعد البيانات المترابطة Relational Databases

العلاقات Relations

القيود Constraints

العروض Views

الفهارس Indices

-٢-

قواعد بيانات Access

إنشاء الجداول بتطبيق Access ٢٨

إنشاء علاقات بين الجداول

إنشاء قيد Constraint

ضغط قاعدة البيانات Database Compacting

متصفح الخوادم Server Explorer

الاتصال بقواعد بيانات Access

-٣-

قواعد بيانات SQL Server

إعداد Management Studio Express ٤٥

إنشاء قاعدة بيانات سيكويل سيرفر

أنواع البيانات في سيكويل سيرفر

إنشاء العلاقات في سيكويل سيرفر

التحكم في خادم سيكويل

توصيل وفصل قاعدة بيانات

إنشاء قاعدة بيانات سيكويل سيرفر في دوت نت

الاتصال بقواعد بيانات SQL Server

إنشاء الفهارس Indices

إنشاء القيود Constraints

-٤-

لغة الاستعلام المركّبة Structured Query Language (SQL)

باني الاستعلام Query Builder ٨٧

استعلامات التحديد Selection Queries

جملة التحديد SELECT Statement

دوال التجميع Aggregate functions

عمليات الربط SQL Joins

استعلامات الأداء Action Queries

حذف الصفوف باستخدام الأمر DELETE

إدراج سجلات جديدة باستخدام الأمر INSERT

تحرير السجلات الموجودة باستخدام الأمر UPDATE

التحرير المتتابع للسجلات، باستخدام الأمر UPDATE .WRITE

لغة تعريف البيانات Data Definition Language (DDL)

الإجراءات المخزّنة Stored Procedures

إنشاء الإجراءات المخزنة في قواعد بيانات Access

دوال SQL التي يعرفها المستخدم User Defined SQL Functions

أنواع الجداول التي يعرفها المستخدم User-Defined Table Types

مقدمة

بسم الله، والحمد لله، والصلاة والسلام على رسول الله، وبعد:

تعتبر قواعد البيانات القلب المركزي في معظم أنواع المشاريع، سواء كانت تعمل على سطح المكتب Desktop أو موقع ويب Web Site أو شبكة Network أو جهاز يدوي محمول.. لهذا يعتبر تعلم كيفية إنشاء قواعد البيانات والتعامل معها أمرا لا غنى عنه لتنفيذ المشاريع التجارية والإدارية والعملية، التي تتيح للمستخدم استرجاع البيانات وتعديلها وحفظها والبحث فيها بمختلف الطرق.

ويبدأ هذا الكتاب الرحلة معك من الصفر، ليعرفك على المفاهيم الأساسية اللازمة للتعامل مع قواعد البيانات، وكيف تنشئها في Access و SQL Server 2008 وكذلك في بيئة فيجوال ستديو دوت نت VS.NET، وكيف تكتب الاستعلامات والإجراءات المخزنة باستخدام لغة SQL لتحصل على البيانات من الجداول أو تضيفها أو تحذفها.

والله ولي التوفيق

لتحميل أمثلة الكتاب:

http://www.mediafire.com/file/v03vhpw28r62bj0/C_Sharp_DB_Samples.zip/file

– ١ –

تركيب قواعد البيانات
Database Structure

سنتعرف في هذا الفصل على أهم التعريفات والمفاهيم الأساسية التي تلزمنا للتعامل مع قواعد البيانات في باقي فصول هذا الكتاب.

قاعدة البيانات:

قاعدة البيانات Database هي ملف أو مجموعة من الملفات، تستخدم لتخزين البياناتِ التي تربطها علاقات معيّنة، مثل الجداول.. وتتميز قاعدة البيانات عن الملفات العادية، بوجود برامج معينة للتعامل معها، مما يجعل حفظ البيانات وقراءتها والبحث فيها أمورا أكثر سهولةً وتنظيمًا، وخالية من الأخطاء.. هذه البرامج تسمّى "أنظمة إدارة قواعد البيانات":
Database Management Systems (DBMS)

وهناك أنواع كثيرة من قواعد البيانات، تبعا للشركة التي تنتجها والبرامج التي تنشئها، وطريقة تنسيق البيانات المحفوظة في ملف قاعدة البيانات.. فهناك ــ على سبيل المثال ــ قواعد بيانات Access وقواعد بيانات SQL Server وهما من إنتاج ميكروسوفت، وهناك قواعد بيانات Oracle و MySQL، وغيرها من برامج قواعد البيانات من إنتاج شركات أخرى.

ومهما كانَ نوع قاعدة البيانات التي تتعامل معها، ومهما كانت طريقة تخزينها في الملفّ، فإنَّ كل قواعد البيانات تتبع معايير أساسيّة وتحقق شروطا معيّنة متعارفا عليها دوليّا، كما أنّها كلها بلا استثناء تستخدم "لغة الاستعلام المركّبة" Structured Query Language (SQL)، وهي لغة خاصّة تستخدم لحفظ واسترجاع وتحديث البيانات الموجودة في قواعد البيانات.

وتحتوي قاعدة البيانات على أنواع مختلفة من العناصر، منها:

- الجداول Tables.
- العلاقات Relations.
- القيود Constraints.
- العروض Views.

الجداول Tables:

تتركّب قاعدة البيانات في الأساس من مجموعةٍ من الجداول، هي التي يتم تخزين البيانات بها.. فإذا لم تحتوِ قاعدة البيانات على أي جداول، كانت فارغة بلا قيمة.

فمثلا، لو لديك قاعدة بيانات خاصة بمكتبة، فسيوجد بها جدول يحتوى على عناوين الكتب الموجودة في المكتبة (وليكن اسمه Books)، وجدول آخر يحتوى على أسماءِ مؤلفي هذه الكتب (وليكن اسمه Authors)، وجدول ثالث يحتوى على أسماء دور النشر التي نشرت هذه الكتب (وليكن اسمه Publishers).

ويتكون كل جدول من:

أ. **أعمدة Columns (وتسمّى حقولا Fields أيضا):**

ويشبه العمود المتغيرات في لغات البرمجة، فله اسم ونوع.. وتختلف أسماء أنواع البيانات اختلافا طفيفا من قاعدة بيانات إلى أخرى، لكنها في النهاية تشبه أنواع البيانات الموجودة في لغات البرمجة، مثل الأعداد الصحيحة Integers، والأعداد الطويلة Long وغيرها.

وعند إنشاء الجدول، يجب تصميم الأعمدة بشكل صحيح لمراعاة التالي:

١- أن يكون اسم العمود معبرا عن البيانات التي سيتم تخزينها فيه.

٢- أن يتم اختيار نوع البيانات المناسب لحفظ بيانات هذا العمود، بحيث:

- لا يكون كبيرا جدا فيهدر مساحة كبيرة على القرص الصلب ويبطئ من عمليات نقل البيانات عبر الشبكة.

- ولا يكون صغيرا جدا فيعجز عن استيعاب بعض البيانات الهامة، مما يسبب أخطاء في البرامج التي تحاول الكتابة في الجدول، أو يمنعها من كتابة بعض البيانات الهامة إن كانت هذه البرامج مصممة جيدا لمعالجة الأخطاء التي تحدث.

ب. صفوف Rows (وتسمّى سجلات Records أيضا):

يحتوي الجدول على صفوف، كل صف منها يعتبر وحدة بيانات مستقلة، تحفظ القيم فيها تحت كل عمود من أعمدة الجدول.. فمثلا: لو لديك جدول خاص بتلاميذ إحدى المدارس، فسيحتوي هذا الجدول على أعمدة تمثل رقم الطالب واسمه وعمره وفصله، وسيكون لكل تلميذ صف (سجل) في هذا الجدول يحتوي على المعلومات الخاصة به، كما هو موضح:

ID	Name	Age	Class
١	أحمد علي	١٢	٦
٢	محمود رشوان	١١	٥
٣	شريف عمران	٦	١

ويمكن أن يحتوي الجدول على مئات أو آلاف الصفوف تبعا لاحتياجك.. وتتحكم العوامل التالية في الحد الأقصى من الصفوف المسموح بحفظه في جداول قاعدة البيانات:

١- مساحة التخزين المتوفرة على القرص الصلب الذي توجد عليه قاعدة البيانات.. وإن كانت بعض أنواع قواعد البيانات تسمح بتقسيم ملفاتها على أكثر من قرص صلب، للسماح بحفظ ملايين السجلات.

٢- نوع قاعدة البيانات.. حيث إن بعض أنواع قواعد البيانات مثل Access مصممة للتعامل مع قواعد بيانات صغيرة نسبيا، لهذا ينصح بألا تزيد عدد السجلات المحفوظة في قاعدة البيانات الخاصة بها عن ١٠٠ ألف سجل، أو

مليون سجل على أقصى تقدير!.. أعرف أن هذا يبدو رقما ضخما بالنسبة لك، ولكن في قواعد البيانات العملاقة هذا رقم في غاية التفاهة، وقواعد بيانات SQL Server و Oracle تتعامل مع سجلات تتجاوز عشرات وربما مئات الملايين!

٣- نوع العمليات التي تجريها على الجدول.. فلو كنت تحتاج إلى إجراء عمليات كثيرة على الجداول، وكان هناك عدد كبير من المستخدمين الذين يتعاملون مع هذه الجداول في نفس اللحظة، فإن ضخامة أحجام الجداول تؤدي إلى إبطاء نقل البيانات وإبطاء عمليات الإضافة والحذف والبحث.. في مثل هذه الحالة ينصح بتقسيم الجدول إلى جداول أصغر فيما يعرف باسم عملية التطبيع Normalization.. هذه العملية يمكن أن تتم:

أ. رأسيا: وذلك بتقسيم الأعمدة على أكثر من جدول، مع ربطها معها باستخدام رقم السجل ID والذي يعرف في هذه الحالة باسم المفتاح Key.

ب. أفقيا: وذلك بتقسيم الصفوف على أكثر من جدول تبعا لعامل مشترك بينها.. مثلا: يمكن تقسيم الصفوف على حسب تواريخها، وبدلا من أن يكون لديك جدول واحد لكل السنوات، يكون لديك جدول لكل سنة (هذا مناسب لجدول التلاميذ على سبيل المثال).. كما يمكنك أن تقوم بعملية تطبيع لقاعدة البيانات كلها، بتقسيمها إلى عدد من قواعد البيانات، تبعا للعام، أو المنطقة الجغرافية إلخ.

ويتحكم نوع المشروع الذي تتعامل معه وطبيعة بياناته، في كيفية حل مشاكل تضخم قواعد البيانات، وكيفية تقسيمها.. لهذا من المهم أن تحلل النظام الذي تتعامل معه جيدا لاختيار أكفأ تصميم لقاعدة البيانات منذ البداية، لأن محاولة حل مشكلة ظهرت بعد عدة سنوات من العمل على النظام، قد تكون مكلفة جدا ومضيعة للوقت والجهد، وربما تؤدي إلى إعادة كتابة البرامج كلها منذ البداية!

قواعد البيانات المترابطة Relational Databases:

لو كانت لدينا قاعدة بيانات للمكتبات ودور النشر، فسيحتوي جدول الكتب Books على عدّة أعمدة لتخزين بعض المعلومات عن كل كتاب، مثل اسمه، وعدد صفحاته، وموضوعه وسعره... إلخ.

يجب أيضا أن يحتوي جدول الكتب على معلومات عن مؤلف كل كتاب، مثل اسمه ونبذة عنه وبريده الإلكتروني وهاتفه، ليستفيد بها الناشر عندما يريد الاتصال به.. فكيف يا ترى نحفظ هذه المعلومات؟

بداهة، ليس من الذكي وضع كل هذه المعلومات في جدول الكتب، لأنه سيحتوى بالتأكيد على أكثر من كتابٍ لنفس المؤلف، مما سيسبب المشاكل التالية:

- تكرار بيانات المؤلف أكثر من مرّة، مما يزيد من حجم قاعدة البيانات ويبطئ عمليّات البحث.
- إرهاق المستخدم الذي يدخل البيانات، بسبب تكرار كتابة بيانات نفس المؤلفين مرارا وتكرارا.
- احتمال حدوث خطأ في إدخال بيانات المؤلف المكررة، مما يؤدي لتضارب في بيانات نفس المؤلف.
- عند تغير رقم هاتف المؤلف، يجب تعديل هذا في كل سجلات الكتب!
- عند البحث عن بيانات المؤلف، سنحصل على أكثر من سجل بها نفس التفاصيل بلا فائدة!

والأمر نفسه بالنسبة لبيانات دار النشر التي نشرت الكتاب، فمن المتوقع وجود مئات الكتب لكل دار نشر، ومن العبث تكرار معلومات مثل اسم وعنوان وهاتف دار النشر، واسم الناشر أو مدير الدار... إلخ مع كل كتاب!

إذن فما هو الحل المناسب لهذه المشكلة؟

كما ذكرنا من قبل، يجب إجراء عملية تطبيع Normalization لهذه البيانات، وذلك بتخصيص جدولٍ للمؤلفين Authors يحتوى على سجل واحد لكل مؤلف لنحفظ فيه بياناته الأساسية، وجدول للناشرين Publeshers يحتوي على سجل واحد فقط لكل دار نشر

لنحفظ فيه تفاصيلها، مع ربط جدول الكتب بكل من هذين الجدولين.. دعنا نرى كيف نفعل هذا:

في جدول المؤلفين سننشئ عمودا للرقم المسلسل لكل مؤلف، اسمه ID.. وفي جدول الكتب سننشئ عمودا اسمه AuthorID، لنضع فيه رقم مؤلف كل كتاب.. فمثلا لو كان رقم (توفيق الحكيم) في جدول المؤلفين هو ١، فإنّ الخانة AuthorID في جدول الكتب لا بدّ أن تكون ١ لكل من الكتب التالية:

"شهرزاد" و"الأيدي الناعمة" و"أرني الله" و"شجرة الحكم"

كما هو موضح في الجدولين التاليين:

ID	Author	Phone	eMail	About
١	توفيق الحكيم			–
٢	عباس العقاد			–
٣	محمد عبد الحليم عبد الله			–
٤	نبيل فاروق			–
٥	أحمد خالد توفيق		aktowfik@hotmail.com	–
٦	محمد حمدي		msvbnet@hotmail.com	–

ID	Book	AuthorID	Pages	Price
١	أرني الله	١	٢٠٠	٥٠٠
٢	يوميّات نائب في الأرياف	١	٣١٥	٧٠٠
٣	عصا الحكيم	١	١٥٠	٣٥٠
٤	سارة	٢	٢٧٠	٥٠٠
٥	عبقرية محمد	٢	٣٢٩	٦٠٠
٦	عبقرية عمر	٢	٢٨٠	٦٠٠
٧	شجرة اللبلاب	٣	٢٣٠	٤٠٠

٨	مهنتي القتل	٤	١٠٠	٢٠٠
٩	الأفق الأخضر	٤	١١٠	٢٠٠
١٠	أسطورة الغرباء	٥	١٣٠	٢٠٠
١٢	لا تدخلوا شيرود	٥	١٧٨	٢٥٠
١٣	مجرد طريقة للتفكير	٦	٦٠	١٠٠
١٤	حائرة في الحب	٦	١٨٠	٢٠٠
١٥	بين قوسين من الخلود	٦	٦٠	١٠٠

نفس الشيء يمكن فعله بالنسبة لجدول الناشرين، حيث سنضع فيه تفاصيل كل الناشرين، ونربط بينه وبين جدول الكتب بإنشاء عمود اسمه PublisherID في جدول الكتب.

والآن، دعنا نؤكد مرة أخرى فوائد هذه العملية:

١- توفير مساحة التخزين، فبدلا من تكرار نفس بيانات المؤلفين والناشرين مع كل كتاب، صرنا نكرر فقط رقم المؤلف ورقم الناشر.

٢- تسريع عمليّة البحث.. فمثلا لو أردت البحث عن كل كتب توفيق الحكيم في جدول الكتب، فسيكون البحث عن الرقم ١ في الحقل AuthorID (وهو حقل عددي) أسرع بكثير من البحث في الحقل Author (وهو حقل نصي) عن الاسم "توفيق الحكيم".. صحيح أننا سنبحث في جدول المؤلفين أولا عن "توفيق الحكيم" للحصول على الرقم الخاص به، لكن عدد الصفوف في جدول المؤلفين سيكون أقل بكثير من عدد الصفوف في جدول الكتب.

٣- سهولة التعديل وتصحيح الأخطاء.. افترض مثلا أنّ الموظف الذي يُدخل أسماء المؤلفين أخطأ وكتب الاسم "توفيق الحكيم" كالتالي: "تفيق الحكيم".. في هذه الحالة ما عليه إلا أن يعدّل الخطأ مرّة واحدة في جدول المؤلفين.. أمّا لو كانت أسماء المؤلفين موجودة في نفس جدول الكتب، فإنّ تكرار كتابة اسم المؤلف يجعل احتمالات الخطإ أكبر، بالإضافة إلى تضييع الوقت والجهد في كتابتها، ومشقة تعديلها كلها!

وعند استخدام هذا التنظيم، يقال إن هناك علاقةً Relationship بين جدولي الكتب والمؤلفين، حيث إنّ الرابط بينهما هو رقم المؤلف (Author.ID و Books.AuthorID).. كما توجد علاقة بين جدول الناشرين وجدول الكتب، والرابط بينهما هو رقم الناشر (Publisher.ID و Books.PublisherID).

هنا يجب أن نتعرف على المصطلحات التالية:

- يسمّى الحقل المشترك في كلا الجدولين "**حقلا مفتاحيا**" **Key field**.
- يسمّى الحقل ID في جدول المؤلفين باسم "**المفتاح الأساسيّ**" **Primary Key**، وذلك لأنه متفرّد Unique غير قابل للتكرار (يجب ألا يتكرر نفس المفتاح لأكثر من مؤلف).. ولا يحتوي الجدول على أكثر من مفتاح أساسيّ واحد.
- يسمّى الحقل AuthorID في جدول الكتب باسم "**المفتاح الفرعيّ**" أو "**المفتاح الدخيل**" **Foreign Key**، لأنّه موضوع في هذا الجدول ليربطه بجدول آخر.. ويمكن أن تتكرّر بعض القيم في خانات هذا العمود بدون أدنى مشكلة (كما في حالة وجود أكثر من كتاب لنفس المؤلف).. ويمكن أن يحتوى الجدول على أكثر من مفتاح فرعيّ تربطه بعدد من الجداول الأخرى (كأن يحتوي جدول الكتب على المفتاحين الفرعيين AuthorID و PublisherID).
- يسمّى جدول المؤلّفين باسم "**الجدول الرئيسيّ**" **Master**، وجدول أسماء الكتب باسم "**جدول التفاصيل**" **Details**، حيث يمكنك أن تحصل منه على معلومات عن الكتب المتوافرة لكل مؤلف.. وتسمى العلاقة بين الجدولين بعلاقة **Master-Details**.
- تسمّى عمليّة مطابقة الصفوف بين الجدولين المترابطين باستخدام الحقل المفتاحيّ، بعملية "**الربط**" **Join**.. كأن تحاول معرفة اسم مؤلف كتاب معيّن، أو أن تحاول أن تعرف كل الكتب التي كتبها نفس المؤلف.

أنواع العلاقات:

يوجد أربعة أنواع رئيسية للعلاقات بين الجداول:

١- **علاقة "واحد بمتعدّد" One-to-many Relationship:**

تسمّى العَلاقة بين جدول المؤلفين وجدول الكتب في المثال السابق بعلاقة "واحد بمتعدّد" One-to-many Relationship، وذلك لأنّ مؤلفًا واحدا يمكن أن يكون له أكثر من كتاب.

٢- **علاقة "متعدّد بمتعدّد" Many-to-many Relationship:**

افترض أن لدينا كتابا اشترك في تأليفه أكثر من مؤلف.. في هذه الحالة لن نضع العمود AuthorID في جدول الكتب، وبدلا من هذا سننشئ جدولا ثالثا هو جدول "مؤلفو الكتب" BooksAuthors.. هذا الجدول يسمى **جدول الربط Link Table** أو **جدول الاتصال Junction table**، وهو يتكون من عمودين فقط: رقم الكتاب BookID ورقم المؤلف AuthorID كالتالي:

BookID	AuthorID
١	٢٣
١	١٠٦
٢	٨
٣	٨

حيث اشترك المؤلّفان رقما ٢٣ و ١٠٦ في تأليف الكتاب رقم ١، بينما قام المؤلف رقم ٨ بتأليف الكتابين رقمي ٢ و ٣.. هذه هي علاقة متعدّد بمتعدّد Many-to-many.

٣- علاقة "واحد إلى واحد" One-to-one:

في هذه العلاقة يتم تقسيم السجل الواحد على جدولين، وذلك إذا كان عدد أعمدة الجدول كبيرا جدا، وكان بعضها قليل الأهمية ونادر الاستخدام، أو إذا كانت هذه الأعمدة تحتوي على بيانات سرية، لهذا يجب وضعها في جدول خاص ومنح صلاحية التعامل معها لمدير النظام مثلا.. بهذه الطريقة يكون جزء من السجل في الجدول الأول متاحا لكل المستخدمين، وباقي السجل في جدول آخر مخصص لمستخدمين بعينهم.

وفي علاقة "واحد بواحد" يكون كل من المفتاح الأساسي والمفتاح الفرعي متفردا Unique غير قابل للتكرار، وبهذا نضمن ارتباط كل مفتاح أساسي بمفتاح فرعي واحد فقط.

وكمثال: افترض أن لدينا جدولا لبيانات الموظفين اسمه Employees يبدو كالتالي:

ID	Employee	Phone	eMail	Address
١	أحمد شريف	....	....	....
٢	مجدي نوارة	....	....	....
٣	صالح رشدان	....	....	....
٤	ياسر الهواري	....	....	....

في هذه الحالة سنحتاج إلى جدول آخر نضع فيه اسم المستخدم وكلمة السر الخاصة بكل موظف، وليكن اسمه EmployeesPasswords، وسنربطه بالجدول Employees باستخدام مفتاح فرعي اسمه EmployeeID مع جعله مفتاحا متفردا Unique:

ID	USerName	Password	EmployeeID
١	ash	....	١

٢	mgdy	....	٢
٣	rshdan	....	٣
٤	ysrhwary	....	٤

لاحظ أن العمودين ID و EmployeeID في الجدول السابق متماثلان في تركيبهما وبياناتهما، لهذا لا بأس من التخلص من أحدهما (وليكن العمود ID).

٤- العلاقة الذاتية Self-Relation:

في بعض الأحيان يمكن إنشاء علاقة بين سجل من سجلات الجدول، وسجل آخر في نفس الجدول.. يسمى هذا بالعلاقة الذاتية Self-Relation، حيث يحتوي الجدول على عمود المفتاح الأساسي وعمود المفتاح الفرعي معا.. هذا مفيد عند تمثيل أي تركيب هرمي كالشجرة Tree.. في هذه الحالة يشير المفتاح الفرعي لكل سجل إلى المفتاح الأساسي للسجل الرئيسي الذي ينتمي إليه.. وعلينا تتبع هذه السلسلة إلى أن نصل إلى سجل رئيسي يعمل كجذر، وفي هذه الحالة يجب أن نضع صفرا أو -١ في خانة المفتاح الفرعي، للإشارة إلى عدم وجود سجل رئيسي لهذا السجل.. افترض مثلا أن عندنا شجرة العائلة التالية:

- كامل

- شريف
  - أحمد
  - نهى
- هاني
  - رشيد
  - فتحي
- عمر
  - نجوى
  - تقوى
  - معاذ

يمكن تمثيل هذه الشجرة في جدول واحد كما هو موضح.

ID	Name	ParentID
١	كامل	٠
٢	شريف	١
٣	أحمد	٢
٤	نهى	٢
٥	هاني	١
٦	رشيد	٥
٧	فتحي	٥
٨	عمر	١
٩	نجوى	٨
١٠	تقوى	٨
١١	معاذ	٨

في هذا الجدول، يعمل الحقل ID كمفتاح أساسي، بينما يعمل الحقل ParentID كمفتاح فرعي.. وبهذا يكون من السهل استخراج كل التفاصيل المطلوبة لو بدأنا من أي سجل.. مثلا:

- لمعرفة السجلات الفرعية للسجل الحالي، عليك بالبحث في الحقل ParentID عن الخانات التي تحتوي على القيمة الموجود في الحقل ID للسجل الحالي.. مثلا: فتحي ليس له أبناء، لأن معرفه ID = ٧، غير موجود في الحقل ParentID في أي سجل في الجدول.. بينما عمر له ثلاثة أبناء، لأن معرفه ID = ٨، موجود في الحقل ParentID في السجل رقم ٩ والسجل رقم ١٠ والسجل رقم ١١.

- ولمعرفة السجل الرئيسي للسجل الحالي، يكفي أن نقرأ المعرف ID المحفوظ في الحقل ParentID للسجل الحالي.. مثلا: عمر هو ابن الرجل الذي سجله يحمل المعرف ١، (وهو كامل).

إن هذه أفضل طريقة لتمثيل العلاقات الشجرية، فلو حاولت استخدام أكثر من جدول لتمثيلها، فسيكون عليك إنشاء جدول لكل مستوى من مستويات الشجرة، وإنشاء علاقات بينها، وهو أمر غير عملي ويضيع منك وقتا أطول في تصمم الجداول، خاصة لو ازداد عدد المستويات الفرعية للشجرة، كما أن كتابة الكود الذي يتعامل مع مثل هذه الجداول سيكون مزعجا ومعقدا!.. انظر كيف يمكن تمثيل الشجرة السابقة بثلاثة جداول:

جدول الجدود Grandparents	
ID	**Name**
١	كامل

جدول الآباء Parents		
ID	**Name**	**GrandParentID**
١	شريف	١
٢	هاني	١
٣	عمر	١

جدول الأبناء Sons		
ID	**Name**	**ParentID**
١	أحمد	١
٢	نهى	١
٣	رشيد	٢
٤	فتحي	٢
٥	نجوى	٣

٦	تقوى	٣
٧	معاذ	٣

طبعا سيكون الأمر كارثيا لو أردت تمثيل عدة أجيال أخرى، كالأحفاد وأبناء الأحفاد وهلم جرا!

القيود Constraints:

ماذا سيحدث لو تمّ حذف سجل توفيق الحكيم من جدول المؤلفين؟

في هذه الحالة ستظل هناك روايات في جدول الكتب تشير إلى سجل توفيق الحكيم، بينما هو محذوف.. هذا بالتأكيد وضع مثاليّ لحدوث أخطاءٍ مزعجة في برنامجك، لهذا يجبُ أن تراعي منطقيّة وصحّة العلاقات المرجعيّة، وهو ما يعرف باسم التكامل المرجعي بين الجداول **Referential Integrity**، فلو قمت بحذف سجل توفيق الحكيم من جدول المؤلفين، فيجب عليك أيضا أن تحذف كل الروايات والكتب التي ألفها من جدول الكتب.

وللتأكد من فرض هذا التكامل، يمكننا تعريف ما يسمى بالقيود Constraints.. مثلا: لو أضفت قيد المفتاح الأجنبي Foreign-Key Constraint، فإن قاعدة البيانات سترفض حذف أي مؤلف بينما ما تزال هناك كتب تابعة له.. هذا يجبرك على حذف كتب هذا المؤلف أولا قبل أن تقوم بحذفه.

ويوجد قيد آخر نوهنا إليه سابقا، وهو قيد التفرد Unique Constraint، وهو يفرض على المفتاح الأساسي للجدول لضمان تفرد معرف كل سجل.. ويفرض قيد التفرد أيضا على المفتاح الفرعي في حالة واحدة فقط، وهي عند التعامل مع علاقة واحد بواحد One-to-one.

العروض Views:

العرض هو طريقة لعرض بعض البيانات من جدول أو أكثر.. ويمكن أن تحفظ طريقة العرض هذه في قاعدة البيانات باسم خاص، إذا كنت تحتاج إلى الرجوع إليها أكثر من مرة.

مثلا: يمكنك أن تنشئ في قاعدة بيانات الكتب عرضا اسمه TopSales، يعرض الكتب الأكثر مبيعا.

لاحظ أن العرض لا يستهلك مساحة في قاعدة البيانات، لأن ما يحدث فعليا هو حفظ الاستعلام Query الذي يجمع بيانات هذا العرض، وعند فتح العرض، يتم تنفيذ هذا الاستعلام مرة أخرى لجمع البيانات من الجداول.. هذا يضمن أنك ترى دائما أحدث البيانات الموجودة في الجداول.

الفهارس Indices:

افترض أنّك تريد أن تبحث عن قيمة معيّنة في أحد الأعمدة، كالبحث مثلا عن (أحمد خالد توفيق) في حقل المؤلفين.. أوّل ما سيجول بذهنك هو أنّ الحاسب سيفحص أسماء المؤلفين واحدا وراء الآخر من البداية إلى النهاية.. ربّما تبدو هذه الطريقة سهلة، ولكنّها أسوأ طريقة يمكن أن يتمّ البحث بها، لأنها قد تتسبّب في بطءٍ شنيع في أداء البرنامج، خاصّة وأنّ الجدول قد يحتوي على عدد كبير من الصفوف، قد يصل إلى عشرات أو مئات الآلاف، وكل خانة من خاناته قد تحتوي على نصوص طويلة، وأنت تعرف أن مقارنة عددين تتم بمقارنة الوحدات الثنائية Bytes في كل منهما (لهذا يحتاج العدد الصحيح int مثلا إلى أربع عمليات مقارنة)، بينما تحتاج مقارنة نصين إلى مقارنة حروف كل منهما، مما يزيد كثيرا من عمليات المقارنة، فقد تحتاج قاعدة البيانات إلى إجراء ٥٠ عملية لمقارنة نصين اثنين طول أقصرهما ٥٠ حرفا بافتراض أسوأ حالة ممكنة، وهي تشابه أول ٤٩ حرفا في النصين!.. هذا مع العلم بأن حقول النصوص قد تحتوي على نصوص أطول من هذا بكثير!

لمثل هذه التحديات، توجد العديد من الطرق لتسريع عملية البحث، تبدأ بترتيب العمود أبجديّا، وذلك لأنّ البحث في عمود مرتب أسرع بكثير من البحث في عمود عشوائيّ، خاصّة مع استخدام طريقة بحث جيّدة، مثل البحث الثنائي Binary Search أو غيره من الطرق.

ولكن، رغم سرعة عملية البحث بعد الترتيب، إلا إنّ الترتيب نفسه قد يكون مأساة!.. هل تتخيّل كم من الوقت يمكن أن يضيّعه برنامجك لترتيب خمسين ألف نصّ مثلا، يتكون كل منها في المتوسط من ٣٠ حرفا؟.. فما بالك إذن حينما يقوم البرنامج بتكرار البحث في نفس العمود أكثر من مرّة؟!

إذن فما الحل؟

الحل هو أن نرتب بيانات العمود في قاعدة البيانات منذ البداية، وبهذا نضمن سرعة البحث، دون الحاجة إلى تكرار عملية الترتيب في كل مرة.. لكن هذا يقودنا إلى سؤال آخر: وماذا لو كان الجدول يحتوي على أكثر من عمود تحتاج للبحث فيها، فتبعا لأيّ منها سوف نرتب الجدول يا ترى؟!!

هنا يبرز حل رائع اسمه الفهارس Indexes:

تخيّل أننا طلبنا من قاعدة البيانات إنشاء فهرس لعمود الأسماء في جدول المؤلفين.. هنا سيتم ترتيب هذا العمود، مع إنشاء فهرس يحتفظ ببيانات توضح ترتيب كل مؤلف.. فمثلا، إذا كان توفيق الحكيم هو المؤلف رقم ١ في الجدول، ونبيل فاروق هو المؤلف رقم ٧، وأحمد خالد توفيق هو المؤلف رقم ١٠، فإن الفهرس سيحتفظ بمعلومات تفيد بأن أول مؤلف في الترتيب الأبجدي يوجد في الصف رقم ١٠، يليه في الترتيب المؤلف الذي يقع في الصف رقم ١، يليه المؤلف الذي يقع في الصف رقم ٧.. هذا يحقق لنا الفوائد التالية:

- يمكن استخدام هذا الفهرس لتطبيق خوارزميات البحث على عمود المؤلفين وكأنه عمود مرتب، وبالتالي ستكون عمليّة البحث أسرع.
- لن نحتاج إلى إجراء أي تغيير في طريقة عرض البيانات في قاعدة البيانات، حيث سيظل ترتيب المؤلفين في الجدول كما هو، وبالتالي سنترك للمستخدم حريّة ترتيب البيانات كما يحلو له.

- يمكننا إنشاء أكثر من فهرس لأكثر من عمودٍ في نفس الجدول، ممّا يعني قدرتنا على ترتيب كل هذه الأعمدة والبحث فيها.
- سنوفر وقت ترتيب العمود أثناء البرمجة، وإن كان هذا الوقت سيستهلك عند إنشاء الفهرس لأوّل مرّة في وقت التصميم.. لاحظ إن إنشاء الفهرس والجدول فارغ أو به القليل من الصفوف، يجعل عملية إنشاء الفهرس أسرع، وكلما أضفت سجلا جديدا إلى الجدول، يتم تحديث الفهرس لوضع هذا السجل في ترتيبه في الفهرس.

لاحظ أنك تستطيع إنشاء فهرس لأكثر من عمود معا، حيث سيتم الترتيب تبعا للعمود الأوّل، فإن تماثلت قيم بعض الصفوف (مؤلفان يشتركان في الاسم الثلاثي مثلا)، يتم ترتيب هذه الصفوف على أساس العمود الثاني (اسم الدولة مثلا).

ملحوظة:
بمجرد جعل الحقل مفتاحا أساسيا للجدول، يتم إنشاء فهرس له، نظرا لأن المفتاح الرئيسي يستخدم بغزارة في الاستعلامات التي تبحث عن سجلات معينة في الجدول، خاصة مع استخدامها في علاقات مع جداول أخرى.

ولكن.. للأسف لا توجد طريقة مثاليّة، فهناك عيبان أساسيّان للفهارس:

١- الفهارس المنشأة يتمّ حفظها في قاعدة البيانات، ممّا يعمل على زيادة حجمها.. لهذا إذا كان حجم قاعدة بياناتك كبيرا، فلا بدّ أن توازن بين حاجتك إلى توفير الوقت وحاجتك إلى توفير الحجم.. لكني لا أظن هذه مشكلة كبيرة اليوم، مع السعات الهائلة لوسائط التخزين.

٢- عند تعديل قيمة أيّ خانة في العمود المفهرس، أو حذف أو إضافة أيّ صفّ، لا بدّ من إعادة تحديث الفهارس لمراعاة الاختلاف الذي أحدثه هذا التغيير.. مثل هذا الأمر يسبّب عيبا خطيرا، هو بطء عمليّات التعديل والحذف والإضافة بصورة ملحوظة، تزداد سوءا مع ازدياد طول الجدول، ومع تكرار هذه العمليات في برنامجك بشكل كبير.. هنا يجب أن توازن في برنامجك، بين حاجتك إلى سرعة البحث وحاجتك إلى سرعة التحديث والحذف والإضافة.. لديك هذه الحالات:

أ- إذا كان برنامجك يقوم بمئات من عمليّات التحديث والإضافة والحذف بطريقة متتالية ومن عدد كبير من المستخدمين، بينما لا توجد الكثير من عمليات البحث، فلا داعي لإنشاء الفهرس!

ب- إذا كنت تستخدم قاعدة بيانات يتعامل معها المستخدمون في معظم الوقت لقراءة البيانات وإجراء الاستعلامات وعمليات البحث، فإن الفهرس سيجعل هذه العمليات أسرع بكثير، ومع قلة عمليات التحديث والإضافة والحذف (المتاحة فقط لمدير النظام)، لن يكون عبء الفهارس ملحوظا.

ج- إذا كنت تحتاج إلى كلا النوعين من العمليات في نفس الوقت، فيمكنك تقليل العبء الناتج من الفهارس في برنامجك بحيلة صغيرة: فعندما تكتب في الكود حلقات التكرار Loops أو الاستعلامات التي تقوم بتعديل أو إضافة أو حذف مئات أو آلاف السجلات على التوالي، فاحذف الفهرس من الجدول قبل بدأ تنفيذ هذا الكود، ثمّ قم بإعادة إنشائه مرّة أخرى بعد انتهاء هذه العمليات، وبهذا تقلل من عبئه وتستفيد منه في تسريع عمليات البحث في نفس الوقت!

لكن عليك ملاحظة أن هذه الحيلة ستكون خطيرة إذا كان هناك عدد كبير من المستخدمين يتعاملون مع قاعدة البيانات في نفس اللحظة، وبعضهم يبحث وبعضهم يحذف السجلات!!.. لهذا فأنت الوحيد الذي يعرف طبيعة عمل البرنامج وأنسب اختيار لجعله أكفأ ما يمكن.

وكقاعدة: أنشئ فهارس للحقول النصية التي تتوقع أن يحتاج مستخدمو برنامجك للبحث فيها بغزارة.

وسنتعرّف على كل هذه المفاهيم عمليّا، أثناء إنشاء قواعد البيانات، وأثناء استخدامها من خلال برامجنا.

– ٢ –

قواعد بيانات Access

لن تستطيع التعامل مع قواعد البيانات، ما لم تكن لديك المقدرة على إنشاء قواعد البيانات وما تحتويه من جداول.. تعالَ نأخذ فكرةً سريعةً عن هذا الأمر.

إنشاء الجداول بتطبيق Access:

Access هو أحد تطبيقات Microsoft Office.. لن تجد فارقا يذكر بين إصدارات Access 2000 و Access XP و Access 2003.. لكنك ستجد اختلافا كبيرا في واجهة الاستخدام بدءا من Access 2007.. دعنا هنا نتعامل مع Access XP وعليك التعامل مع الفوارق البسيطة بين الإصدارات المختلفة بنفسك.

افتح تطبيق Access ومن القائمة "ملفّ" File اختر "جديد" New.. أنا أكتب أسماء القوائم بالعربية والإنجليزية لأن البعض لديه واجهة استخدام عربية، والبعض لديه واجهة إنجليزية، على حسب النسخة التي أعدها على جهازه.

ستظهر لك نافذة "ملفّ جديد" New File في جانب واجهة التطبيق كما في الصورة التالية:

تحت الشريط "جديد" New اضغط الأمر "قاعدة بيانات فارغة" Blank Database.. سيظهر لك مربّع حوار "حفظ ملفّ" Save-File Dialog، وهو يطلب منك تحديد اسم قاعدة البيانات وموقع حفظها على الجهاز.. غيّر اسم قاعدة البيانات من الاسم الافتراضي إلى "Books.mdb"، وقم بحفظها في الموضع الذي تريده.

بعد إغلاق مربّع الحوار ستظهر لك نافذة قاعدة البيانات كما في الصورة التالية:

انقـــــر الأمـــــر "إنشـــــاء جـــــدول فـــــي طريقـــــة عـــــرض التصـــــميم" Create Table in Design View مرّتين بالفأرة.. ستظهر لك نافذة تصميم الجدول كمـــا في الصورة التالية:

كل ما عليك الآن، هو كتابة أسماء الحقول وتحديد نوع البيانات التي ستضعها في كل منها.. تعالَ نبدأ بتصميم جدول أسماء المؤلفين.. هذا الجدول سيتكوّن من الحقول التالية:

	اسم الحقل Field Name	نوع البيانات Data Type	الوصف Description
	ID	(ترقيم تلقائي) AutoNumber	المفتاح الرئيسيّ لهذا الجدول
	Author	Text (نص)	اسم المؤلف
	CountryID	(رقم) Number	رقم الدولة التي ينتمي إليها المؤلف.
	Phone	Text (نص)	رقم هاتف المؤلف.
	About	(مذكرة) Memo	نبذة عن المؤلف

نصيحة:
سمّ الجداول والأعمدة بأسماء أجنبيّة، لأنّ هذه الأسماء ستدخل في كتابة الكود في لغات البرمجة، وسيكون من المرهق الانتقال من الإنجليزية إلى العربيّة والعكس أكثر من مرّة أثناء كتابة الكود.

بعد أن تنشئ هذه الحقول، قم بما يلي:

- اضغط بزر الفأرة الأيمن على المربّع الرماديّ المجاور للحقل ID، ومن القائمة الموضعيّة Context Menu اضغط الأمر "مفتاح أساسي" Primary Key لجعل هذا الحقل مفتاحا أساسيّا للجدول.. ستلاحظ ظهور رمز المفتاح أمام هذا الحقل.. ستلاحظ كذلك أنّ الخاصيّة "مفهرس" Indexed في المنطقة السفلى من النافذة ستتحوّل إلى "نعم بدون تكرار" Yes (No Duplicates)، وهو شيء متوقع، فلقد اتفقنا أنّ قيم المفتاح الأساسيّ غير قابلة للتكرار.

- اضغط بالفأرة على حقل اسم المؤلف Author، وفي المنطقة السفلى من النافذة غيّر قيمة الخاصيّة "حجم الحقل" Field Size إلى ٣٠ بدلا من ٥٠، وذلك بافتراض أن أطول اسم مؤلف لن يزيد عن ٣٠ حرفا.
- اضغط بالفأرة على الحقل CountryID، وفي المنطقة السفلى من النافذة غيّر قيمة الخاصيّة "حجم الحقل" Field Size إلى Byte.. السبب في هذا أننا لن نتعامل إلا مع ٢٢ دولة عربية فقط.. وحتى لو أضفت بعض البلاد الأجنبية، فإن الوحدة الثنائية Byte تقبل أرقاما من ٠ إلى ٢٥٥، وهو ما يستوعب أهم أسماء دول العالم.. لاحظ أن اختيار حجم أقل للحقل يوفر مساحة قاعدة البيانات، ويجعل التعامل معها أكفأ.. لكن عليك أن تراعي أي توقعات مستقبلية للتوسع، فمحاولة تكبير حجم الحقل بعد ذلك تستغرق وقتا ملموسا في قواعد البيانات الضخمة، وهو ما قد يؤدي إلى تعطيل الموظفين الذين يتعاملون مع قاعدة البيانات في ذلك الوقت.. لهذا ينصح بإجراء التعديلات على تركيب قاعدة البيانات في الأوقات التي يقل فيها تعامل الموظفين والعملاء معها.
- اضغط بالفأرة على الحقل Phone، وفي المنطقة السفلى من النافذة غيّر قيمة الخاصيّة "حجم الحقل" Field Size إلى ٢٠.. أعرف أنك تتساءل لماذا لم نجعل رقم الهاتف رقما.. في الحقيقة هناك عدة أسباب:

١- حفظ رقم الهاتف كنص يتيح لك وضع بعض المسافات وعلامات الترقيم المميزة فيه، مثل:

+2 - 0102020375
+2 – 050 – 17354-036

هذا يجعله أكثر وضوحا عند قراءته.. لكن البعض يرد على هذه النقطة قائلا إن من الممكن حفظ رقم الهاتف في قاعدة البيانات كرقم، وعند عرضه في لغة البرمجة يمكن تنسيقه بأي شكل مطلوب.. من وجهة نظري أن هذا قد يسبب بعض المشاكل، لاختلاف تركيب رقم الهاتف من دولة إلى أخرى.

٢- نظرا لوجود مفتاح لكل دولة، ومفتاح لكل منطقة، فإن رقم الهاتف سيكون عددا كبيرا جدا، ولن يكفيه العدد الصحيح الطويل

Long Integer (وهو يستخدم 4 Bytes)، وهو ما يتطلب استخدام نوع بيانات كبير، وفي هذه الحالة لن يكون هناك توفير كبير عن استخدام النص!

٣- أهم نقطة في الأمر، هو أن تطبيق آكسيس يحذف الأصفار المكتوبة على يسار أي رقم، لهذا لو كتبت 0020102020375 فسيحولها إلى 20102020375، وهو ما سيجعل من الصعب عليه معرفة مفتاح الدولة!

٤- إذا أردنا حل مشاكل استخدام الأرقام، فلا بد من حفظ مفتاح كل دولة بمفرده في عمود خاص به.. في هذه الحالة يجب وضعه في جدول أسماء الدول وربطه بجدول المؤلفين.. وربما نحتاج إلى فعل نفس الشيء مع مفتاح كل منطقة (مفتاح المحافظة).. فعل هذا قد يجعل الأمور أعقد برمجيا ويضيع وقتا دون أن يوفر مساحة تذكر.

الآن اضغط Ctrl+S لحفظ الجدول.. سيظهر لك مربّع إدخال يطالبك بإدخال اسم الجدول.. سمّه Authors واضغط زر OK.

الآن لو أغلقت نافذة التصميم، فستجد الجدول Authors قد أضيف إلى نافذة قاعدة البيانات.. انقره مرّتين بالفأرة.. سيؤدي هذا إلى فتح نافذة جديدة تعرض لك جدول المؤلفين في وضع التحرير، حيث يمكنك إدخال بيانات كل مؤلف، مع ملاحظة أن حقل الترقيم التلقائي ID لا يسمح لك بالكتابة، فقيمته تزيد تلقائيا كلما أدخلت سجلا جديدا.

ملاحظات حول تحرير الجداول:

- ضغط رأس العمود بالفأرة، يحدد العمود كله.
- ضغط رأس الصفّ بالفأرة، يحدد الصفّ كله.
- يمكن الانتقال من خانة إلى أخرى بضغط الزرّ Tap.
- لإدخال سجل جديد اكتب في السجل الفارغ في نهاية الجدول.
- عند تغيير قيمة أيّ خانة، لا يتمّ حفظ التغييرات إلا عند الخروج من الصف الذي توجد فيه إلى صف آخر، أو عند ضغط زر الحفظ.. فإذا وجد آكسيس أن إحدى القيم غير مقبولة، عاد إلى نفس الصف وأعلن رفضه للتغييرات التي قمت بها، ولا يسمح لك

بمغادرة الصف قبل تصحيح القيمة الخاطئة.

- لحذف أيّ صفّ، حدّده واضغط الزر Delete من لوحة المفاتيح، أو اضغط أمر Delete من القائمة الموضعية Context Menu.

وبنفس الطريقة يمكنك تصمّم باقي الجداول.. ستجد هذه الجداول في قاعدة البيانات Books.mdb المرفقة بأمثلة هذا الكتاب، وستجد فيها الجداول التالية:

Authors	جدول المؤلفين.
Books	جدول الكتب.. لاحظ أننا سنحفظ في هذا الجدول رقم الناشر ورقم المؤلف وعدد النسخ الكلية المتاحة من كل كتاب، وسعر النسخة، وتاريخ طباعة الكتاب، ورقم الطبعة.
Classes	جدول أصناف الكتب، وهو مرتبط بجدول الكتب.
Countries	أسماء الدول، وهو مستخدم مع جدولي المؤلفين والناشرين.
Publishers	جدول الناشرين.
Sales	جدول مبيعات الكتب، وهو يحتوي على فواتير بيع الكتب، حيث سنسجل عدد النسخ المبيعة من كل كتاب، والخصم الممنوح للمشتري، واسم المشتري، وتاريخ البيع.

إنشاء علاقات بين الجداول:

دعنا الآن نربط حقلي رقم المؤلف بين جدولي المؤلفين والكتب، لتحقيق التكامل المرجعيّ Referential Integration بين الجدولين:

من قائمة الأدوات Tools اضغط الأمر "علاقات" Relationships.. ستظهر لك نافذة "إظهار جدول" وفيها أسماء كل جداول قاعدة البيانات.. اختر الجدول الذي تريد استخدامه في العلاقة واضغط زرّ الإضافة Add، وكرر هذا مع كل الجداول المطلوبة.. بعد أن تنتهي أغلق هذه النافذة، لتظهر لك نافذة العلاقات، وفيها ستجد كل جدول في صورة مستطيل عليه عنوان الجدول، وبه أسماء أعمدته.

ولإنشاء علاقة بين الجدولين، اسحب الحقل ID بالفأرة من جدول المؤلفين، وتحرّك بالفأرة إلى الحقل AuthorID في جدول الكتب.. ستلاحظ تغيّر شكل مؤشّر الفأرة.. اترك زر الفأرة الأيسر فوق هذا الحقل.. هنا ستظهر لك نافذة إنشاء العلاقة بين الحقلين:

اضغط الاختيار "فرض التكامل المرجعي" Enforce Referential Integrity، لتقوم قاعدة البيانات آليّا بالتحقّق من صحّة البيانات بين الجدولين.. وفي هذا الصدد لديك اختياران:

- **تتالي تحديث الحقول المرتبطة Cascade Update Related Fields:**

فمثلا، لو غيّرت رقم المؤلف (توفيق الحكيم) من ١ إلى ١٠، فسيتمّ تغيير رقم المؤلف AuthorID تلقائيّا ليصير ١٠، في سجلات كل الكتب التي ألفها.

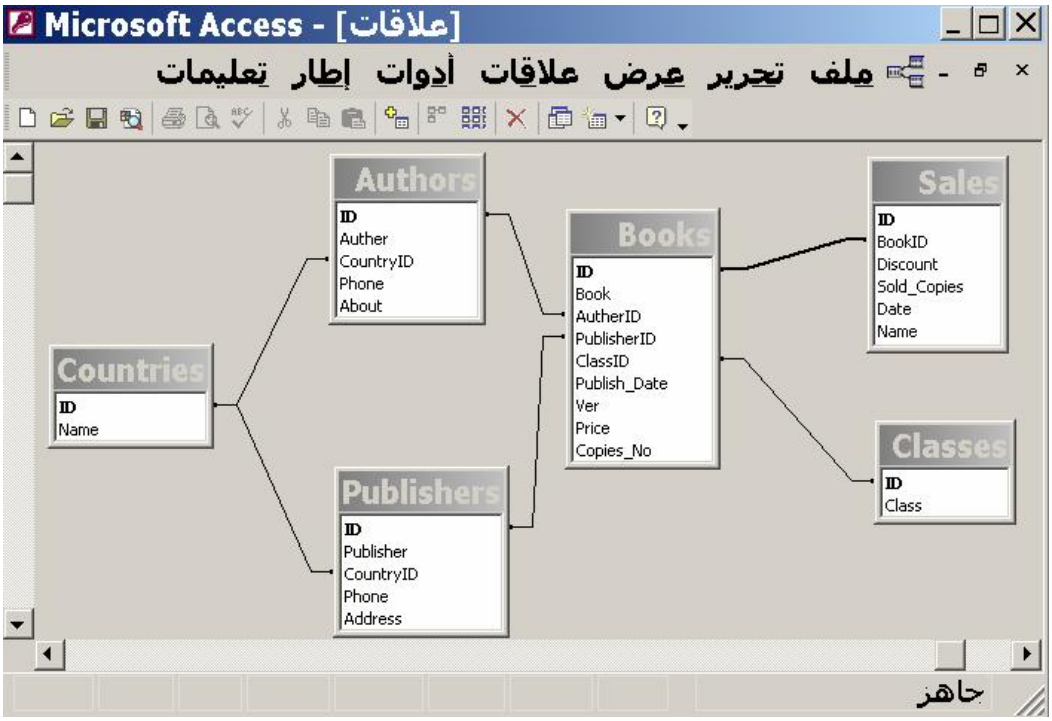
ملحوظة:
لن تستطيع تغيير قيمة رقم المؤلف ID في جدول المؤلفين، لأنّه ترقيم تلقائيّ.. ولو أردت تغييره، فعليك بتغيير نوع الحقل من ترقيم تلقائي إلى رقم، على أن يكون هذا قبل إنشاء العلاقة، فمن غير المسموح تغيير نوع بيانات حقل داخل في علاقة.. فإذا كنت قد أنشأت العلاقة فعلا، فعليك حذفها أولا ثم تغيير نوع الحقل، ثم إعادة إنشاء العلاقة مجددا!

- **تتالي حذف الحقول المرتبطة Cascade Delete Related Fields:**

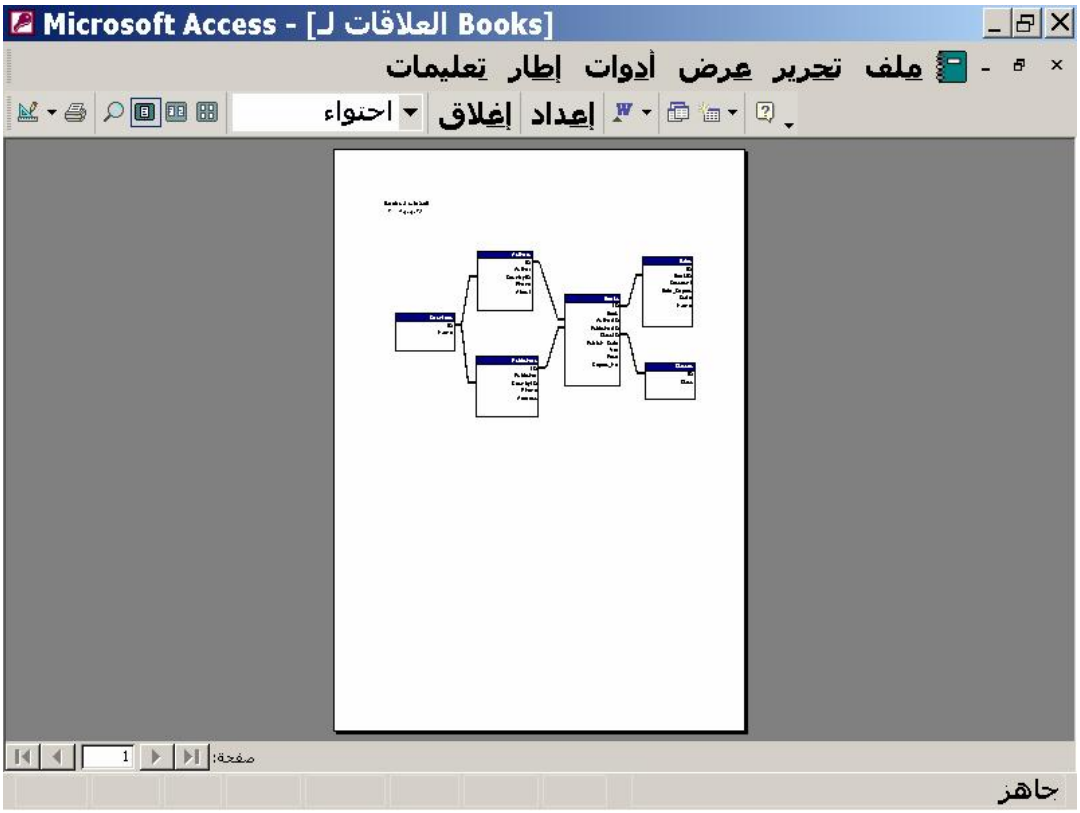
فمثلا، لو حذفت سجل المؤلف (نبيل فاروق) من جدول المؤلفين، فسيتمّ حذف كل كتب (نبيل فاروق) تلقائيّا من جدول الكتب.

اضغط الزرّ "إنشاء" Create لإغلاق هذه النافذة.. الآن سيظهر خطّ يربط الجدولين.. ولو أردت حذف أي علاقة، فاضغط الخط الخاص بها بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اختر "حذف" Delete.

بالمثل يمكنك إنشاء العلاقات بين جداول قاعدة بيانات الكتب، كما هو موضح في الصورة:

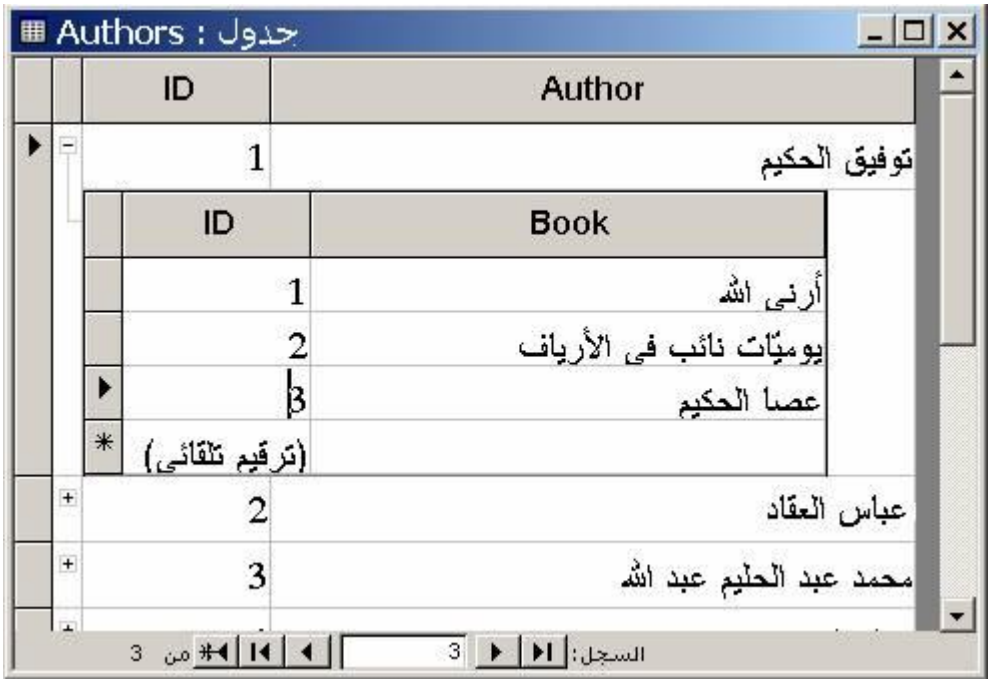


وكما ترى، تقدم لك نافذة العلاقات مخططا بيانيا يلخص كل تفاصيل قاعدة البيانات والعلاقات التي تربط بين جداولها، مما يسهل عليك مراجعة تصميمها في أي لحظة، كما يمكنك طباعة هذا المخطط باستخدام الأمر "طباعة العلاقات" Print Relations من القائمة File، حيث سيؤدي هذا إلى عرض نافذة جديدة بها مخطط العلاقات في شكل صفحة قابلة للطباعة:



في هذه النافذة يمكنك استخدام الأمر "إعدادات الطباعة" Setup Rrint و "طباعـــة" Print من القائمة File لطباعة الصفحة المعروضة.

قم بحفظ التغييرات، وأغلق نافذة العلاقات.. افتح الآن جدول المؤلفين.. ســتلاحظ ظهــور العلامة "+" بجوار كل حقل.. اضغط أيّا منها، ولتكن المجاورة لــ (توفيق الحكيم).. ســتجد أنّ العلامة "+" تحوّلت إلى العلامة "–"، وأنّ جدولا صغيرا يحتوي على الكتب التي ألفهــا توفيق الحكيم قد ظهر.. ولإخفائه ثانيةً أعد ضغط العلامة "–".



ويمكنك إدخال المزيد من كتب (توفيق الحكيم) في هذا الجدول الفرعيّ، دون أن تُدخل رقمه في الخانة AuthorID في كل مرّة، إذ ستتمّ كتابة هذا الرقم آليّا، ممّا يوفر عليك الوقت ويعفيك من احتمالات الخطأ.

إنشاء قيد Constraint:

افترض أنك لسبب ما تريد أن تمنع مُدخل البيانات من كتابة مؤلف اسمه (موهوب) في حقل أسماء المؤلفين.. لفعل هذا افتح تصميم جدول المؤلفين، وحدّد الحقل Authors.. ستجد ضمن الخصائص الموجودة في المنطقة السفليّة من النافذة خاصيّة تسمّى "قاعدة التحقق من الصحّة" Check Constraint.. هذا هو المكان الذي سنظلم فيه المؤلف العبقريّ (موهوب) ونحرمه من حقّه المشروع في الظهور في قائمة المؤلفين!!

اكتب في هذه الخانة ما يلي:

<> "موهوب"

اضغط زرّ الحفظ.. ستظهر لك رسالة تستوثق من رغبتك في أداء هذه القاعدة، التي ربّما تستهلك بعض الوقت نتيجة فحص أسماء المؤلفين أولا للتأكد من أنها تخضع لهذا القيد.. اضغط OK.

افتح تخطيط الجدول، وحاول أن تكتب الاسم (موهوب) في حقل المؤلفين.. تذكر أنّ التغيير لن يحدث قبل أن تغادر الصف الذي تكتب فيه.. وفور أن تفعل هذا، ستظهر لك رسالة تخبرك أنّ إحدى القيم المدخلة غير مسموح بها، وسيعود المؤشّر إلى تلك الخانة مرة أخرى!.. غيّر هذا الاسم، فمهما حاولت لن تستطيع إقناع قاعدة البيانات بقبوله.

هذه الإمكانيّة مفيدة، وتمكنك من وضع شروط معقدة تدخل فيها بعض دوال فيجيوال بيزيك المسموح باستخدامها في جمل SQL.. ويمكنك أن تضغط الزرّ المجاور لخانة التحقق من الصحّة لتظهر لك نافذة تساعدك في بناء الشرط، حيث تعرض لك كل الدوال والتعبيرات المسموح باستخدامها.

ضغط قاعدة البيانات Database Compacting:

عندما تحذف أحد السجلات من أحد الجداول، فإنه يختفي من عرض الجدول، ولكنه في الواقع يظل في موضعه في ملفّ قاعدة البيانات.. كل ما حدث هو أنّ قاعدة البيانات قد أشارت إلى حالته باعتباره سجّلا محذوفا، حتى لا يتمّ التعامل معه.. ولو حدث في برنامجك أن أصبح السجل الذي تتعامل معه محذوفا (بسبب قيام مستخدم آخر لقاعدة البيانات بحذفه)، فإنّ محاولة قراءة البيانات أو كتابتها في هذا السجل ستؤدّي إلى حدوث خطإ في برنامجك.

وقد تظن أنّ هدف إخفاء السجل المحذوف هو منحك القدرة على التراجع عن حذفه.. ولكنّ هذا ليس صحيحا على الإطلاق، فالسجل الذي يحذف لا يمكن استعادته، رغم أنه ما زال موجودا في ملفّ قاعدة البيانات موسوما بأنه محذوف!

لماذا إذن تستخدم قاعدة البيانات هذه الطريقة؟

تخيّل أنك تريد حذف سجل موجود في منتصف ملفّ قاعدة البيانات.. كيف في نظرك يحدث هذا؟

لحذف هذا السجل يجب أن ينشئ تطبيق قاعدة البيانات ملفّا احتياطيّا، وينسخ إليه البيانات السابقة للسجل المراد حذفه، ثمّ البيانات التالية له، وبهذا يتكوّن ملفّ جديد لا يحتوي على السجل المحذوف.. بعد هذا يجب أن يحذف التطبيق ملفّ قاعدة البيانات القديم، ثمّ يسمّى الملفّ الجديد باسم قاعدة البيانات.. واضح أنّ هذه الطريقة ستؤدي إلى بطء عمليّة الحذف، بسبب الاضطرار إلى نسخ كل محتويات ملفّ قاعدة البيانات لمجرد حذف سجل واحد.. تخيل حجم الكارثة التي ستحدث لو حاولت حذف ألف سجل على التوالي بهذه الطريقة!

لهذا كان الحل هو أن يتمّ ترك السجل المحذوف في موضعه، على أن تتمّ الإشارة إلى حالته باعتباره سجلا محذوفا.

ولكنّ معنى هذا أنّ حجم قاعدة البيانات يمكن أن يتضخّم باستمرار مع استمرار العمل عليها والإضافة إليها، لأنّ كل ما يحذف منها يظل في موضعه.. لدرجة أنّك لو حذفت كل السجلات من كل الجداول، فسيظل لقاعدة البيانات نفس حجمها!!

ولحل هذه المشكلة، يجب أن تقوم بين فترة وأخرى بعملية تسمى "ضغط قاعدة البيانات" Database Compacting، وفيها يتمّ إنشاء نسخة جديدة من ملفّ قاعدة البيانات لا تحتوي على السجلات المحذوفة.

وللقيام بهذا في Access، اضغط قائمة الأدوات Tools، ومنها اضغط القائمة الفرعيّة "أدوات مساعدة لقواعد البيانات" ومنها اضغط الأمر "ضغط قاعدة بيانات وإصلاحها" Compact and repair database.. هنا سيقوم Access بإغلاق قاعدة البيانات وضغطها ثمّ إعادة فتحها.. ولو اختبرت حجم ملفّ قاعدة البيانات، فربّما تجده أقل مما كان عليه.

***

بهذه النبذة السريعة، نكون قد أنهينا حديثنا عن Access.. وطبعا ليست هذه هي كل إمكانيّات هذا التطبيق الشهير، فهو قادر في حدّ ذاته على إنشاء النماذج والتقارير، وكتابة كود لغة فيجيوال بيزيك للتطبيقات:

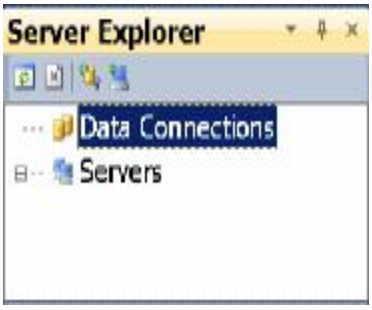
Visual Basic For Applications (VBA)

وهو ما يخرج عن نطاق هذا الكتاب.. إنّ Access له وحده مراجع متخصصة!

والآن، دعنا نتعرف على كيفية الاتصال بقاعدة بيانات Access في بيئة فيجوال ستديو دوت نت VS.NET 2010.

متصفح الخوادم Server Explorer:

توجد في فيجيوال ستديو نافذة اسمها متصفح الخوادم Server Explorer، وهي تحوي كل الأدوات الأساسيّة اللازمة للاتصال بقاعدة البيانات والتعامل معها.. ويمكنك عرض هذه النافذة بضغط الأمر Server Explorer من القائمة View.. وعندما تظهر لك نافذة متصفح الخوادم، ستجد فيها عنصرين:



روابط البيانات Data Connections:

تحت هذا العنصر، تظهر أسماء قواعد البيانات التي تتعامل معها في تطبيقاتك المختلفة.. وسنرى كيف يمكن الاتصال بقواعد البيانات لاحقا.

الخوادم Servers:

تحت هذا العنصر، تظهر أسماء خوادم قواعد البيانات Database Servers التي تتعامل معها، وما تمنحه لك من كائنات.. وستجد مبدئيا اسم جهاز الحاسب الخاص بك موجودا تحت هذا الفرع، وذلك لأنه يعمل كخادم محلي Local Server لتتمكن من خلاله من التعامل مع قواعد البيانات الموجودة على جهازك.

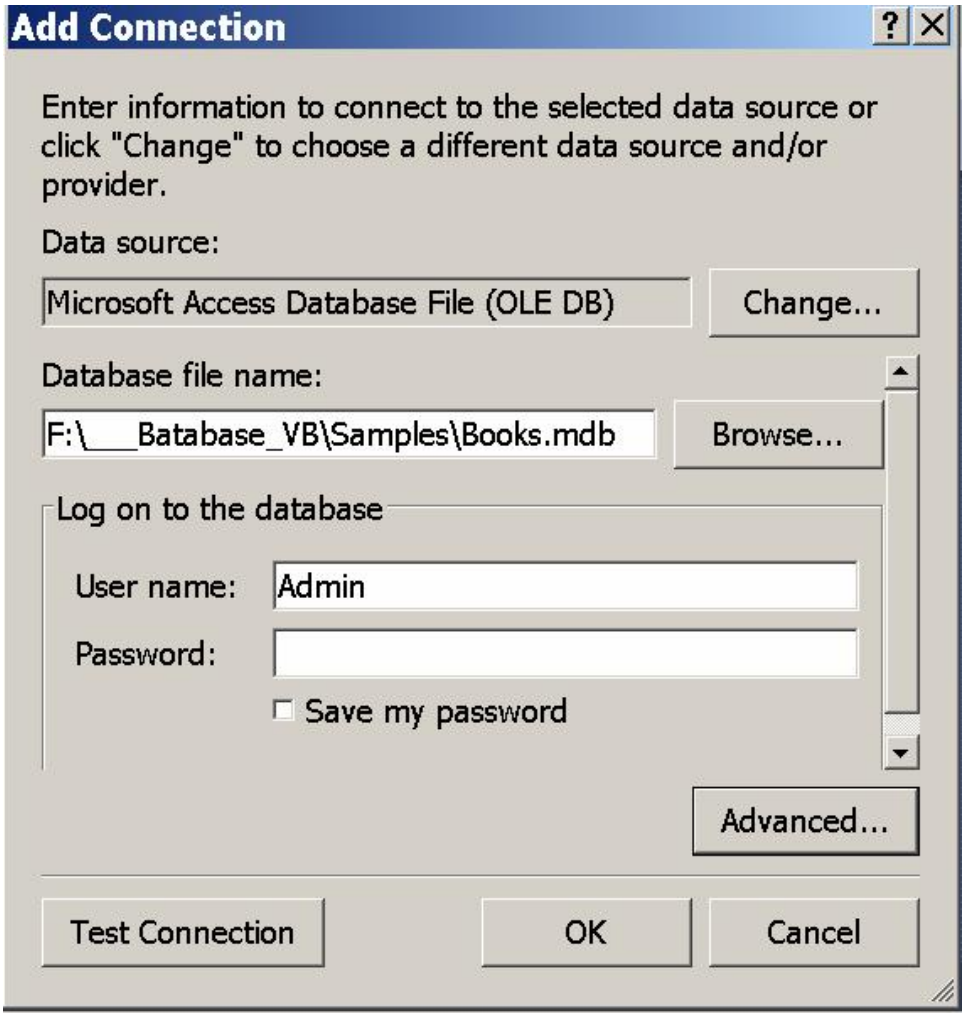
الاتصال بقواعد بيانات Access:

اضغط بزر الفأرة الأيمن العنصر Data Connections في متصفح الخوادم، ومن القائمة الموضعيّة اضغط الأمر "إضافة اتصال" Add Connection.. سيظهر لك مربّع حوار "اختيار مصدر البيانات" Choose data source، وفيه قائمة تحتوي على أسماء العديد من برامج قواعد البيانات:

اختر من القائمة العنصر: Microsoft Access Database File.. لاحظ وجود قائمة منسدلة أسفل النافذة، تتيح لك اختيار مزود البيانات Data Provider الذي ستستخدمه في برنامجك للاتصال بقاعدة البيانات.. بالنسبة لقواعد بيانات Access لن تحتوي القائمة المنسدلة إلا على المزود الخاص بقواعد بيانات OLE DB:

.Net Framework Data Provider for OLE DB

بعد هذا اضغط الزر Continue للانتقال إلى النافذة التالية، وهي نافذة "إضافة اتصال" Add Connection:



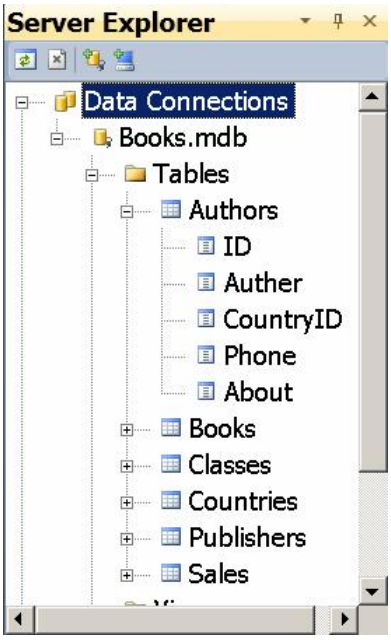
ستجد اسم مزود البيانات في الخانة العلوية في النافذة.. لو أردت تغيير هذا المزود، فيمكنك الضغط على الزر Change للرجوع إلى النافذة السابقة واختيار مزود لنوع قاعدة بيانات من نوع آخر.

الآن عليك اختيار ملف قاعدة البيانات، وبكتابة مسار واسم الملف مباشرة في الخانة Database File Name، أو ضغط الزرّ Browse لاختيار ملفّ قاعدة البيانات من على القرص الصلب.. اختر الملفّ Books.mdb من الموضع الذي حفظته فيه على جهازك (ستجد هذا الملف على القرص الضوئي المرفق بهذا الكتاب).

ويتيح لك النصف السفلي من النافذة إدخال اسم المستخدم وكلمة السر الخاصة بقاعدة البيانات.. لكن نظرا لأننا لم نضع كلمة سر لقاعدة بيانات الكتب، فاترك هاتين الخانتين بدون تغيير: اسم المستخدم الافتراضيّ Admin، وكلمة السر فارغة.

ويمكنك ضغط الزر Test Connection لاختبار الاتصال بقاعدة البيانات.. ستظهر لك رسالة تؤكد نجاح الاتصال.. أما لو فشل الاتصال لسبب ما، فستظهر رسالة خطأ تخبرك أن هناك مشكلة.

اضغط الزر OK لإغلاق مربّع الحوار.. سيظهر اسم قاعدة البيانات Books.mdb في متصفح الخوادم، كما هو موضح في الصورة المجاورة.



اضغط العلامة "+" بجوار اسم قاعدة البيانات.. سيتم إسدال العناصر التي تتكون منها قاعدة البيانات.

أوّل عنصر من هذه العناصر هو الجداول Tables، ولو أسدلت ما تحتها من عناصر، فستجد أسماء الجداول التي أنشأناها في قاعدة البيانات.

ولو أسدلت عناصر أي جدول، فستجد تحته أسماء الأعمدة التي يحتويها.

-٣-

قواعد بيانات SQL Server

كانت إصدارات فيجيوال ستديو السابقة لدوت نت، تدعم استخدام قواعد بيانات Access باعتبارها أهم قاعدة بيانات أصدرتها ميكروسوفت.. لكن مع ظهور دوت نت، احتلت قواعد بيانات SQL Server هذه المكانة، نظرا لأنها الأنسب للتعامل مع قواعد البيانات العملاقة التي يتصل بها آلاف المستخدمين في نفس اللحظة عبر شبكة داخلية Network أو عبر الشبكة الدولية Internet.. لعل هذا يوضح لك سبب وجود المصطلح "Server" في اسمها، فقواعد بيانات سيكويل سيرفر يجب أن تعمل على خادم محلي Local Server أو خادم موجود عبر الشبكة Online Server، ليقوم بتنظيم عمليات الاتصال بها وطلب البيانات منها.. ولو تم فصل قاعدة البيانات من هذا الخادم Disconnected، لا يمكن استخدامها إلا بعد إعادة توصيلها بالخادم مجددا Connected.. إذن فنحن نتعامل هنا مع خادم البيانات، ولا نتعامل مع ملف البيانات مباشرة كما كنا نفعل في قواعد بيانات آكسيس.
وهناك ثلاثة إصدارات شهيرة من سكيول سيرفر حتى الآن:

١- SQL Server 2000:

وهو متكامل مع دوت نت ٢٠٠٢ ودوت نت ٢٠٠٣، ولاستخدامه يجب أن تعد على جهازك أولا برنامجا اسمه:

Microsoft SQL Server Desktop Engine

أو اختصارا (MSDE)، والذي ستجده على أسطوانة Microsoft Office XP في المجلد MSDE2000.

٢- SQL Server 2005:

وهو متكامل مع دوت نت ٢٠٠٥ ودوت نت ٢٠٠٨، ولاستخدامه يجب أن تعد على جهازك أولا برنامجا اسمه:

SQL Server Management Studio Express

وهو نسخة مجانية خفيفة من SQL Server 2005 تمتلك قدرات أقل من النسخة الكاملة، لكنها كافية للغاية لإنشاء قواعد بيانات ضخمة يصل حجمها إلى ٤ جيجا بايت، ويمكنها أن تتصل بأكثر من ٣٢ ألف قاعدة بيانات، والاستجابة لاتصال بضع مئات من المستخدمين في نفس اللحظة.. ويمكنك الحصول على نسخة من هذا البرنامج من صفحة تنزيل البرامج من موقع ميكروسوفت:

http://www.microsoft.com/downloads

في أعلى هذه الصفحة ستجد مربع البحث.. اكتب فيه النص:

SQL Server Management Studio

واضغط زر البحث (الذي تمثله أيقونة العدسة) .. ستجد في نتائج البحث رابط البرنامج الذي تبحث عنه.. اضغطه لعرض صفحة تنزيله، واضغط زر تنزيل النسخة المناسبة لنظام التشغيل الموجود على جهازك.

٣- SQL Server 2008:

وهو متكامل مع دوت نت ٢٠١٠، ولاستخدامه يجب أن تعد على جهازك أولا برنامجا اسمه:

SQL Server 2008 Management Studio Express

ويمكنك الحصول على نسخة منه من صفحة تنزيل البرامج من موقع ميكروسوفت بطريقة مماثلة للتي شرحنها سابقا.

تنويه:

كان ذلك حتى كتابة هذا الكتاب.. حاليا وصل سيكويا سيرفر إلى الإصدار ١٦ والإصدار ١٩ في الطريق.

إعداد Management Studio Express:

قبل أن تبدأ إعداد هذا البرنامج على جهازك، عليّ أن أنبهك إلى ضرورة التأكد من أنك قد اخترت إعداد العنصر:

Microsoft SQL Server 2008 Express Edition

أثناء إعداد فيجيوال ستديو دوت نت ٢٠١٠، وإلا فإن عليك إعادة تشغيل ملف إعـداد دوت نت، واختيار Add or Remove Features، ومن ثم وضع علامة الاختيـار أمـام هـذا العنصر، حتى تستطيع فيجوال ستديو التعامل مع سيكويل سيرفر ٢٠٠٨.. أيضـا، يحتـاج برنامج الإعداد وجود البرامج التالية على جهازك:

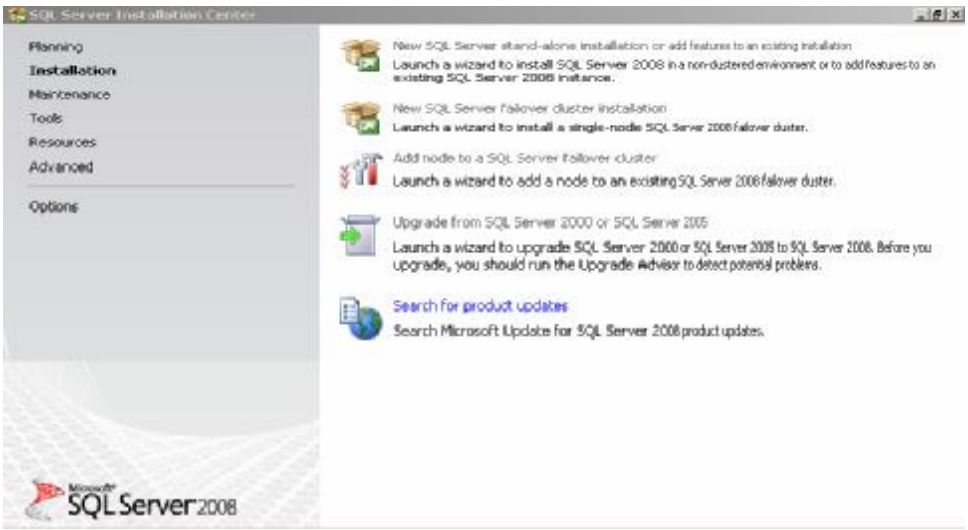
Windows Installer 4.5
WS-Management v1.1
Windows PowerShell V2 (CTP3)
.NET Framework 3.5 SP1

ويمكنك الحصول عليها من صفحة تنزيل البرامج من موقع ميكروسوفت كما شرحنا أعلاه.

والآن، اضغط برنامج الإعداد الخاص بـ:

SQL Server 2008 Management Studio Express

وانتظر حتى تظهر الشاشة الموضحة في الصورة:



من الهامش الأيسر، اضغط الاختيار الثاني Install، حيث سيعرض لك الجزء الأيمن مـن النافذة عدة اختيارات.. إذا لم تكن قد أعددت إصدارا سابقا مـن سـيكويل سـيرفر علـى

جهازك، فاضغط الاختيار الأول.. سيتم أولا التأكد من وجود المتطلبات اللازمة للإعداد على جهازك.. فإن كانت كلها موجودة، فاضغط OK وواصل باقي خيارات الإعداد.. وتأكد دائما من ضغط الزر Select All في كل صفحة تعرض العناصر المتاحة للإعداد، لتضمن إعداد كافة مكونات البرنامج.

بعد انتهاء عملية الإعداد، توجه إلى سطح المكتب Desktop واضغط:

Start\Programs\Microsoft SQL Server 2008\SQL Server Management Studio

سيؤدي هذا إلى فتح مدير قواعد بيانات سيكويل سيرفر SSMS، وأول ما سيعرضه لك هو نافذة الاتصال بخادم قاعدة البيانات:

هذه النافذة تتيح لك تحديد ثلاث خيارات:

١- نوع محرك خادم البيانات Server Type:

تعرض لك هذه القائمة المنسدلة نوعين من الخوادم:

أ- Database Engine: وهو النوع الذي سنستخدمه في برامج الويندوز وصفحات المواقع ASP.NET.

ب- SQL Server Compact Edition: وهو النوع المستخدم مع الأجهزة الكفية المحمولة.

٢ – اسم الخادم Server Name:

يمكنك كتابة اسم خادم البيانات في هذه القائمة مباشرة، أو ضغط زر الإسدال لاختيار اسم الخادم، حيث ستعرض لك القائمة المنسدلة أسماء الخوادم التي اخترتها سابقا، بالإضافة إلى العنصر <Browse for more...>، ولو ضغطت هذا العنصر فستظهر لك نافذة اختيار الخادم، وهي تعرض شريطين علويين Tabs 2، أولهما يتيح لك اختيار خادم محلي (موجود على جهازك)، والآخر يتيح لك اختيار خادم موجود على الشبكة Network.. وسنستخدم في هذا الكتاب خادما محليا Local Server، ولو ضغطت العلامة + الموجودة بجوار Database Engine، فستجد تحتها أسماء الخوادم المحلية الموجودة على جهازك.. وعلى الأقل ستجد خادما واحدا، هو خادم سيكويل سيرفر، حيث يتكون اسم هذا الخادم من اسم جهازك وليكن MyPC) يليه اسم خادم سيكويل سيرفر SQLEXPRESS.. اختر هذا الخادم واضغط OK للعودة إلى النافذة الأولى.. ستجد في مربع النص اسم الخادم MYPC\SQLEXPRESS.. لاحظ مجددا أننا سنستخدم في هذا الكتاب الاسم MyPC باعتباره اسم الجهاز.. لكن لو كان اسم جهازك مثلا MohammadPC فستجد في هذه الخانة الاسم: MohammadPC\SQLEXPRESS... وهكذا.

٣- تحقيق الهوية Authentication:

تتيح لك هذه القائمة المنسدلة نوعين من تحقيق الهوية:

أ- تحقيق الهوية بواسطة الويندوز Windows Authentication:

في هذا النوع لا يتم حماية الاتصال بالخادم باسم مستخدم وكلمة سر خاصة، بل تكون الحماية معتمدة على بيانات مستخدم الويندوز.. بمعنى أن مستخدم الويندوز الذي أنشأ الاتصال يستطيع الاتصال بالخادم والتعامل مع قواعد البيانات المتاحة بمجرد تسجيل دخوله على الويندوز.. وهذا هو نوع تحقيق الهوية الذي سنستخدمه مع الخادم المحلي لأنه غير محمي.

ب- تحقيق هوية خادم سيكويل SQL Server Authentication:

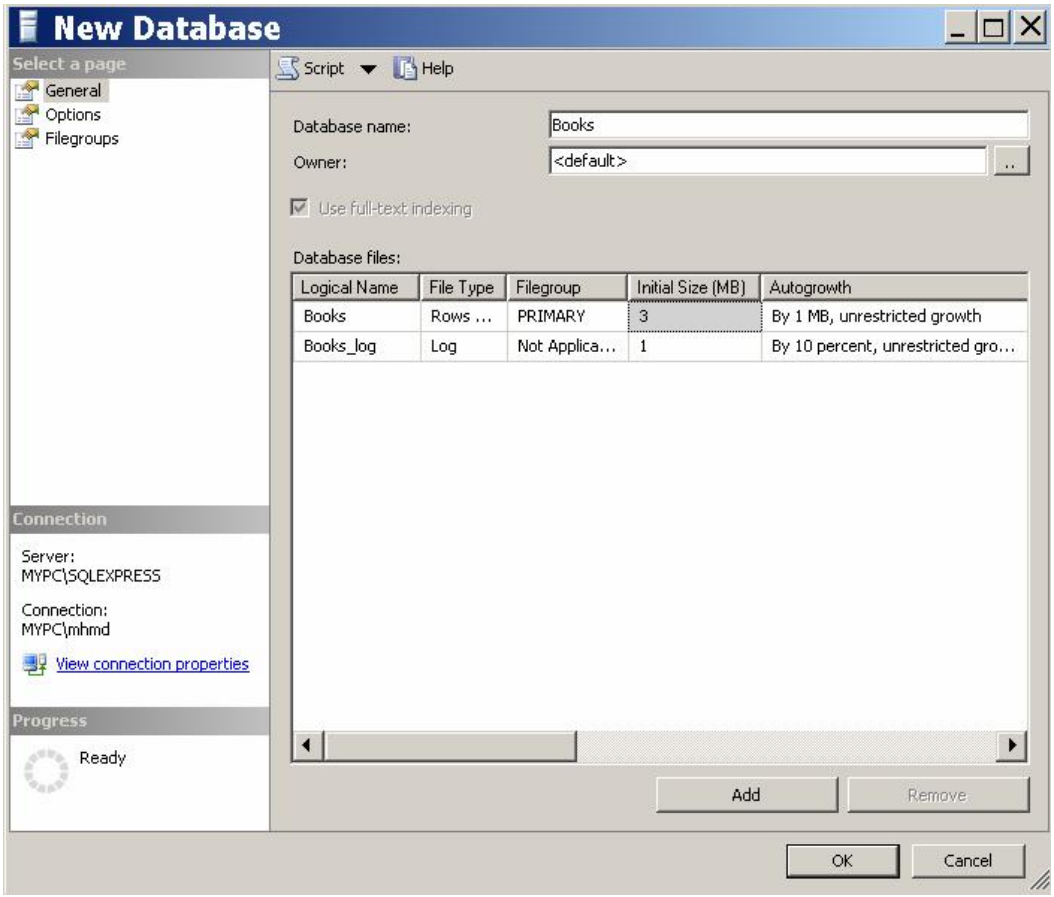
في هذا النوع تتم حماية الاتصال بالخادم وكل قواعد البيانات التي ستنشئها عليه باسم مستخدم Username وكلمة مرور Password، حيث يتم تفعيل مربعي النص الخاصين باسم المستخدم وكلمة المرور.. ويستخدم هذا النوع مع الخوادم الموجودة على الشبكة، لأنها تكون محمية ضد محاولات الاختراق، ويكون لكل مستخدم صلاحيات معينة في التعامل مع خادم البيانات كما يحددها مدير النظام.. فهناك مستخدمون يستطيعون إضافة أو حذف قواعد البيانات أو الجداول والأعمدة والسجلات، بينما تقتصر صلاحيات بعض المستخدمين على قراءة البيانات من بعض الجداول.. لهذا في الغالب لا تسمح الخوادم الحقيقية (غير الخادم المحلي الافتراضي) بالاتصال بها بدون أن يحدد المتصل اسمه وكلمة مروره.

بعد أن تنتهي من إدخال هذه البيانات، اضغط OK لإغلاق النافذة وإنشاء الاتصال.. سيؤدي هذا إلى إضافة اسم الخادم الذي اتصلت به إلى متصفح الكائنات Object Explorer الموجود على يسار نافذة مدير سيكويل سيرفر، ولو أسدلت العناصر الفرعية

لهذا الاتصال، فستجد من ضمنها العنصر Databases، وهو العنصر الذي تظهر تحته كل قواعد البيانات التي تعمل على هذا الخادم، كما يمكنك من خلال هذا العنصر إضافة قواعد بيانات جديدة إلى الخادم، أو تشغيل قواعد بيانات موجودة مسبقا.

إنشاء قاعدة بيانات سيكويل سيرفر:

في متصفح الكائنات Object Explorer، اضغط فرع قواعد البيانات Databases بزر الفأرة الأيمن، ومن القائمة الموضعية Context Menu اضغط الأمر New Database لعرض نافذة إنشاء قاعدة بيانات جديدة.



هذه النافذة تتيح لك إنشاء أكثر من قاعدة بيانات، حيث تحتوي على جدول يعرض تفاصيل ملفات قواعد البيانات التي تريد إنشاءها.. كل ما عليك هو كتابة اسم قاعدة البيانات في الخانة Database Name أعلى النافذة، حيث ستلاحظ ظهور ملفين في الجدول السفلي يحملان اسم هذه القاعدة.. فمثلا، لو كتبت الاسم Books في خانة اسم قاعدة البيانات، فسيعرض الجدول الملفين الآتيين:

١- Books.mdf:

هذا هو الملف الذي يحفظ بيانات قاعدة البيانات، وستجد أن خانة النوع المجاورة لاسمه تشير إلى أنه Rows Data أي أنه يحمل بيانات السجلات.. وستجد أن خانة الحجم المبدئي تشير إلى أن حجم هذا الملف سيكون مبدئيا ٣ ميجا، وسيزيد بمعدل ١ ميجا في كل مرة تتجاوز فيها البيانات حجم الملف.. ويمكنك أن تغير الحجم المبدئي ومعدل الزيادة تبعا لنوع قاعدة البيانات التي تتعامل معها، فلو كنت تتوقع معدلا كبيرا لإضافة البيانات إليها، فالأفضل أن تجعل معدل زيادة حجمها أكبر، لأن هذا يوفر الوقت المستهلك عند تكرار عملية تكبير الملف.

٢- Books_log.ldf:

هذا هو الملف الذي يحفظ سجل أداء قاعدة البيانات Log، وستجد أن خانة النوع المجاورة لاسمه تشير إلى أنه Log.. وستجد أن خانة الحجم المبدئي تشير إلى أن حجم هذا الملف سيكون مبدئيا ١ ميجا، وسيزيد بمعدل ١٠% من حجمه في كل مرة تتجاوز فيها البيانات حجم الملف.. ويمكنك أن تغير الحجم المبدئي ومعدل الزيادة تبعا لحاجتك.

لاحظ أن ملف سجل الأداء Log يخزن كل التفاصيل التي تتم على قاعدة البيانات من إضافة وحذف واستعلامات وغيرها، ومن قام بها ومتى، وهذا يجعل حجمه يتضخم بسرعة كبيرة، لأنه سيحتوي على البيانات الأصلية وكل التعديلات التي تمت عليها.. هذا يحقق الفوائد الآتية:

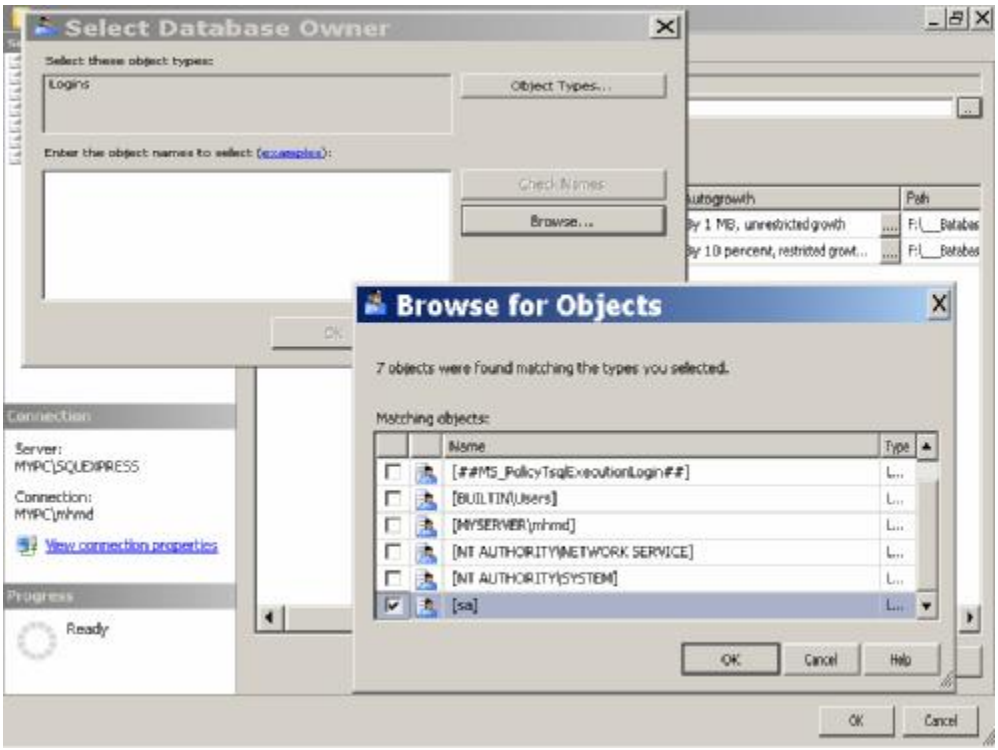
١- يقدم تاريخا مفصلا للعمليات التي تمت على قاعدة البيانات، ويسهل مراقبة أدائها ومعرفة حجم نشاط المستخدمين المتعاملين معها.
٢- يتيح لك استعادة أي وضع سابق لقاعدة البيانات، ليمكنك التراجع عن أية عملية غير مرغوبة لاستعادة حالة قاعدة البيانات قبل إجرائها.
٣- يحمي البيانات من الضياع.. ففي قاعدة بيانات Access يمكن أن يؤدي انقطاع الكهرباء أو حدوث أي مشكلة في الويندوز أثناء حفظ بعض البيانات إلى تدمير ملف قاعدة البيانات كلها، لأن عملية الحفظ تتم في الملف مباشرة.. بينما في سيكويل سيرفر يتم حفظ التغييرات في ملف سجل الأداء Log أولا، وبعد مدة زمنية معينة يتم نسخ كل التغييرات من ملف سجل الأداء إلى ملف قاعدة البيانات.. هذا يعني أنه لو حدثت أي مشكلة أثناء الحفظ في سجل الأداء فإن ملف قاعدة البيانات سيظل سليما، وستكون البيانات المفقودة أقل ما يمكن، حيث سيتم فقد الجزء الذي كان يتم حفظه فقط.. أما لو حدثت مشكلة أثناء نقل البيانات من ملف سجل الأداء إلى ملف قاعدة البيانات وأدت إلى تدمير ملف قاعدة البيانات، فلن تفقد أي بيانات، لأن سجل الأداء ما زال يحتفظ بنسخة منها.

وتعرض الخانة Path مسار حفظ كل ملف من هذين الملفين، وهو مبدئيا:
C:\Program Files\Microsoft SQL Server\MSSQL10.SQLEXPRESS\MSSQL\DATA
ويمكنك تغيير هذا المسار بضغط زر الانتقال الموجود في هذه الخانة، لعرض مربع حوار اختيار مجلد.

تنبيه:
أنصحك بتغيير هذا المسار إلى أي مسار آخر خاص بك على جزء من القرص الصلب غير الذي يوجد عليه نظام الويندوز، حتى لا تفقد قواعد بياناتك في عملية تهيئة Format طائشة، أو بسبب حدوث أي تدمير لنظام التشغيل لأي سبب كان، يجعلك تعيد إعداده دون أن تنتبه إلى أخذ نسخة من ملفاتك الهامة.

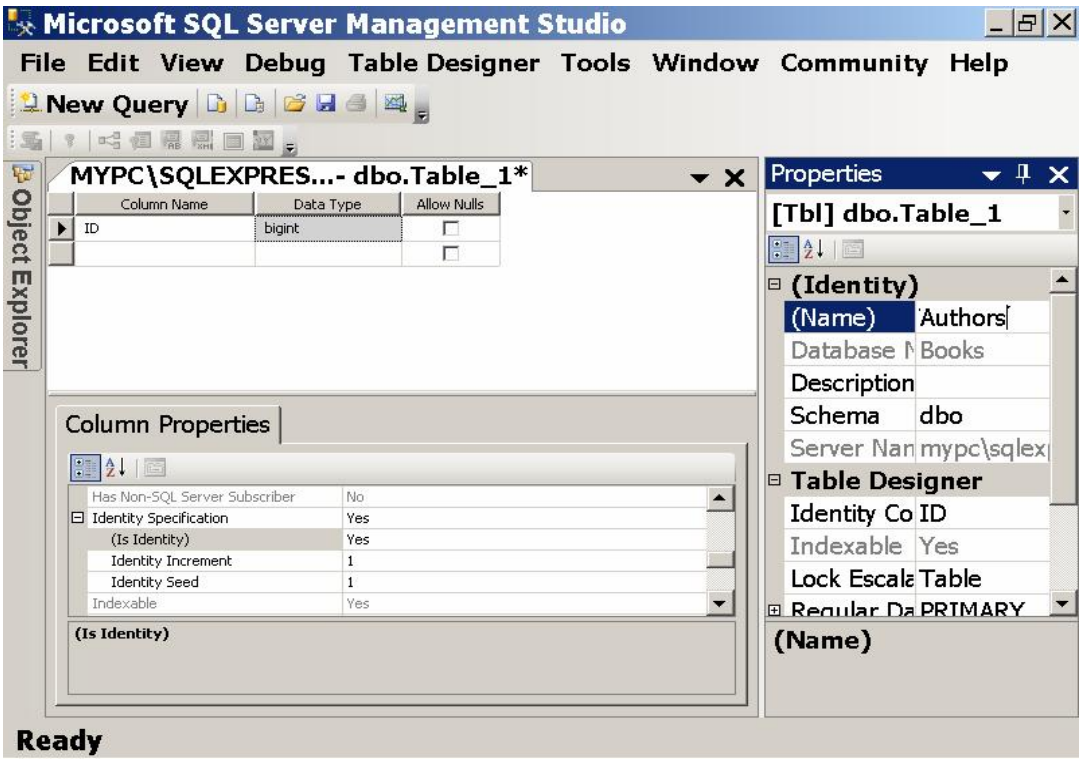
ولإنشاء قاعدة بيانات أخرى، اضغط الزر Add الموجود أسفل النافذة، حيث سيظهر صفان جديدان في الجدول.. حدد أي صف منهما، واكتب اسم قاعدة البيانات الجديدة في الخانة Database Name أعلى النافذة.. وإذا أردت حذف أي قاعدة بيانات، فحدد أيا من ملفيها، واضغط الزر Remove أسفل النافذة.

يتبقى جزء مهم في هذه النافذة، هو خانة مالك قاعدة البيانات Owner.. لو تركت قيمة هذه الخانة <Default> فسيكون اسم المستخدم الخاص بك على الويندوز Username هو مالك هذه القاعدة، لكن هذا قد يحرمك بعض الصلاحيات.. لهذا لو أردت صلاحيات كاملة، فاضغط الزر المجاور لهذه الخانة لعرض مربع حوار اختيار مالك قاعدة البيانات.. اضغط الزر Browse الموجود في النصف السفلي من هذه النافذة، ومن النافذة التي ستظهر اختر [sa] وهو اختصار لتعبير مدير النظام System Administrator واضغط OK لإغلاق النافذتين.



بعد أن تفرغ من إنشاء كل قواعد البيانات التي تريدها، اضغط الزر OK لإغـــلاق النافـــذة وإنشاء الملفات.. غني عن الذكر أن ضغط الزر Cancel سيلغي كل ما فعلتـــه ولـــن يـــتم إنشاء أي قاعدة من التي أدخلت أسماءها، فكل ما تدخله في هذه النافذة هو مجرد تفاصـــيل، ولا تدخل حيز التنفيذ إلا بعد ضغط الزر OK.

الآن، ستجد اسم قاعدة البيانات Books تحت الفرع Databases في متصفح الكائنـــات.. اضغط العلامة + المجاورة لهذا الاسم، لإسدال عناصر قاعدة البيانات.. ستجد ضمن هـــذه العناصر العنصر Tables، وهو العنصر الذي يحتوي على كل جداول قاعـــدة البيانـــات.. اضغط هذا العنصر بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط الأمر New Table لإنشاء جدول جديد.. سيؤدي هذا إلى عرض نافذة تصميم الجدول في مدير سيكويل، كمـــا ستظهر نافذة الخصائص التي تعرض خصائص هذا الجدول.. الجميل في الأمر هو أن بيئة مدير سيكويل تشبه إلى حد بعيد بيئة التطوير المتكاملة IDE الخاصة بـــدوت نـــت، ممـــا سيشعرك بالألفة معها، ويجعل استخدامك لها سهلا وسريعا.



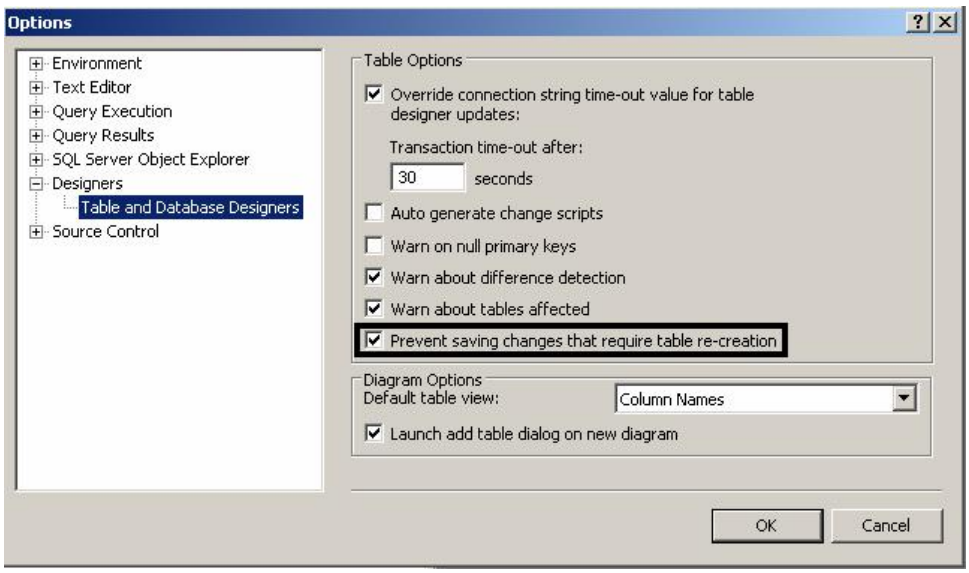
دعنا ننشئ جدول المؤلفين:

أدخل أسماء الأعمدة ونوع كل منها في الجدول العلوي بنفس الطريقة التي استخدمناها في آكسيس، واستخدم الجزء السفلي من النافذة للتحكم في خصائص كل نوع.. وإذا لم تكن تريد السماح للمستخدم بترك أي خانة من خانات هذا العمود فارغة، فأزل علامة الاختيار من مربع الاختيار Allow Nulls.. ولا تنسَ تغيير اسم الجدول من Table1 إلى Authors، ثم اضغط زر الحفظ من شريط الأدوات.

ملحوظة:

بعد أن تقوم بحفظ تصميم الجدول قد يرفض سيكويل سيرفر إجراء أي تعديل على هذا التصميم (كإضافة أو حذف أعمدة)، مع ظهور رسالة تخبرك أن التغييرات التي قمت بها تحتاج إلى إنشاء الجدول من جديد!

لحل هذه المشكلة افتح SQL Server Management Studio، ومن القائمة Tools اضغط الأمر Options، وفي نافذة الخيارات أسدل العنصر Designers من القائمة اليسرى، واضغط العنصر الفرعي Table and Database Designers كما هو موضح في الصورة:



وفي الجزء الأيمن، تأكد من إزالة علامة الاختيار من الخيار:

Prevent Saving Changes that require table re-creation.
فهذا الخيار يمنع حفظ التغييرات التي تسبب إعادة إنشاء الجدول.. اضغط Ok لحفظ التغييرات وإغلاق النافذة.
أما إن استمر هذا الخطأ في الحدوث في بيئة دوت نت، فاستخدم خادم سيكويل لإجراء التعديل الذي تريده في تركيب الجدول.. فإذا أعيتك الحيل، فبإمكانك نسخ تصميم جميع الأعمدة ولصقها في جدول جديد.. لفعل هذا، افتح نافذة تصميم الجدول واضغط هامش العمود الأول بالفأرة، ثم اضغط الزر Shift من لوحة المفاتيح، واضغط هامش العمود الأخير بالفأرة، وبهذا سيتم تحديد كل الأعمدة.. اضغط Ctrl+C لنسخ الأعمدة، ثم أنشئ جدولا جديدا، وفي نافذة التصميم اضغط هامش العمود الفارغ بالفأرة، واضغط Ctrl+V للصق الأعمدة التي نسختها، ثم أجرِ عليها التعديلات التي تريدها، أو أضف إليها أعمدة جديدة.. بعد هذا احذف الجدول القديم، وسمِّ الجدول الجديد باسم الجدول المحذوف.

لو أغلقت النافذة الآن، فسترى اسم جدول المؤلفين أسفل العنصر Tables، وسيكون كالتالي dbo.Authors، حيث إن الحروف dbo هي اختصار للتعبير "مالك قاعدة البيانات" Database Owner، وهو مستخدم افتراضي لا يمكن حذفه.. والسبب في إضافة هذا الجدول إلى المستخدم dbo هو أننا تركنا للخاصية Schema الخاصة بالجدول القيمة dbo (انظر نافذة الخصائص)، ولو غيرناها إلى Guest مثلا، فسيصير اسم الجدول Guest.Authors، لأنه سينتمي إلى المستخدم Guest.. وفي حالات أخرى، يضاف الجدول إلى اسم المستخدم الذي سجل دخوله على قاعدة البيانات، وذلك إذا لم يكن يملك صلاحيات مدير قاعدة البيانات Administrator.

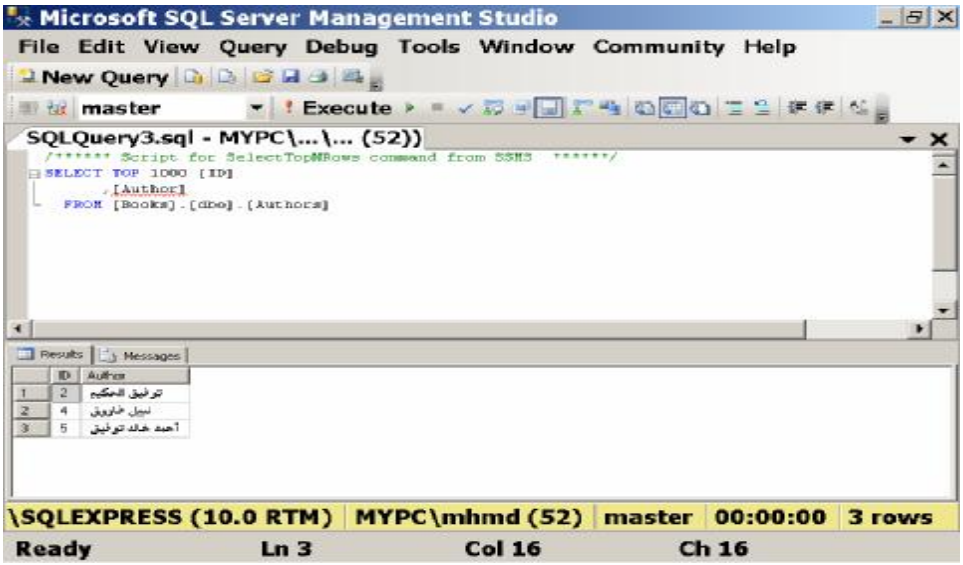
والآن، اضغط العنصر dbo.Authors بزر الفأرة الأيمن.. ستعرض لك القائمة الموضعية مجموعة من الأوامر الهامة، منها:

- **Design:**

يعرض نافذة تصميم أعمدة الجدول.

- **Select Top 1000 rows:**

يعرض أول ١٠٠٠ سجل في الجدول، وفي هذه الحالة لن تتمكن من تعديل أي شيء في هذه السجلات.. لاحظ أن الرقم ١٠٠٠ يهدف إلى تقليل الوقت المستهلك في عرض الجداول الضخمة، لكن لو شئت أن تعرض أي عدد تريده من السجلات وبأي شرط، فإن الجزء العلوي من النافذة يعرض جملة الاستعلام التي تم استخدامها لعرض الألف سجل، ومن ثم يمكنك تعديلها كما تشاء، ثم تضغط زر الفأرة الأيمن، ومن القائمة الموضعية تضغط الأمر Execute لتنفيذ الاستعلام وعرض ناتجه في الجدول السفلي.. وسنتعلم في فصل لاحق كيف نتعامل مع جمل الاستعلام SQL Queries.



- **Edit Top 200 Rows:**

يؤدي هذا الأمر إلى عرض نافذة تحرير السجلات، وبها أول ٢٠٠ سجل إن وجدت، حيث يمكنك تعديل بيانات السجلات الموجودة أو حذفها، كما يمكنك إضافة سجلات جديدة.

- **Rename:**

يؤدي إلى تحول اللافتة التي تعرض اسم الجدول إلى مربع نص، ليمكنك تغيير اسمه.. لاحظ أنك تستطيع فعل نفس الشيء مباشرة، بمجرد ضغط لافتة اسم الجدول ضغطة واحدة بالفأرة بعد تحديده، تماما كما تفعل مع أسماء الملفات في متصفح الويندوز.

- **Delete:**

يؤدي إلى حذف الجدول من قاعدة البيانات، حيث ستظهر لك شاشة تعرض اسم الجدول لتأكيد أمر حذفه، فإن كنت مصرا فاضغط OK، وإن رجعت عن عزمك فاضغط Cancel.. وطبعا يمكنك فعل نفس الشيء بتحديد اسم الجدول وضغط الزر Delete من لوحة المفاتيح مباشرة.

- **Properties:**

تعرض نافذة بها خصائص الجدول والكثير من تفاصيله.

والآن، بعد أن انتهينا من جدول المؤلفين، يمكنك اتباع نفس الطريقة لإنشاء باقي جداول قاعدة الكتب، لتكون كتلك التي أنشأناها في آكسيس.. وستجد قاعدة بيانات الكتب Books.mdf على القرص الضوئي المرفق بهذا الكتاب.

أنواع البيانات في سيكويل سيرفر:

الجدول التالي يلخص لك أهم أنواع البيانات الخاصة بالأعمدة التي تنشئها في سيكويل سيرفر:

النوع	الحجم (Bytes)	الشرح
bigint	٨	عدد صحيح كبير، وهو يقبل أرقاما من -٦٣٨٢ إلى ٦٣٨٢ -١ (وهو رقم ضخم جدا يساوي تقريبا ١٠ مليار مليار).
int	٤	عدد صحيح، وهو يقبل أرقاما من -٣١٨٢ إلى ٣١٨٢ -١ (وهو رقم يساوي تقريبا ٢ مليار).
smallint	٢	عدد صحيح صغير، وهو يقبل أرقاما من -١٥٨٢ إلى ١٥٨٢ -١ (وهو رقم يساوي ٣٢،٧٦٧).
tinyint	١	وحدة ثنائية، وهي تقبل أرقاما من ٠ إلى ٢٥٥.
bit		خانة ثنائية واحدة، تقبل صفرا أو ١، وهي مفيدة عند التعامل مع قيم منطقية (Yes أم No) أو (True أم False).
decimal(x, y) أو **numeric(x, y)**	١٧	عدد عشري، يتيح لك تحديد الدقة العشريّة التي تريد التعامل معها، حيث x تمثل عدد أرقام العدد (بحد أقصى ٣٨ رقما)، و y تمثل عدد الخانات العشرية في هذا العدد.

money	٨	قيمة نقدية، تتيح لك التعامل مع كسور تصل دقتها العشرية إلى ٤ خانات عشرية، وتنحصر القيم المقبولة بين -٩٢٢٣٣٧٢٠٣٦٨٥،٤٧٧,٥٨٠٨ و +٩٢٢٣٣٧٢٠٣٦٨٥٤٧٧,٥٨٠٧.
smallmoney	٤	قيمة نقدية، تتيح لك إدخال كسور تصل دقتها العشرية إلى ٤ خانات عشرية، وتنحصر القيم المقبولة بين -٢١٤٧٤٨,٣٦٤٨ و +٢١٤٧٤٨,٣٦٤٧.
float	٤ أو ٨	عدد عشري مقرب، تتحدد مساحة تخزينه تبعا لعدد الخانات العشرية المطلوبة.. فلو اخترت أن تكون دقته العشرية ٧ خانات، فسيتم حفظه في ٤ وحدات 4 Bytes، أما لو اخترت أن تكون دقته العشرية ١٥ خانة، فسيتم حفظه في ٨ وحدات 8 Bytes.
real	٤	عدد حقيقي، وهو حالة خاصة من النوع float، فأقصى دقة عشرية له هي ٧ خانات.
datetime	٨	تاريخ ووقت تصل دقته إلى الميلي ثانية ممثلة في ٣ خانات عشرية.. مثال: '01/01/1998 12:59:59.999' ولا يقبل هذا النوع إلا التواريخ التي تقع بين ١٧٥٣/٠١/٠١ و ٩٩٩٩/١٢/٣١.
datetime2(n)	من ٦ إلى ٨	تطوير للنوع datetime، يتيح لك التعامل مع تواريخ من ١/١/١ إلى ٩٩٩٩/١٢/٣١.. ويمثل الرقم n عدد الخانات العشرية التي تريد استخدامها مع أجزاء الثانية.. فمثلا لو أردت التعامل مع أجزاء من مئة ألف جزء من الثانية، فاجعل قيمة n تساوي ٥.

smalldatetime	٤	تاريخ ووقت تصل دقته إلى الدقيقة فقط.. مثل: 01/01/1998 12:30
date	٣	تاريخ فقط بدون أي وقت، مثل: 01/01/1998
time(n)	من ٣ إلى ٥	وقت بدون تاريخ.. ويمثل الرقم n عدد الخانات العشرية المستخدمة لعرض أجزاء الثانية.
datetimeoffset(n)	من ٨ إلى ١٠	تاريخ ووقت مرفق به فارق التوقيت عن التوقيت العالمي.. مثال: 01/01/1998 12:35:29.123 +2:15 هذا المثال يوضح أن هذا التاريخ مرتبط بموقع جغرافي يزيد بساعتين وربع عن توقيت خط جرينتش. ويمثل الرقم n عدد الخانات العشرية التي تريد استخدامها مع أجزاء الثانية.
char(n)	n	نص ثابت الطول يتكون من عدد n من الحروف.. لاحظ أن هذا النوع لا يدعم الكود الموسع Unicode، لهذا ستكون الحروف المسموح بكتابتها هي فقط الحروف المتاحة على الخادم Server الذي توجد عليه قاعدة البيانات.. انتبه لهذا جيدا، لأنك في المشاريع الحقيقية قد تضع قاعدة البيانات على خادم في دولة أجنبية، وبالتالي لن يدعم اللغة العربية. ويمكنك التحكم في عدد الحروف، بتغيير قيمة الخاصية Length من الجزء السفلي من نافذة تصميم الجدول، مع ملاحظة أن الحد الأقصى لطول هذا النص هو ٨٠٠٠ حرف.. والقيمة الافتراضية لطول النص هي ١٠ حروف، وفي هذه الحالة لو حاولت كتابة نص طوله ١١ حرفا فسيرفض سيكويل سيرفر ذلك، بينما لو كتبت نصا طوله ٨ حروف فسيتم إكماله إلى ١٠ حروف بإضافة مسافتين في نهايته.

varchar(n)	n بحد أقصى	مشابه للنوع char في كل شيء إلا في أمر واحد، وهو أنه يمثل نصا مرنا متغير الطول لا يقبل الحروف الموسعة، وأقصى عدد يقبله من الحروف هو n (الحد الأقصى لقيمة n هو ٨٠٠٠ حرف).. والقيمة الافتراضية لطول النص في هذا النوع هي ٥٠ حرفا، وفي هذه الحالة لو حاولت كتابة نص طوله ٥١ حرفا فسيرفض سيكويل سيرفر ذلك، بينما لو كتبت نصا طوله ٨ حروف فسيتم حفظه فعلا في ٨ وحدات، ولن يتم إكماله بمسافات.. واضح أن هذا يوفر مساحة التخزين، ويعطيك مرونة في حالة تفاوت أطوال النصوص التي ستدخلها في العمود.
varchar(MAX)		هذه الصيغة حالة خاصة من النوع السابق، وهي تجعل الحقل يستوعب نصا ضخما يصل إلى حجمه إلى ٢^٣١ -١ حرفا.. وفي هذه الحالة يكون حجم الحقل = عدد حروف النص المكتوب فيه + ٢. ولا يمكن استخدام عمود من هذا النوع كمفتاح Key للجدول أو كمفتاح في أحد الفهارس Indexes.
text	٢ مليار بحد أقصى	مشابه للنوع varchar في كل شيء، إلا أنه يسمح بحفظ نص مرن متغير الطول يمكن أن يصل إلى حوالي ٢ مليار حرف (٢^٣١ -١ حرف).. وأيضا لا يقبل هذا النوع حروف الكود الموسع Unicode.

nchar(n)	٢×n	نص ثابت الطول طوله n يدعم الحروف الموسعة Unicode، لهذا يتم حفظ كل حرف في ٢ وحدة 2 Byte، ولهذا فإن أقصى عدد مسموح به من الحروف في هذا النوع هو ٤٠٠٠ حرف.. هذا النوع مناسب للغة العربية.
nvarchar(n)	٢×n +٢ بحد أقصى	نص مرن متغير الطول يدعم الحروف الموسعة Unicode، بحد أقصى ٤٠٠٠ حرف.. هذا النوع مناسب للغة العربية.
nvarchar(MAX)		هذه الصيغة حالة خاصة من النوع السابق، وهي تجعل الحقل يستوعب نصا ضخما يصل إلى حجمه إلى ٢^٣١ -١ وحدة ثنائية.. وفي هذه الحالة يكون حجم الحقل = ٢× عدد حروف النص المكتوب فيه + ٢.
ntext	١ مليار بحد أقصى	نص مرن متغير الطول يدعم الحروف الموسعة Unicode، بحد أقصى حوالي مليار حرف (٢^٣٠ - ١ حرف).. هذا النوع مناسب للغة العربية.
binary(n)	٤+ n	بيانات ثنائية ثابتة الطول، وبشرط ألا تزيد قيمة n عن ٨٠٠٠ وحدة ثنائية Bytes.. لو اخترت n = ٥٠ مثلا، وحفظت في إحدى الخانات بيانات تقل عن ٥٠ وحدة، فسيتم إكمال الباقي بالعدم Null (الحرف رقم ٠ في ترميز ASCII).

varbinary(n)	n +٤ كحد أقصى	بيانات ثنائية مرنة متغيرة الطول، وبحد أقصى ٨٠٠٠ وحدة ثنائية.. ولو حفظت بيانات طولها أقل من n، لا يتم إكمال الطول الناقص بأي شيء.
Varbinary (MAX)		هذه الصيغة حالة خاصة من النوع السابق، وهي تجعل الحقل يستوعب بيانات يصل حجمها إلى ٢^٣١ -١ وحدة ثنائية.. وفي هذه الحالة يكون حجم الحقل = عدد الوحدات المكتوبة فيه + ٢.
image	٢ مليار بحد أقصى	بيانات ثنائية مرنة متغيرة الطول، يمكن أن تصل إلى حوالي ٢ مليار وحدة ثنائية (٢^٣١ -١ Bytes).
sql_variant	٨٠١٦ بحد أقصى	متغير عام يقبل باقي أنواع البيانات (الأرقام والنصوص)، ولا يستثنى من هذا إلا الأنواع التالية: text, ntext, timestamp, varchar(MAX)
uniqueidentifier	١٦	معرف عام متفرد GUID وهو رقم سداسي عشري Hexadecimal غير قابل للتكرار.
xml	٢ مليار بحد أقصى	يتيح لك هذا النوع تخزين صفحات XML أو أجزاء منها في خانات العمود.
Timestamp	٨	مصفوفة من ٨ خانات ثنائية، تقوم بإنتاج أعداد ثنائية Bytes، بحيث تكون المصفوفة متفردة عبر كل الجدول، وذلك لاستخدامها في تمييز النسخ المحدّثة من كل صف Version-Stamping.
Udt		نوع يعرفه المستخدم User-Defined Type.
Structured		نوع خاص للبيانات المركبة في صورة جدول Table-valued، وسنتعرف عليه لاحقا.

ملحوظة ١:

عند استخدام أكثر من عمود متغير القيمة (ينتهي بقوسين بينهما طوله (n))، يجب ألا يزيد الطول الإجمالي للسجل ككل عن ٨٠٠٠ وحدة ثنائية Byte، وإلا حدث خطأ.. ولتجاوز هذا الحجم المحدود للسجل، عليك استخدام المتغيرات Text وnText و image أو المتغيرات الجديدة التي ظهرت في سيكويل سيرفر ٢٠٠٥، وهي تحمل نفس أسماء المتغيرات محددة الطول، لكنها تنتهي بالقيمة القصوى (Max)، وبذلك تتيح لك التعامل مع بيانات يتجاوز طولها ٢ مليار وحدة ثنائية Byte.. كما تتيح لك هذه المتغيرات إمكانيات جديدة في كتابة وقراءة البيانات بطريقة تتابعية Sequential، كما سنرى لاحقا.

لاحظ أيضا أن حفظ أي حروف عربية في حقول من أنواع لا تبدأ بالحرف n سيجعلك تفقدها، حيث ستتحول إلى علامات استفهام.

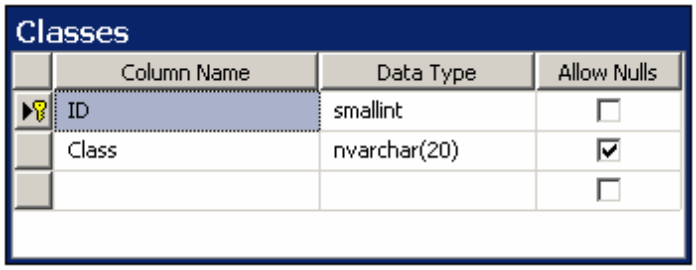
ملحوظة٢:

لجعل أحد الحقول ترقيما تلقائيا، اختر له نوعا رقميا مناسبا، وفي الجزء السفلي من النافذة اضغط العلامة + المجاورة للعنصر Identity Specification لإسدال خصائصه، واجعل للخاصية (Is Identity) القيمة Yes لجعل الحقل معرفا متفردا.. هذا سيغير قيمة كل من الخاصيتين "معدل الزيادة" Identity Increment وبذرة الزيادة Identity Seed إلى القيمة ١.. ولو أردت أن تبدأ الترقيم التلقائي من العدد ١٠٠ مثلا، فضع في الخاصية Identity Seed القيمة ١٠٠.. ولو أردت أن يزيد ترقيم كل حقل عن السابق له بمقدار ٣، فضع في الخاصية Identity Increment القيمة ٣.

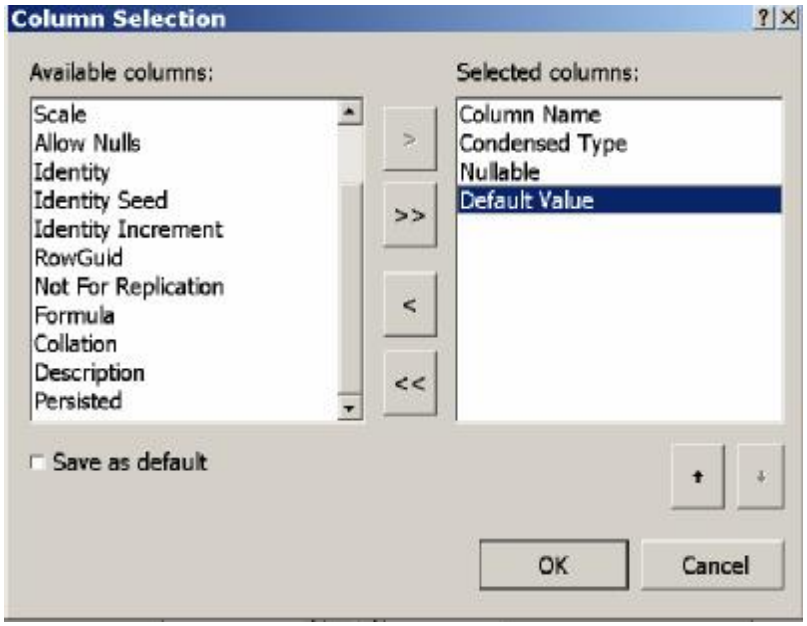
ولجعل الحقل مفتاحا أساسيا، اضغط الهامش الأيسر المجاور له بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط الأمر et Primary Key.. ولو أردت أن تتراجع عن جعل الحقل مفتاحا أساسيا، فاضغطه بزر الفأرة الأيمن مجددا، ومن القائمة الموضعية اضغط الأمر Remove Primary Key.

إنشاء العلاقات في سيكويل سيرفر:

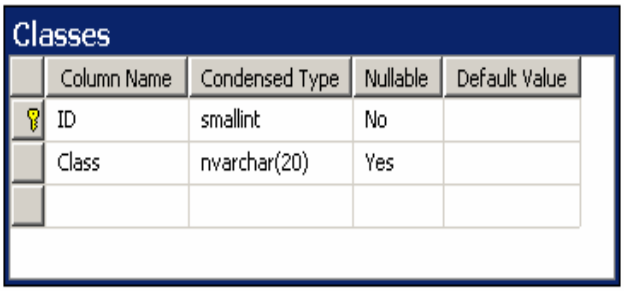
في متصفح الكائنات Object Explorer أسدل عناصر قاعدة بيانات الكتب Books، وبزر الفأرة الأيمن اضغط "مخططات قاعدة البيانات" Database Diagrams، ومن القائمة الموضعية اضغط الأمر New Database Diagram.. سيؤدي هذا إلى عرض نافذة اختيار الجداول التي ستستخدمها في رسم المخطط.. حدد جدولا أو أكثر بالفأرة، ثم اضغط الزر Add لإضافتها إلى صفحة المخطط، ثم اضغط Close لإغلاق النافذة والانتقال إلى صفحة المخطط.. هذه النافذة تشبه نافذة إنشاء العلاقات في آكسيس، ولكنها تمنحك ميزات إضافية، فبإمكانك تغيير أسماء أعمدة أي جدول بمجرد ضغطها وتحريرها في هذا المخطط، كما يمكنك ضغط أي حقل بزر الفأرة الأيمن وضغط الأمر Set Primary Key من القائمة الموضعية لجعل هذا الحقل مفتاحا أساسيا.



وفي الوضع الطبيعي، يعرض المستطيل الممثل لكل جدول أسماء أعمدته، لكنه تستطيع أن تعرض أيضا تفاصيل كل عمود (اسمه، ونوع بياناته، والسماح بتركه فارغا)، وذلك بضغط مستطيل الجدول بزر الفأرة الأيمن لعرض القائمة الموضعية، ثم ضغط القائمة الفرعية View، Table وضغط الأمر Slandered.. كما يمكنك اختيار طرق

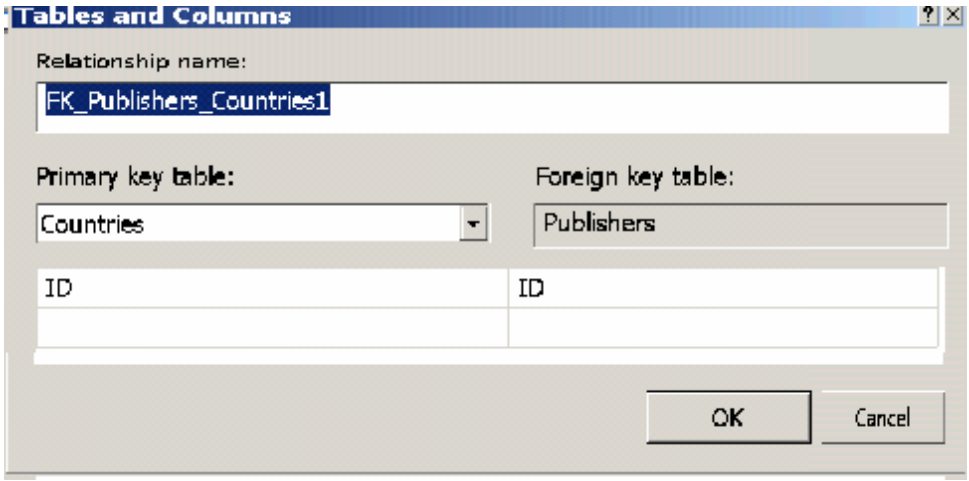


عرض أخرى، كأن تختار عرض المفاتيح الأساسية الموجودة في الجدول فقط Keys، أو اختيار عرض اسم الجدول فقط Name Only، بل يمكنك تصميم طريقة عرض خاصة بك Custom.. مبدئيا ستعرض هذه الطريقة تفاصيل الجدول الأساسية، ولكن يمكنك إضافة أية تفاصيل أخرى بضغط الزر Modify Custom، واستخدام النافذة التي ستظهر لنقل التفاصيل التي تريد عرضها من القائمة اليسرى إلى القائمة اليمنى، وذلك بضغط الزر > .

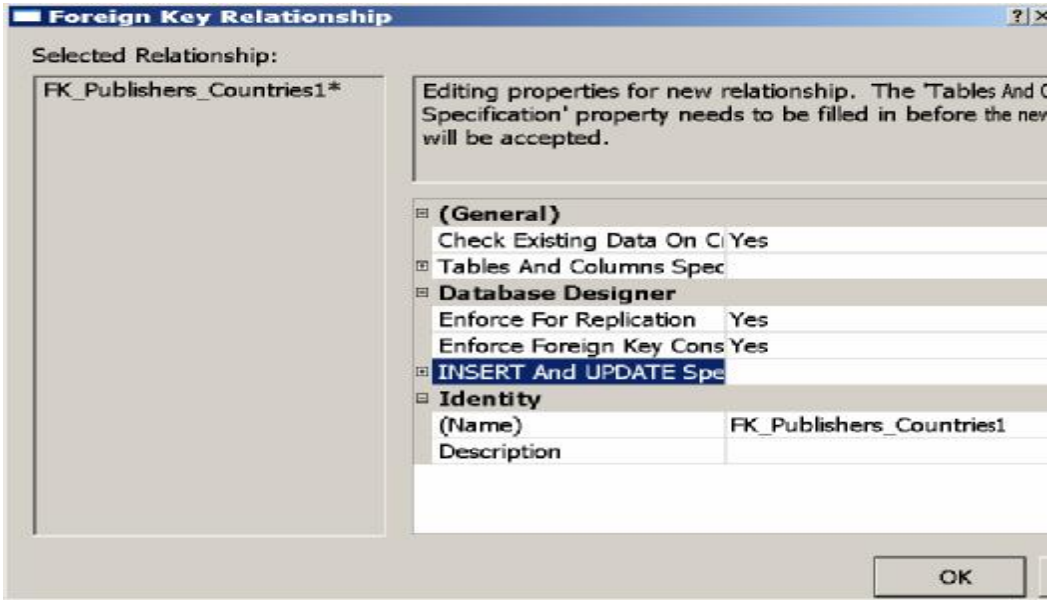


وكما ترى في الصورة أعلاه، فقد أضفنا عمود القيمة الافتراضية للعمود، وهذا سيؤدي إلى عرض مخطط الجدول كما في الصورة.. ولو أردت العودة إلى طريقة العرض التقليدية، فاضغط الأمر Column Names لعرض أسماء الأعمدة فقط.

وكما في آكسيس، يمكنك سحب أي حقل من جدول وإلقاؤه على أي حقل في جدول آخر لتكوين علاقة بين الجدولين.. سيؤدي هذا إلى ظهور نافذة توضح الجدولين والعمودين الداخلين في العلاقة، لتتيح لك تغييرهما إن أردت:



وبعد أن تضغط OK ستظهر نافذة توضح اسم وخصائص العلاقة بين الجدولين:



في هذه النافذة يمكنك إدخال اسم العلاقة، وتحديد مواصفاتها، بطريقة مماثلة للتي اتبعناها في Access.. لاحظ أن اسم العلاقة الافتراضي يكون على الصيغة:

FK_T1_T2n، حيث FK اختصار "المفتاح الفرعي" Foreign Key، و T1 هو اسم الجدول الأول (الذي يحتوي على المفتاح الأساسي)، و T2 هو اسم الجدول الثاني (الذي يحتوي على المفتاح الفرعي)، و n هو رقم العلاقة، وهو يبدأ بالرقم ١، ويمكنك حذفه لو كانت هناك علاقة واحدة فقط بين الجدولين.. وعلى كل حال، يمكنك تغيير هذا الاسم إلى أي اسم آخر يناسبك، بتغيير الخاصية (Name) الموجودة أسفل النافذة.. كما تحتوي هذه النافذة على الخصائص التالية:

- **Check existing data on creation or re-enabling:**

إذا جعلت قيمة هذه الخاصية Yes (وهي القيمة الافتراضية)، فسيتم التحقّق من صحّة البيانات الموجودة حاليا في الجدولين، للتأكد من أنها تتفق مع شروط العلاقة التي سيتم إنشاؤها.

- **Enforce for replication:**

إذا جعلت قيمة هذه الخاصية Yes (وهي القيمة الافتراضية)، فسيتم إنشاء العلاقة في قاعدة البيانات الجديدة عند نسخ قاعدة البيانات الحالية.

- **Enforce foreign key constraint:**

إذا جعلت قيمة هذه الخاصية Yes (وهي القيمة الافتراضية)، فسيتم التحقق من قيد المفتاح الفرعي، وهو يتأكد من عدم وجود بيانات في الجدول الفرعي مرتبطة بمفتاح غير موجود في الجدول الأساسي.. بعبارة أخرى: هذا القيد يدفع قاعدة البيانات لتحقيق التكامل المرجعي Referential Integrity بين الجدولين، لهذا لو حاولت حذف سجل من الجدول الأساسي وكان مرتبطا ببيانات في الجدول الفرعي، فسيحدث خطأ ولن يتم تنفيذ هذه العملية.. ولو أردت أن تنفذ عملية الحذف، فعليك أولا أن تحذف السجلات ذات الصلة من الجدول الفرعي، ثم تحذف بعدها السجل الأساسي المرتبط بها من الجدول الأساسي.

- **INSERTs and UPDATEs Specifications:**

اضغط العلامة + المجاورة لهذه الخاصية، لتظهر لك الخاصيتان التاليتان:

- **قاعدة الحذف Delete Rule:**

تتحكم هذه الخاصية في ماذا يحدث عندما يتم حذف بعض السجلات الداخلة في العلاقة.

- **قاعدة التحديث Update Rule:**

تتحكم هذه الخاصية في ماذا يحدث عندما يتم تحديث بيانات بعض السجلات الداخلة في العلاقة.

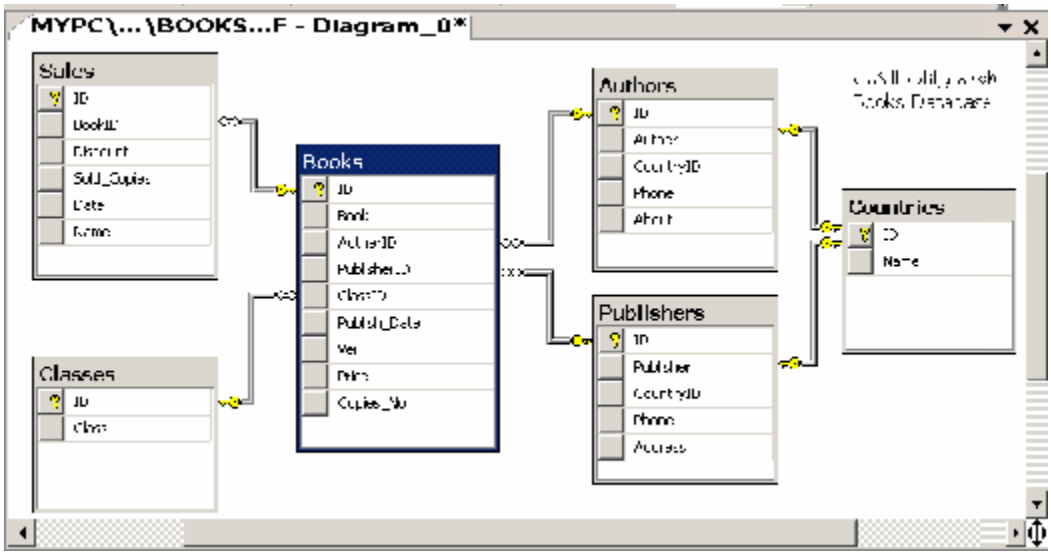
والجدول التالي يوضح القيم المختلفة لهاتين الخاصيتين:

No Action	عدم اتخاذ أي إجراء.. هذه هي القيمة الافتراضية في النافذة.
Cascade	عند تحديث المفتاح الأساسي، يتم تلقائيا تحديث خانة المفتاح الفرعي في السجلات المرتبطة به في الجدول الفرعي. وعند حذف سجل المفتاح الأساسي، يتم تلقائيا حذف جميع السجلات المرتبطة به في الجدول الفرعي.
Set Null	عند حذف أو تحديث المفتاح الأساسي، يتم وضع قيمة منعدمة DBNull في خانة المفتاح الفرعي في السجلات المرتبطة به في الجدول الفرعي.
Set Default	عند حذف أو تحديث المفتاح الأساسي، يتم وضع القيمة الافتراضية في خانة المفتاح الفرعي في السجلات المرتبطة به في الجدول الفرعي.

اضغط OK لإغلاق النافذة، حيث سيظهر خط بين الجدولين، يمثل هذه العلاقة.

ملاحظة:
لكي تنشئ علاقة بين حقلين، يجب أن يكون لهما نفس نوع البيانات ونفس الطول، وإلا فسيرفض سيكويل سيرفر إنشاء العلاقة.. لهذا تأكد من أن للمفاتيح الأساسية والمفاتيح الفرعية نفس النوع لتصلح للدخول في العلاقات.

وتستطيع تغيير شكل خط العلاقة بمرونة، بمجرد سحب أي جزء منه بالفأرة، كما أن طرف بدايته يأخذ شكل مفتاح للإشارة إلى حقل المفتاح الأساسي، بينما يأخذ طرفه الآخر شكل علامة "ما لا نهاية" للإشارة إلى علاقة "واحد بمتعدد".



ولمراجعة أو تعديل خصائص العلاقة، اضغط خطّ العلاقة بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Properties.. ستظهر نافذة الخصائص وبها خصائص العلاقة، بحيث يمكنك تعديلها كيفما شئت.. أما لو شئت حذف العلاقة، فاضغط الخط الممثل لها بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط الأمر Delete Relationship From database، وأكد رسالة الحذف الذي ستظهر لك بضغط Yes.

وتتيح لك صفحة المخططات بعض التسهيلات الإضافية:

- إذ يمكنك مثلا أن تغير نسبة العرض Zoom باستخدام القائمة المنسدلة الموجودة على شريط الأدوات، أو باستخدام القائمة الفرعية Zoom التي تعرضها القائمة الموضعية عند الضغط بزر الفأرة الأيمن في أي منطقة خالية من صفحة المخططات.
- كما يمكنك أن تختار أيضا من القائمة الموضعية إضافة تعليق نصي، وذلك بضغط الأمر Add Text Annotation، حيث سيؤدي هذا إلى إدراج مربع نص على صفحة المخطط لتكتب فيه ما تشاء، مع قدرتك على سحب إطاره بالفأرة لتغيير حجمه أو موضعه، ولو ضغطت داخله بزر الفأرة الأيمن، فيمكنك اختيار تغيير الخط من القائمة الموضعية وذلك بضغط الأمر Set Text Annotation Font.

- ويمكنك عرض لافتة على خط كل علاقة توضح اسم العلاقة، وذلك بضغط الأمر Show Relationship Labels.
- كما يمكنك عرض خط أزرق يمثل حدود صفحة الطباعة لكي تضمن عدم تجاوز الجداول له في حالة رغبتك في طباعة المخطط، وذلك بضغط الأمر View Page Breaks.
- وبعد أن تنتهي من تصميم المخطط، استخدم الأمر Copy Diagram to Clipboard لنسخه كصورة إلى لوحة قصاصات الويندوز، ومن ثم يمكنك لصقه في أي برنامج تحرير صور مثل Paint أو حتى Word لتستطيع طباعته أو استخدامه في وثائقك.
- كما يمكنك طباعة المخطط مباشرة باستخدام الأمرين Page Setup و Print من القائمة الرئيسية File.

والآن قم بحفظ هذا المخطّط، وذلك بضغط Ctrl+S حيث ستظهر لك نافذة تسألك عن اسم المخطّط.. سمّه Diagram1 مثلا.. ستظهر لك رسالة تنذرك بأنّ هناك تغييرات ستحدث لبعض الجداول لتمنحك الفرصة للتراجع.. لو أردت ألا تظهر لك هذه الرسالة في كل مرّة، فأزل علامة الاختيار من مربّع الاختيار Warn About Tables Affected.. اضغط Yes لإتمام العمليّة وإغلاق الرسالة.

لاحظ أنك تستطيع إنشاء أكثر من مخطط وحفظها بأسماء مختلفة.. هذا يتيح لك تقسيم العلاقات بين الجداول على أكثر من مخطط، وذلك إذا كانت جداول قاعدة البيانات كثيرة جدا ومن العسير وضعها كلها معا في مخطط واحد.

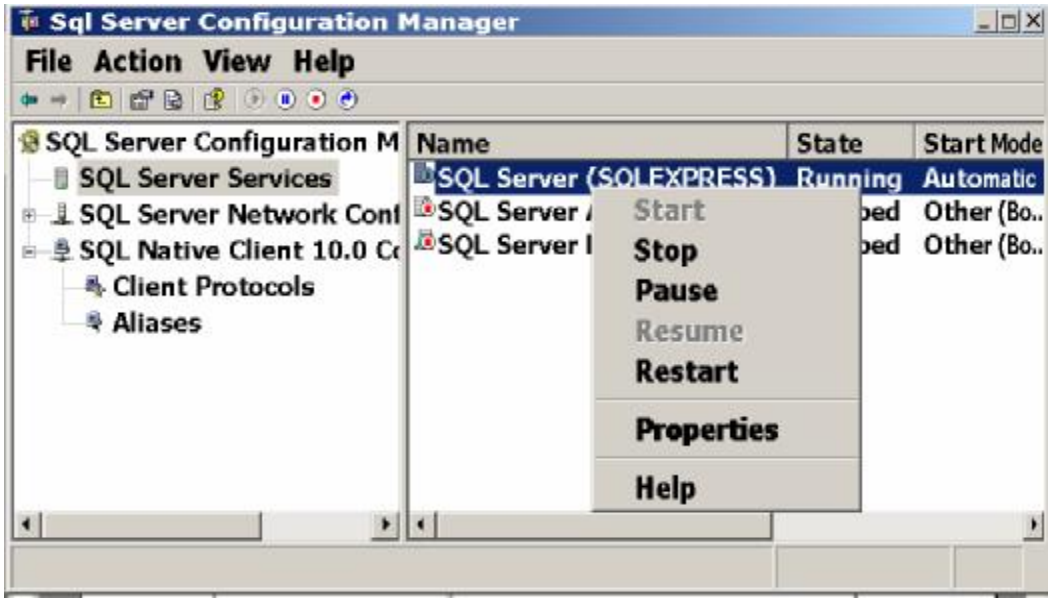
التحكم في خادم سيكويل:

يمكنك إيقاف خادم سيكويل عن العمل وإعادة تشغيله في أي وقت، وذلك باستخدام مدير تهيئة خادم سيكويل SQL Server Configuration Manager.. لفتح هذا البرنامج، اضغط:

Start\Programs\Microsoft SQL Server 2008\

Configuration Tools\ SQL Server Configuration Manager

وفي النافذة التي ستظهر لك، اضغط العنصر SQL Server Services من الشجرة الموجودة على اليسار:



سيعرض هذا ثلاث خدمات من خدمات سيكويل سيرفر، ما يهمنا منها هي خدمة سيكويل سيرفر نفسها (SQLExpress) SQL Server.. لو ضغطت هذه الخدمة بزر الفأرة الأيمن، فستعرض لك القائمة الموضعية الأوامر التي تتيح لك إيقاف Stop أو تشغيل Start أو إعادة تشغيل Restart خادم سيكويل.. وعليك أن تنتبه إلى أن إيقاف هذا الخادم سيمنعك من التعامل مع أي قاعدة بيانات موجودة عليه، لهذا عليك أن تتأكد أنه يعمل قبل أن تحاول الاتصال به من أي برنامج خارجي.. لاحظ أيضا أن حالة الخادم تظل ثابتة

(متوقفا كان أم يعمل) عند إغلاق الحاسب وإعادة تشغيله.. لهذا إن كنت أوقفته على سبيل التجريب، فلا تنسَ إعادة تشغيله مجددا، واتركه يعمل في سلام ☺.

توصيل وفصل قاعدة بيانات:

لفصل قاعدة بيانات الكتب من العمل على الخادم، اضغط اسمها بزر الفأرة الأيمن في متصفح الكائنات Object Explorer، ومن القائمة الموضعية اضغط القائمة الفرعية Tasks، ومنها اضغط الأمر Detach، وعند ظهور نافذة تأكيد الأمر اضغط OK.. قد يحدث خطأ يخبرك بأن هناك عمليات اتصال بقاعدة البيانات حاليا.. في هذه الحالة اضغط علامة الاختيار في مربعي الاختيار المجاورين لاسم القاعدة في النافذة، لإنهاء عمليات الاتصال بها، واضغط زر OK مجددا.

ولإعادة توصيل هذه القاعدة مرة أخرى، اضغط العنصر Databases في متصفح الكائنات بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط الأمر Attach، وفي نافذة توصيل قاعدة البيانات اضغط الزر Add واختر ملف قاعدة البيانات من الموضع الذي حفظتها فيه على جهازك.. ويمكنك أن تضيف أكثر من قاعدة بيانات إلى القائمة لتوصيلها كلها مرة واحدة.. بعد هذا اضغط OK لتوصيل قواعد البيانات التي اخترتها، حيث ستظهر أسماؤها في متصفح الكائنات.

التعامل مع قواعد بيانات سيكويل سيرفر من دوت نت:

تقدم دوت نت ٢٠١٠ دعما كاملا للتعامل مع قواعد بيانات سيكويل سيرفر ٢٠٠٨.. ولا يتوقف الأمر على الاتصال بقاعدة البيانات وقراءة البيانات منها، بل إن الأمر يتعدى ذلك إلى إنشاء قواعد البيانات وتصميم جداولها وعلاقاتها وكل كائناتها من داخل بيئة التطوير المتكاملة IDE الخاصة بدوت نت، بل إنك ستشعر أنه يكاد لا يوجد فارق بين طريقة التعامل مع قاعدة البيانات في دوت نت، وبين التعامل معها في مدير سيكويل سيرفر SQL Server Management Studio.. الفارق الوحيد هو أنك هنا ستتعامل مع متصفح الخوادم Server Explorer بدلا من متصفح الكائنات Object Explorer!

إنشاء قاعدة بيانات سيكويل سيرفر في دوت نت:

افتح متصفح الخوادم Server Explorer من القائمة View، واضغط بزر الفأرة الأيمن العنصر Data Connections، ومن القائمة الموضعيّة اضغط الأمر Create New SQL Server Database.. سيظهر لك مربّع الحوار الموضّح في الصورة التالية:

في هذه النافذة مطلوب منك أن تكتب اسم خادم سيكويل الذي ستنشئ عليه قاعدة البيانات.. اكتب اسم الخادم MYPC\SQLEXPRESS.

ملحوظة:
هناك اختصار متعارف عليه يستخدم بدلا من اسم الخادم المحلي، وهو الكلمة (local) موضوعة بين قوسين، أو النقطة "." .. الحكمة في هذا أنك تستطيع تغيير اسم جهازك من لوحة تحكم الويندوز Control Panel، وفي هذه الحالة ستحدث أخطاء في برامجك التي

تستخدم الاسم القديم، لهذا فمن الأذكى أن تستخدم الاسم SQLEXPRESS\. بدلا من MYPC\SQLEXPRESS .

ولو كان الخادم محميا باسم مستخدم وكلمة سر، فاضغط الاختيار Use SQL Server Authentication، حيث سيتم تفعيل مربعي النص الخاصين بكلمة المرور واسم المستخدم.. ولكن نظرا لأننا نستخدم الخادم المحلي بدون حماية، فاترك الاختيار الأوّل Use Windows NT Integrated Security.

أخيرا، اكتب اسم قاعدة البيانات التي تريد إنشائها، وذلك في مربع النص الموجود أسفل النافذة.. اكتب مثلا الاسم Test واضغط OK لإغلاق النافذة.

ستجد أنّ الاسم mypc\sqlexpress.Test.dbo قد ظهر في متصفح الخوادم.

وسيتمّ إنشاء قاعدة البيانات Test.mdf على المسار:

C:\Program Files\Microsoft SQL Server\
MSSQL10.SQLEXPRESS\MSSQL\DATA

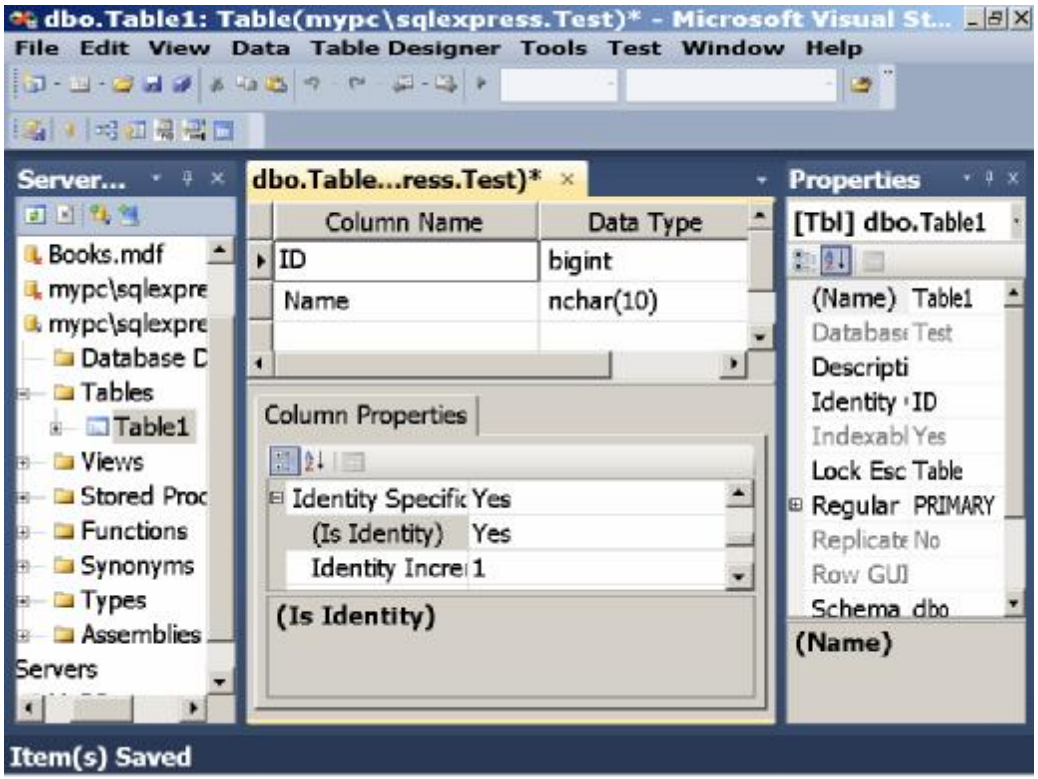
ولو شئت أخذ نسخة احتاطيّةً منها، فلا تنسَ أن تنسخ أيضا الملف Test_log.LDF، ولو رفض الويندوز تنفيذ النسخ، فأغلق البرامج المتصلة بقاعدة البيانات، وإن اضطررت فافصل قاعدة البيانات من العمل على الخادم.

ملحوظة:

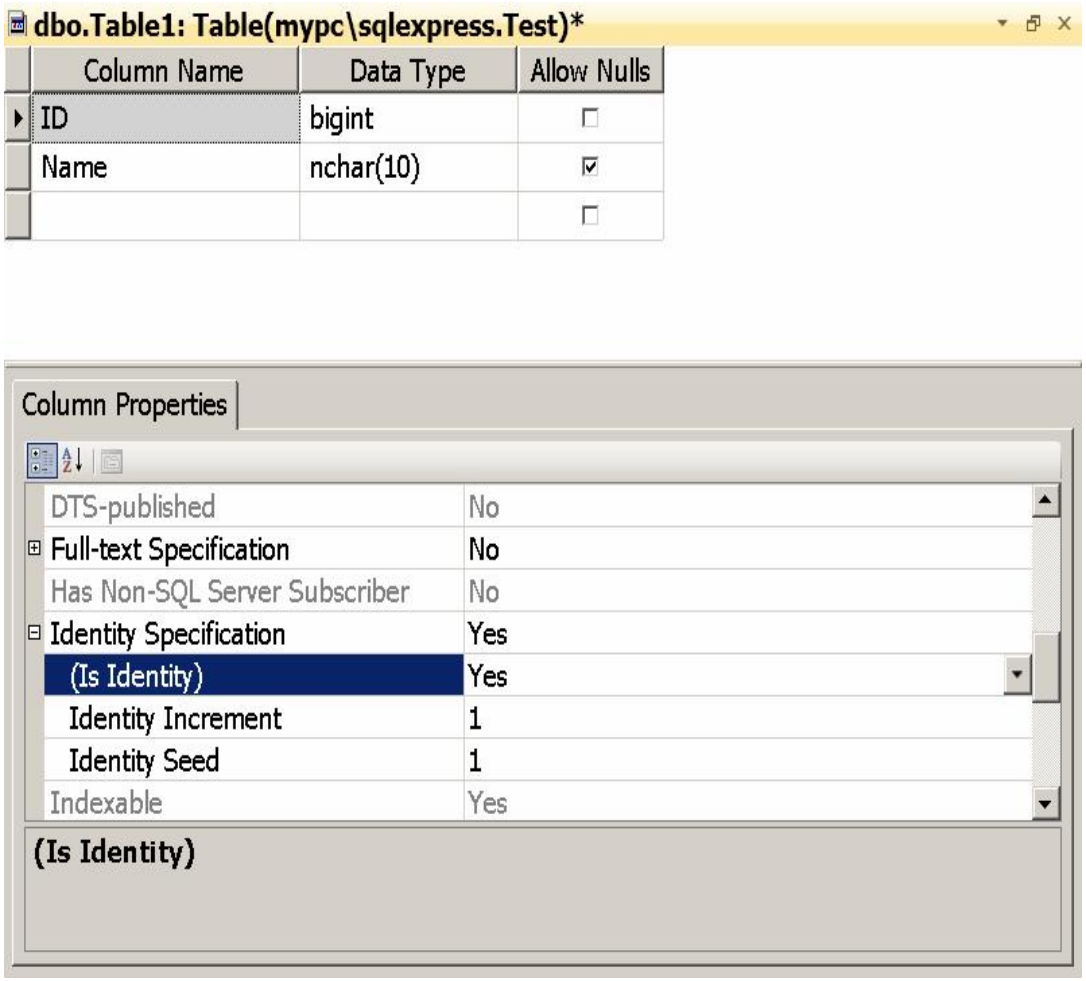
لو ضغطت القاعدة Test في متصفح الخوادم Server Explorer بزر الفأرة الأيمن، وضغطت الأمر Delete من القائمة الموضعيّة، فسيتم حذف الاتصال بقاعدة البيانات من متصفح الخوادم، لكن لن يتم حذف قاعدة البيانات نفسها، ومن ثم يمكنك إعادة الاتصال بها مرة أخرى من جديد بالطريقة التي سنوضحها لاحقا.. لاحظ أن هذا يختلف تماما عما سيحدث في مدير سيكويل SSMS، فضغط الأمر DELETE سيعرض نافذة حذف قاعدة البيانات نفسها، لذا عليك الحذر، استخدام الأمر Tasks\Detach لفصل قاعدة البيانات.

والآن، أسدل عناصر قاعدة البيانات Test في متصفح الخوادم.. لن تشعر بأية غربة، فأنت تشاهد نفس العناصر التي شاهدتها في مدير سيكويل من قبل، ولا تحتاج لأن يعلمك أحد

كيف تنشئ الجداول أو العلاقات، فلا يوجد أي اختلاف يذكر!.. لا أظنك تعتبره اختلافا جذريا، لو أخبرتك أن نقر اسم الجدول مرتين بالفأرة يفتح صفحة تصميمه، أو أن الأمر الذي يفعل هذا من القائمة الموضعية اسمه "فتح تعريف الجدول" Open Table Definition بدلا من الأمر Design في مدير سيكويل، أو أن عرض وتحرير بيانات الجدول يتم باستخدام الأمر Show Table Data بد من الأمرين: Select Top 1000 rows و Edit Top 200 Rows!.. لاحظ أن نافذة عرض بيانات الجدول التي تظهر بعد ضغط الأمر Show Table Data هي جزء النتائج Result Pane في نافذة باني الاستعلام Query Builder، ويمكنك عرض أي جزء آخر بالضغط بزر الفأرة في أي موضع، ومن القائمة الموضعية تضغط القائمة الفرعية Pane وتختار الجزء الذي تريد عرضه.



لاحظ أنك تستطيع تغيير طريقة عرض النوافذ في دوت نت من العرض المـــتلاحم Dock إلى العرض الحر.. مثلا، لو نقرت الشريط العلوي لنافذة تصميم الجدول مـــرتين بالفـــأرة، فستتحول إلى نافذة مستقلة يمكنك تكبيرها وتصغيرها وتحريكها خارج بيئة دوت نـــت كمـــا تريد:

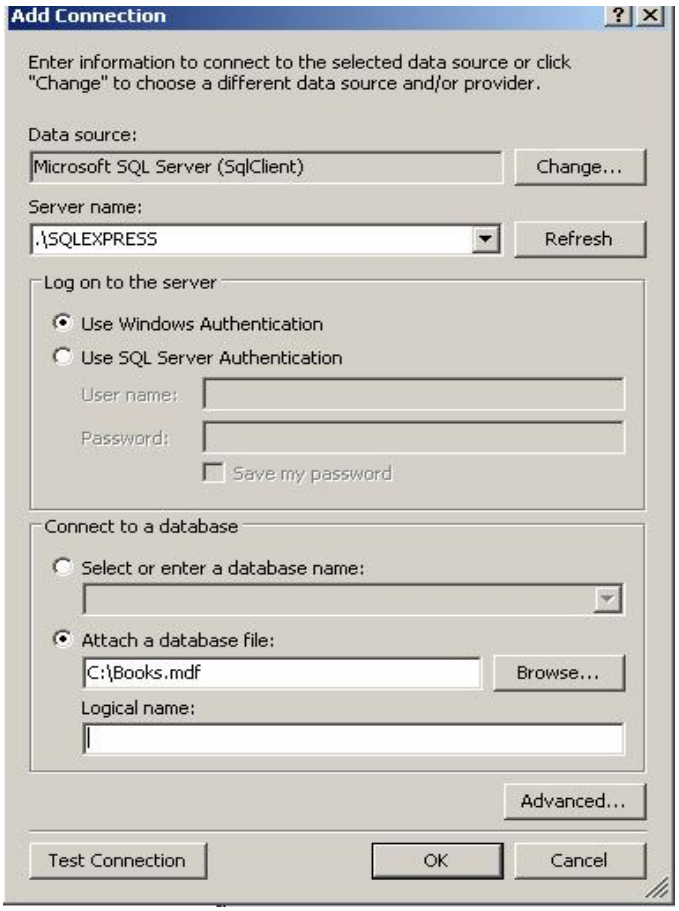


ولو أردت إعادتها إلى وضعها القديم، فاضغط شريط النافذة العلوي بزر الفـــأرة الأيمـــن، ومن القائمة الموضعية اضغط الأمر Dock.

الاتصال بقواعد بيانات SQL Server:

نريد الآن الاتصال بقاعدة البيانات Books.mdf التي أنشأناها بمدير سيكويل.. في هذه الحالة سنتبع نفس الخطوات التي اتبعناها عندما أردنا الاتصال بقاعدة البيانات Books.mdb التي أنشأناها في Access:

- اضغط بزر الفأرة الأيمن العنصر Data Connections في متصفح الخوادم، ومن القائمة الموضعيّة اضغط الأمر "إضافة اتصال" Add Connection.
- في مربّع حوار "اختيار مصدر البيانات" Choose data source، اختر العنصر: Microsoft SQL Server Database File واضغط OK.
- ستظهر لك النافذة الموضحة في الصورة:



في النصف العلوي من النافذة، اكتب اسم خادم سيكويل واختر طريقة تحقيق الهوية.

وفي النصف السفلي من النافذة، اختر قاعدة البيانات التي تريد التعامل معها.. إذا كانت هذه القاعدة تعمل على الخادم فعلا في هذه اللحظة، فاختر الخيار Select or enter a database file واكتب اسم قاعدة البيانات، أو اختره من القائمة المنسدلة.. أما إذا لم تكن قاعدة البيانات متصلة بالخادم في هذه اللحظة، فاختر Attach a database file، واضغط الزر Browse، واختر ملف قاعدة البيانات (وليكن Books.mdf) من المجلد الذي حفظته فيه على جهازك.. ويمكنك أن تكتب في الخانة Logical Name أي اسم تريده لتسمية الاتصال به، وليس شرطا أن يكون له نفس اسم قاعدة البيانات.. أو يمكنك ترك هذه الخانة فارغة.

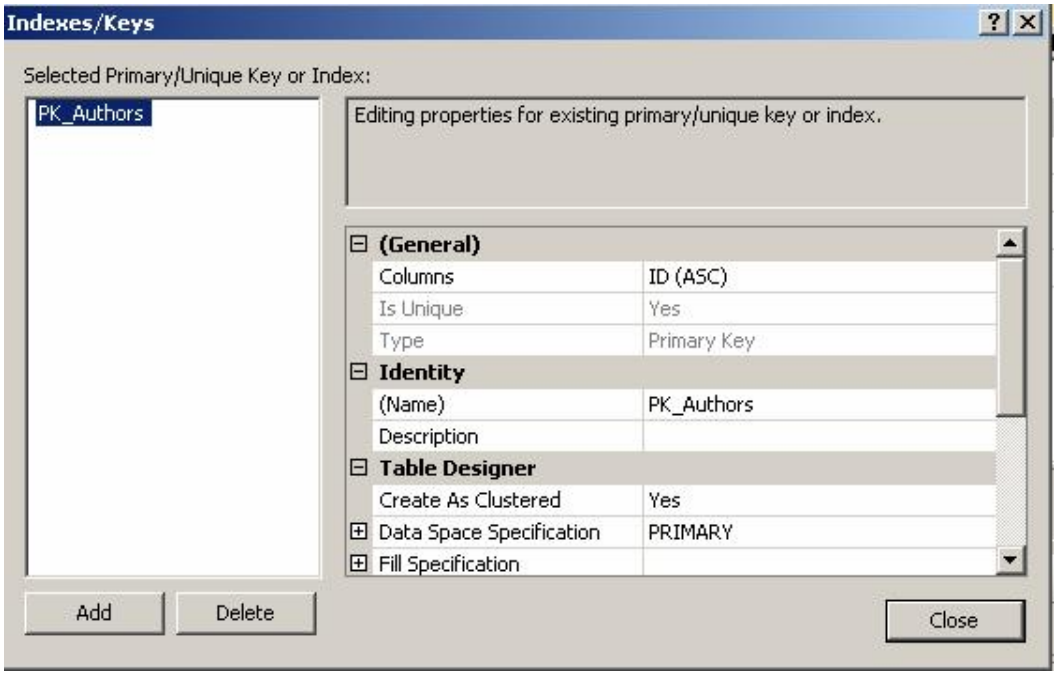
- اضغط الزر Test لاختبار الاتصال.. إذا حدثت مشكلة وأخبرتك رسالة الخطأ أن قاعدة البيانات موجودة من قبل، فافتح مدير سيكويل سيرفر SQL Server Management Studio، وقم بفصل Detach قاعدة بيانات الكتب منه، ثم أعد محاولة الاتصال بها من دوت نت.. ويمكنك أن تلجأ إلى هذه الخطوة كلما واجهت مشكلة في الاتصال بقاعدة البيانات.. فإذا لم يحل هذا المشكلة، فيمكنك عمل Restart لخادم سيكويل نفسه بالطريقة التي شرحناها من قبل.

- بعد نجاح تجربة الاتصال، اضغط OK لإغلاق النافذة.. ستجد أن اسم قاعدة البيانات قد ظهر في متصفح الخوادم، ويمكنك التعامل معها بنفس الطريقة التي اتبعناها في مدير سيكويل.

إنشاء الفهارس Indices:

نريد الآن إنشاء فهرس لأحد الأعمدة في قاعدة بيانات سيكويل.. من المتوقع في قاعدة بيانات الكتب مثلا، أن يتم البحث عن اسم الكتاب أو اسم المؤلف.. هذا يعني أننا نحتاج إلى فهرسة العمود Authors.Author والعمود Books.Book.. دعنا نرى كيف نفعل هذا:

في متصفح الخوادم، انقر مرتين بالفأرة على اسم جدول المؤلفين Authors لفتح تصميم الجدول، واضغط بزر الفأرة الأيمن في أي موضع خال من الجزء العلوي من صفحة التصميم، ومن القائمة الموضعية اختر الأمر Indexes/Keys.. سيعرض هذا نافذة إنشاء الفهارس والمفاتيح الموضحة في الصورة:



كما ترى، ستجد تعريفا للمفتاح الأساسي لهذا الجدول واسمه الافتراضي PK_Authors، وكما ترى من خصائص هذا الفهرس (المفتاح) أنه مرتبط بالحقل ID ومرتب تصاعديا ASC.

والآن، اضغط الزر Add لإضافة فهرس جديد.. سيضاف عنصر إلى الشجرة الموجودة على اليسار اسمه الافتراضي IX_Authors.. لاحظ أن البادئة IX هي اختصار Index،

ومن المنطقي افتراض أن العنصر الجديد فهرس وليس مفتاحا أساسيا، لأن الجدول لا يمكن أن يحتوي على أكثر من مفتاح أساسي واحد، بينما يمكن أن يحتوي على أكثر من فهرس.. على كل حال، تستطيع تغيير اسم هذا الفهرس لو أردت، والأفضل أن توضح فيه اسم الحقل أيضا.. غير قيمة الخاصية Name إلى IX_Authors_Author.

علينا الآن أن نحدد الحقل الذي يخصه هذا الفهرس.. اضغط الزر الموجود في خانة قيمة الخاصية Columns.. سيعرض هذا نافذة اختيار أعمدة الفهرس، وهي كما في الصورة:

وظيفة هذه النافذة، هي إتاحة الفرصة لك لاختيار أكثر من عمود لتتم فهرستها معا.. وكما ذكرنا من قبل، سيتم الترتيب على أساس العمود الأول، فإن تشابهت بعض عناصره، فسيتم ترتيبها على أساس العمود الثاني، فإن تشابهت بعض الأزواج من العمودين الأول والثاني معا، فسيتم ترتيبها على أساس العمود الثالث... وهكذا.. كل ما عليك في هذه الحالة، هو الضغط بالفأرة في صف فارغ في النافذة، واختيار اسم الحقل من القائمة المنسدلة من العمود الأول، واختيار نوع الترتيب (تصاعديا Ascending أم تنازليا Desinding) من

العمود الثاني.. لكن في معظم الأحول لن تحتاج إلا إلى فهرسة عمود واحد كما في حالتنا هذه.. لهذا اختر اسم العمود Authors واترك الترتيب تصاعديا كما هو وأغلق النافذة بضغط الزر OK للعودة إلى نافذة خصائص الفهرس.

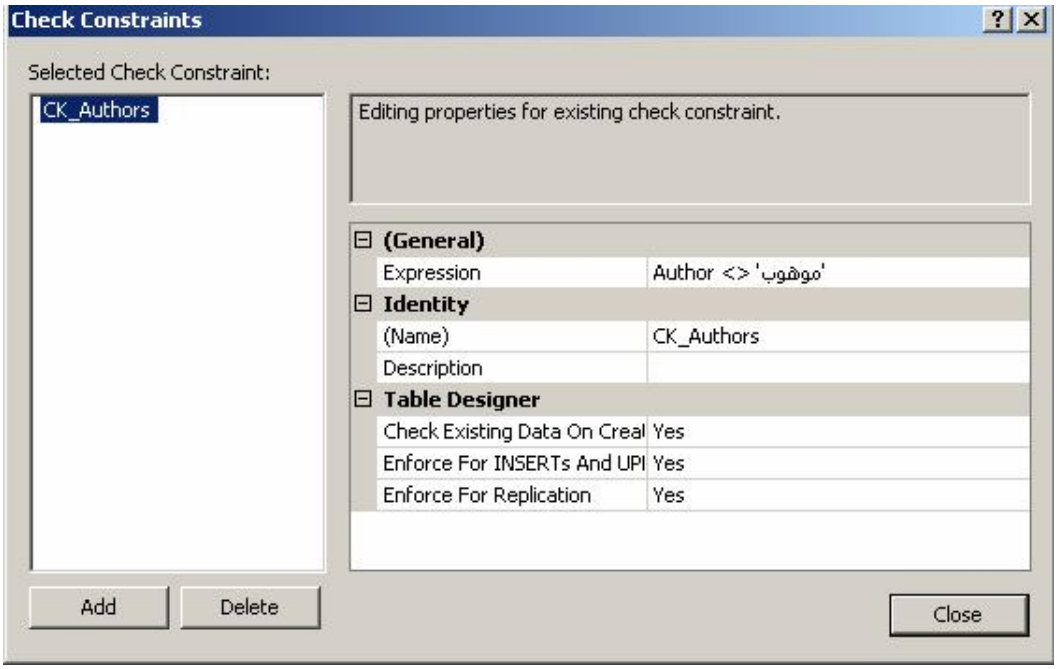
يمكنك تغيير بعض خصائص الفهرس على حسب احتياجك.. مثلا: لو أردت عدم السماح للمستخدم بتكرار أسماء المؤلفين في العمود Author، فاجعل للخاصية Is Unique القيمة Yes.. وإذا أردت اعتبار هذا الحقل مفتاحا متفردا Unique Key للجدول، فاجعل للخاصية Type القيمة Unique Key بدلا من Index.. وهكذا.

اضغط Close لإغلاق النافذة، واضغط زر الحفظ لحفظ التغييرات التي حدثت للجدول في قاعدة البيانات.. لقد تم إنشاء فهرس حقل المؤلفين الآن!

وبنفس الطريقة، يمكنك إنشاء فهرس لحقل Book في جدول الكتب.

إنشاء القيود Constraints:

رأينا كيف ننشئ قواعد التحقق من الصحّة في Access.. هنا أيضا يمكن أن نفعل المثل.. اضغط بزر الفأرة الأيمن في أي موضع خال من الجزء العلوي من صفحة تصميم الجدول Authors، ومن القائمة الموضعية اضغط الأمر Check Constrains.. سيؤدي هذا إلى عرض نافذة إنشاء قواعد التحقق من الصحة، كما هي في الصورة:



في الخاصية Expression اكتب قاعدة التحقق من الصحّة.. لاحظ أنّ هناك اختلافا طفيفا عن طريقة كتابة القاعدة في Access، فأنت هنا مضطرّ إلى إدخال اسم الحقل في التعبير، فبينما كانت قاعدة التحقّق من الصحّة خاصيّة للحقل في Access، فإنّها هنا جزء من الجدول ككل.. هذا يمنحك القدرة على كتابة قاعدة معقدة تتحقق من قيمة أكثر من حقل معا.. فمثلا، لاستبعاد صديقنا (موهوب) من حقل المؤلفين، اكتب صيغة الشرط كالتالي:

Author <> 'موهوب'

ولو كانت صيغة الشرط طويلة جدا، فيمكنك ضغط الزر الموجود في خانة الشرط، لعرض نافذة بها مربع نص متعدد الأسطر، لتكتب شروطا طويلة ومعقدة، وعند ضغط الـــزر OK لإغلاق نافذة الشرط، سيتم التأكد إن كانت صيغة الشرط الذي كتبته مقبولة أم لا.
أغلق النافذة وقم بحفظ التغييرات، ثمّ انتقل إلى تخطيط الجدول وجرّب كتابة اسم المؤلـــف الفذ (موهوب)!

***

الآن، نكون قد ألممنا بأهمّ مفاهيم قواعد بيانات سيكيول سيرفر وكيفيّة إنشائها والاتصال بها من دوت نت.. وفي الفصل التالي، سنتعرّف على كيفيّة التعامـــل معهـــا، باســـتخدام لغـــة الاستعلام المركبة SQL.. فإلى هناك.

– ٤ –

لغة الاستعلام المركّبة
Structured Query Language (SQL)

مهما كان نوع تطبيق قاعدة البيانات الذي تتعامل معه، فإنه يستخدم لغة خاصّة بقواعد البيانات متفقا عليها دوليّا اسمها SQL (تنطق "اس كيو إل" أو "سيكويل").. صحيح أن هناك اختلافات في تركيب هذه اللغة بينَ تطبيق وآخر، ولكنها اختلافات طفيفة لا تكفي للادعاء بأنها نسخ مختلفة تمام الاختلاف.

وبالمقارنة بلغات البرمجة المألوفة، تعتبر SQL لغة غير إجرائيّة Non-procedural، فهي لا تحتوي على تركيبات لغويةٍ مثل جمل الشرط وجمل التكرار وما شابهها.. وتعتبر SQL لغة برمجة عالية المستوي، حيث يمكنها في سطر واحد إجراء عمليّات بليغة التعقيد على قاعدة البيانات.

ملحوظة:
يقدم لنا SQL Server لغة إجرائية تسمى T-SQL وهي تستخدم جمل SQL مع بعض جمل تعريف المتغيرات وجمل الشرط لكتابة ما يسمى بالإجراءات المخزنة Stored Procedures كما سنرى لاحقا.. لكن نظرا لأن لغة T-SQL أكبر من أن نغطيها هنا، فسنتركها إلى كتاب لاحق بإذن الله.

وتنقسم جمل لغة SQL إلى طائفتين رئيسيّتين:

١- لغة التعامل مع البيانات Data Manipulation Language (DML):

وتختصّ باسترجاع أو تحديث أو إضافة أو حذف السجلات التي تحقق شروطا معيّنة، لهذا تسمّى بالاستعلامات Queries.. وتسمّى جمل استرجاع البيانات باستعلامات التحديد Selection Queries، بينما تسمّى جمل الحذف والتعديل والإضافة باستعلامات الفعل Action Queries.

٢- لغة تعريف البيانات Data Definition Language (DDL):

وتختصّ بإنشاء أو حذف كائنات قاعدة البيانات، كالجداول والأعمدة والفهارس والعلاقات والقيود Constrains.

باني الاستعلام Query Builder:

لكيّ تجرّب الأمثلة التي سنشرحها في هذا الفصل، يمكنك استخدام نافذة تسمى "باني الاستعلام" Query Builder، لإنشاء جمل SQL بطريقة مرئيّة Visual.

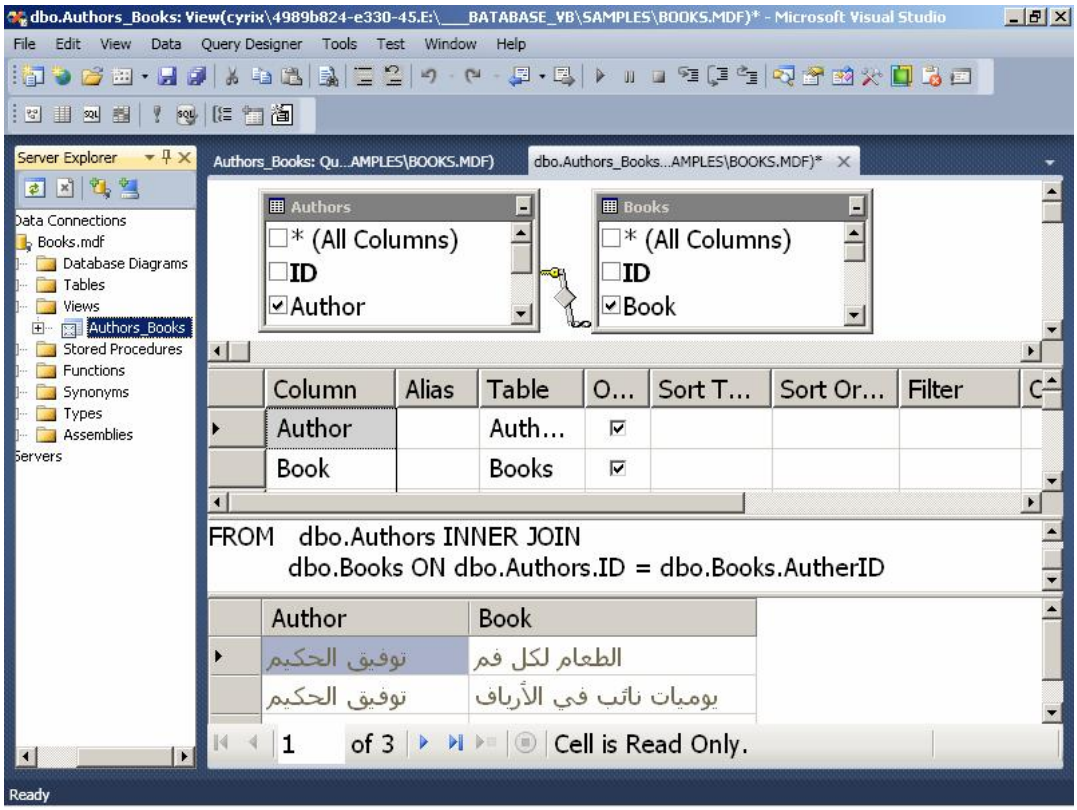
ولعرض باني الاستعلام، أسدل عناصر قاعدة البيانات Books في متصفح خوادم الإنترنت بزرّ الفأرة الأيمن.. ستجد ضمن عناصر قاعدة البيانات عنصرا اسمه Views.. تحت هذا العنصر يمكنك أن تنشئ العُروض Views.

اضغط العنصر Views برز الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Add New View.. ستظهر لك نافذة "إضافة جدول" Add Table، وهي تحتوي على أسماء جداول قاعدة بيانات الكتب.. حدّد الجدولين Authors و Books واضغط الزرّ Add، ثم الزر Close.

الآن أنت ترى نافذة باني الاستعلام، التي تتكوّن من أربعة أجزاء 4 Panes:

- جزء المخطط Diagram Pane.
- جزء المعايير Criteria Pane.
- جزء جملة الاستعلام SQL Pane.
- جزء النتائج Results Pane.

وتستطيع عرض أو إخفاء أي جزء من هذه الأجزاء، وذلك بضغط باني الاستعلام بزر الفأرة الأيمن، وضغط اسم الجزء المطلوب من القائمة الفرعية Pane.



قسم المخطّط Diagram Pane:

في هذا الجزء، تظهر الجداول التي اخترت استخدامها في بناء الاستعلام، والتي ستحصل على البيانات منها.. هذه الجداول تظهر في صورة مشابهة للتي رأيناها في مخططات قاعدة البيانات Database Diagrams.

ويمكنك حذف أيّ جدول بضغطه بالفأرة، وضغط الزرّ Delete من لوحة المفاتيح.. ولإضافة الجدول مرّة أخرى، اضغط بزرّ الفأرة الأيمن في أيّ موضع من المخطّط، ومن القائمة الموضعيّة اضغط Add Table، لتظهر لك نافذة إضافة جدول.

وإذا كانت هناك علاقات قد تمّ إنشاؤها بين الجداول سابقا، فستجد الخطوط التي توضّحها مرسومة بين الجداول.. كما يمكنك إنشاء العلاقات بين الجداول، بسحب اسم الحقل من

الجدول الرئيسيّ وإسقاطه على اسم الحقل في الجدول الفرعي، بنفس الطريقة التي تعلمناها عند التعامل مع مخططات قواعد البيانات.

وفي حالتنا هذه سيظهر خط يصل بين الحقل ID في جدول المؤلفين، والحقل AuthorID في جدول الكتب، وعلى منتصف هذا الخط سيظهر رمز يمثل نوع الربط الذي سيتمّ تنفيذه في جملة SQL ما بين الجدولين.. لاحظ أن للربط أنواعا عديدة، يمكن اختيارها بضغط الرمز بزرّ الفأرة الأيمن لعرض القائمة الموضعيّة.

ويمثل شكل الجوهرة ربطا داخليّا Inner Join، وهذه هي القيمة الافتراضيّة ما لم تختر أنت غير ذلك.. (سنفهم معنى هذه الارتباطات لاحقا).

أوّل خطوة لبناء جملة SQL، هي اختيار الحقول التي ستظهر في النتيجة.. لفعل هذا اضغط بالفأرة مربّع الاختيار على يسار كل حقل تريد إضافته.. فإذا كنت تريد إضافة كل حقول الجدول مرّة واحدة، فاضغط مربّع اختيار العنصر (All Columns).. وفي مثالنا هذا، اختر الحقل Author من جدول المؤلفين، والحقل Book من جدول الكتب، لنعرض أسماء المؤلفين وكتب كل منهم.

قسم المعايير Criteria Pane:

يعرض هذا الجزء جدولا يحتوي على أسماء الحقول التي تمّ اختيارها في جزء المخطّط، ويتيح لك تحديد الشروط والمعايير التي تريد تطبيقها على هذه الحقول، لاختيار بيانات معينة منها.

ويمكنك أن توضح أن بعض هذه الحقول تستخدم فقط لقراءة البيانات منها، لكن دونَ أن تظهر في النتيجة.. لفعل هذا أزل العلامة (✓) من خانة العمود Output المناظرة لهذه الحقول.

وتمثل خانة "الاسم المستعار" Alias الاسم الجديد الذي تريد عرض العمود به في النتيجة.. فقد لا تريد مثلا، أن تعرض لمستخدم برنامجك أسماء المؤلفين العرب تحت عمود اسمه Author، وبدلا من هذا تريد أن تمنح لهذا العمود الاسم المستعار "المؤلف".. أيضا، يمكنك أن تعرض الحقل Book بالاسم "الكتاب".

وهناك حالة أخرى تحتاج فيها لاستخدام الاسم المستعار، تلك إذا كانت النتيجة تحتوي على عمودين أو أكثر من جداول مختلفة لها نفس الاسم، فتحتاج إلى التفريق بينها قبل عرضهما للمستخدم.

ويمكنك ترك الخانة Alias فارغة إذا لم تكن تريد استخدام اسم مستعار.

وسنتعرف لاحقا على كيفية استخدام جزء المعايير Criteria Pane لإضافة الشروط المطلوب تطبيقها على الحقول.

قسم الاستعلام SQL Pane:

في هذا الجزء يظهر نصّ جملة SQL، المولد آليا نتيجة اختياراتك التي أجريتها في قسم المخطط وقسم المعايير.. هذا يجعلك بقليل من الملاحظة تتعلم لغة SQL، وذلك بتجريب بعض التغييرات في المخطط وجدول الحقول وملاحظة تأثيرها على جملة SQL.

وبإمكانك التعديل في جملة SQL المكتوبة في هذا القسم على حسب ما يناسبك، وسيظهر أثر هذا التعديل في جزء المخطط وجزء المعايير.. وبإمكانك أيضا كتابة أيّ جملة SQL (أو لصقها) في هذا القسم لتجريبها، كما يمكنك أن تنسخ جملة SQL من هنا لتستخدمها في أيّ موضعٍ آخر من برنامجك.

ولكن.. إلى الآن لم يتمّ تنفيذ الاستعلام، ولم نختبر ناتجه!

اضغط بزرّ الفأرة الأيمن في أيّ موضع من باني الاستعلام، ومن القائمة الموضعيّة اضغط الأمر Run Query، أو اضغط الاختصار Ctrl+R من لوحة المفاتيح مباشرة.

قسم النتائج Results Pane:

هذا هو الموضع الذي تظهر به نتائج تنفيذ الاستعلام.. ولمحو النتائج، اضغط قسم النتائج بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اختر الأمر Clear Results.

بعد أن تنتهي من تصميم عرض البيانات View، احفظه بالاسم Authors_Books، ثم أغلق نافذة باني الاستعلام.. الآن سيظهر الاسم Authors_Books تحت العنصر Views

في متصفّح الخوادم، ولو أردت إعادة تصميم هذا العرض، فاضغطه بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Design View، أو اضغط اسم العرض مرتين بالفأرة مباشرة.. سيعرض هذا باني الاستعلام وفيه تفاصيل الاستعلام الخاص بهذا العرض.

ويمكنك أيضا أن تضغط اسم العرض Authors_Books بزر الفأرة الأيمن، واختيار الأمر Show Results من القائمة الموضعية.. سيعرض هذا جدولا يحتوي على نتائج تنفيذ الاستعلام.

كانت هذه فكرةً سريعةً عن باني الاستعلام، ليمكنك استخدامه في تجربة جمل SQL التي سنشرحها في المقاطع التالية.. كل ما عليك هو نسخ الجملة إلى قسم SQL وضغط Ctrl+R لتنفيذها.. وسنتعرّف على باقي إمكانيّات باني الاستعلام أثناء شرحنا لجمل SQL، حيث سننوّه إلى كيفيّة استخدامه لبناء كل جملة.

لاحظ أنك لا تحتاج إلى إنشاء عرض View جديد لكل جملة استعلام تريد تجربتها، فبإمكانك ضغط العنصر Views بزر الفأرة الأيمن، وضغط الأمر "استعلام جديد" New Query، فهذا سيؤدي إلى عرض نافذة باني الاستعلام لتجرب فيها ما تشاء، وبعد هذا تغلقها دون حفظ التغييرات.

والآن، دعنا نتعرف على لغة الاستعلام المركبة SQL.

استعلامات التحديد Selection Queries

تستخدم هذه الاستعلامات للحصول على بعض البيانات من قاعدة البيانات أو إضافة بيانات جديدة إليها، أو تعديل أو حذف بيانات موجودة فيها.. وفيما يلي نتعرف على أهم جمل هذا النوع من الاستعلامات:

جملة التحديد SELECT Statement:

تمكنك الجملة SELECT من استرجاع جزء من صفوف الجدول.. وأبسط صورة لهذه الجملة هي:

SELECT أسماء الأعمدة

FROM أسماء الجداول

لاحظ ما يلي:

١- جمل SQL غير حساسة لحالة الأحرف، لكن جرى العرف على كتابة كلماتها المفتاحية كاملة بحروف كبيرة Capital على سبيل التمييز، لهذا لو كتبت select فلن تختلف عن Select ولا عن SELECT في شيء.

٢- أسماء الأعمدة والجداول يفصل بينها العلامة ","، مثال:

```
SELECT Author, Book
FROM   Authors, Books
```

لاحظ أنك لو جرّبت هذا المثال فستفاجأ بنتائج غريبة، حيث سيتم تكرار كل سجل من سجلات المؤلفين مع كل أسماء الكتب، ممّا يبدو معه أن كل مؤلف قد ألف كل الكتب!.. هذا خطأ متوقّع، لأننا لم نضع أيّ شرط يوضح طريقة الربط بين حقول الجدولين، وهو ما سنتعلمه بعد قليل.

٣- إذا كان اسم الجدول أو اسم الحقل يحتوي على أيّة رموز غير مقبولة برمجيّا (مثل المسافات أو العلامات ":" أو "+" أو "=" ... إلخ) فيجب وضعه بين قوسين مضلعين [ ].. وعامة، يمكنك وضع اسم أي جدول أو اسم أي حقل بين قوسين مضلعين.. مثال:

```
SELECT [Author], [Book]
FROM   [Authors], [Books]
```

٤- إذا تشابهت بعض أسماء الحقول في أكثر من جدول، فيجب التمييز بينها بكتابة اسم كل حقل على الصيغة: (اسم الجدول.اسم الحقل).. وعموما، يمكنك كتابة اسم الجدول قبل اسم أي حقل حتى ولو لم يكن له شبيه.. مثال:

```
SELECT [Authors].[Author], [Books].[Book]
FROM   [Authors], [Books]
```

ويمكنك إزالة الأقواس المضلعة إذا لم تكن هناك حاجة إليها:

```
SELECT Authors.Author, Books.Book
FROM   Authors, Books
```

٥- ستظهر الحقول في السجلات المعادة بالترتيب الذي كتبت أسماءها به في جملة SELECT.. هذا يعني أن الحقل Author سيظهر أولا ثم الحقل Book في نتيجة الاستعلام السابق.

٦- يمكنك كتابة جملة SQL على أكثر من سطر، فهي ليست كأوامر فيجيوال بيزيك تنتهي بنهاية السطر، ولكنها كأوامر C++ تنتهي بالفاصلة المنقوطة ";"، وإن كان من الممكن عدم كتابة الفاصلة المنقوطة كاختصار.. هذا معناه أن الجملة التالية صحيحة:

```
SELECT Author, Book FROM Authors, Books
```

لكن تقسيمها على سطرين يجعل قراءتها أسهل.. هذا مجرد تنسيق.

٧- معظم الأخطاء التي تحدث في كتابة جملة SQL تنتج عن الخطإ في كتابة أسماء الحقول والجداول، فانتبه لذلك جيّدا.

٨- لتحديد كل سجلات الجدول، يمكنك كتابة أسمائها جميعا والفصل بينها بالفاصلة.. ولكن هذا سيكون مأساويا لو كان الجدول يحتوي على عشرين عمودا مثلا!.. لهذا تمنحك SQL تسهيلا رائعا، وهو استخدام العلامة "*" (أو الكلمة ALL) للدلالة على تحديد كل أعمدة الجدول كالتالي:

```
SELECT *
FROM Authors
```

أو:

```
SELECT All
FROM Authors
```

ولو أردت استرجاع كل الحقول من جدولين، فاستخدم جملة كالتالية:

```
SELECT Authors.*, Books.*
FROM Authors, Books
```

٩- يمكنك كتابة اسم مختصر للجدول بعد اسمه الحقيقيّ مباشرةً، واستخدامه في باقي جملة الاستعلام للتسهيل.. مثال:

```
SELECT A.*, B.ID, B.Book
FROM Authors A, Books B
```

كما يمكنك استخدام الكلمة AS بين الاسم الحقيقي والاسم المختصر كالتالي:

```
SELECT A.*, B.ID, B.Book
FROM Authors As A, Books As B
```

الفقرة "حيثُ" WHERE Clause:

لم تخترع قواعد البيانات لكي تقرأ منها كل سجلات الجدول كما أدخلتها، فالأهمّ من هذا هو قدرتك على حساب بعض النتائج واستخلاص المعلومات من هذه السجلات.. هنا يبرز الدور الحيويّ للفقرة WHERE، فهي تسمح لك بتحديد الشروط التي سيتمّ على أساسها استرجاع السجلات، بحيث تحصل فقط على السجلات التي تحقق هذه الشروط.

ولن تكون صيغة هذه الفقرة صعبةً عليك، فصيغة الشرط الخاصة بها مماثلةً للشروط التي تعوّدت كتابتها في جملة If في لغة فيجيوال بيزيك.

والمثال التالي يسترجع كل الكتب التي يحمل مؤلفها الرقم ١ أو ٣:

```
SELECT Book
FROM Books
WHERE AuthorID = 1 OR AuthorID = 3
```

ولكن من هو المؤلف رقم ١ ومن هو المؤلف رقم ٣؟

هذه الطريقة تبدو عديمة الجدوى، فهي غير عمليّة في البرمجة!.. إننا نريد أن نعرف الكتب التي ألفها (توفيق الحكيم) مباشرة دونَ أن تبحث عن رقمه أولا.. لا بأس إذن فهناك حل.. دعنا نستخدم الجملة التالية:

```
SELECT Book
FROM Books, Authors
WHERE Author = 'توفيق الحكيم' AND AuthorID = Authors.ID
```

لاحظ أننا كتبنا اسم جدول المؤلفين في الفقرة FROM رغم أن أيا من حقوله لن يظهر في النتيجة، وذلك لأننا نستخدم حقول هذا الجدول في الفقرة WHERE.

والآن هذا ما سيحدث: سيتمّ اختيار السجلات التي تحمل اسم (توفيق الحكيم) في جدول المؤلفين، ثمّ اختيار السجلات من جدول الكتب التي يتساوى فيها رقم المؤلف مع رقم المؤلف في السجلات المختارة من جدول المؤلفين.. ونظرا لأنه لن يوجد سوى رقم (توفيق الحكيم) فحسب، فإن كتبه فقط هي التي سيتمّ عرضها.

ومن الممكن تبديل ترتيب جملتي الشرط فالترتيب ليس مهمّا، حيث تقوم قاعدة البيانات بتنفيذ الشروط على حسب أولويّة تنفيذها، وليس على حسب ترتيب كتابتها:

```
SELECT Book
FROM Books, Authors
WHERE AuthorID = Authors.ID AND Author = 'توفيق الحكيم'
```

والآن تعال نطوّر المثال السابق، لنحصل على كل الكتب التي ألفها (توفيق الحكيم) أو (عباس العقاد).. في هذه الحالة لا بدّ من عرض اسم المؤلف في النتيجة، وإلا لتعذر معرفة مؤلّف كل كتاب:

```
SELECT Book, Author
FROM Books, Authors
WHERE (Author = 'توفيق الحكيم' OR Author = 'عباس العقاد')
      AND AuthorID = Authors.ID
```

ولكن ماذا لو أردنا أن نعرض كل أسماء الكتب مع ما يناظرها من مؤلفين؟

سيكون الأمر في منتهى البساطة، فسيقتصر الشرط في هذه الحالة على تساوي رقم المؤلف في الجدولين:

```
SELECT Book, Author
FROM Books, Authors
WHERE AuthorID = Authors.ID
```

ملحوظة:

يمكنك استرجاع جدول فارغ بوضع شرط خاطئ أو مستحيل في الفقرة WHERE (مثل ١ = ٢) كالتالي:

```
SELECT *
FROM Books
WHERE 1 = 2
```

يفيد هذا عندما تريد أن تحصل على سجل فارغ يحمل أسماء حقول الجدولة كاملة بدون بيانات، بحيث تسمح للمستخدم بإدخال سجلات جديدة، لكن دون العبث بالسجلات القديمة.

إنشاء الفقرة WHERE بباني الاستعلام:

لكي تضيف شروطا في باني الاستعلام على الحقول التي تريد استرجاعها، حدد الحقل في جزء المخطط لإضافته إلى جزء المعايير Criteria Pane.. دعنا نجرب مثلا على الحقل ID في جدول المؤلفين.

ستجد في جدول المعايير عمودا اسمه Filter.. هذا هو العمود الذي تستطيع أن تكتب فيه الشرط الذي تريده، لكن بدون كتابة اسم الحقل.. مثلا، لكي نحصل على كتب المؤلف رقم ٥، اكتب في الخانة Filter في صف الحقل ID الشرط:

```
= 5
```

ولو أردت المؤلفين الذين يزيد رقمهم عن ٥، فاكتب الشرط:

```
> 5
```

وهكذا....

وإذا أردت أن تكتب شرطا آخر مرتبطا بالشرط الأول بالمعامل OR، فانتقل إلى العمود التالي الذي يحمل العنوان OR، واكتب الشرط الذي تريده في هذا العمود.. فإذا كان هناك المزيد من الشروط، فلديك المزيد من الأعمدة التي تحمل العنوان OR.. فمثلا، لو أردت أن تحصل على المؤلفين الذين أرقامهم ٧ و ١٢ و ٢٠، فاجعل صف الحقل ID في جزء المعايير يبدو كالتالي (سأقتصر على ذكر الأعمدة التي تهمنا فقط):

Column	Table	Filter	Or	Or
ID	Authors	= 7	= 12	= 20

ولكن ماذا لو أردت أن تكتب شرطا يستخدم المعامل AND؟

في هذه الحالة يجب أن يظهر هذا الشرط في صفّ جديد.. فمثلا، لو وضعت في العمود Filter شرطا على رقم المؤلف ID وشرطا على اسم المؤلف Author، فسيتم ربط الشرطين معا بالمعامل AND.

ولكن.. ماذا لو أردنا أن يظهر نفس الحقل في أكثر من شرط يربطها المعامل AND، كأن نستعلم عن كل المؤلفين ما عدا المؤلفين رقما ٨ و ٩؟

في هذه الحالة يجب أن نكرر اسم الحقل ID في جزء المعايير.. لفعل هذا اضغط بالفأرة في أيّ خانة فارغة في العمود المسمّى Column، ومن القائمة المنسدلة اختر الحقل Authors.ID.

الآن صار لديك صفان للعمود ID.. اكتب في العمود Filter للأول الشرط:

<> 8

واكتب في العمود Filter الخاص بالثاني الشرط:

<> 9

ولكن ماذا لو كان هناك شرطان كل منهما على حقل مختلف، ويربطهما المعامل OR، كأن تريد اختيار المؤلف الذي رقمه ٦ أو اسمه توفيق الحكيم؟

في هذه الحالة اكتب في العمود Filter الخاص بالحقل ID الشرط:

= 6

واترك العمود Filter الخاص بالحقل Author فارغا، واكتب في العمود OR الخاص به الشرط:

= 'توفيق الحكيم'

الخلاصة:
الشروط التي تظهر في العمود Filter في صفوف مختلفة يربطها المعامل AND، بينما الشروط التي تظهر في العمود OR يربطها المعامل OR.. ولو أردت ربط حقلين بالمعامل Or فاترك العمود Filter الخاص بأحدهما فارغا، واكتب الشرط الخاص به في العمود OR.

معاملات المقارنة:

الجدول التالي يلخص المعاملات التي يمكن استخدامها لتكوين شروط مركبة مع الفقرة WHERE:

المعاملات المنطقيّة	
و	AND
أو	OR
ليس	NOT
معاملات المقارنة	
يساوي	=
لا يساوي	<>
أكبر من	>
أصغر من	<
أكبر من أو يساوي	>=
أصغر من أو يساوي	<=

فمثلا للحصول على أول ٨ كتب في جدول الكتب:

```
SELECT ID, Book
FROM Books
WHERE ID < 9
```

وللحصول على الكتب التي تسبق كتاب "حائرة في الحب" في الترتيب الهجـائيّ، اسـتخدم الجملة التالية:

```
SELECT ID, Book
FROM Books
WHERE Book < 'حائرة في الحب'
```

المعامل "بين" BETWEEN Operator:

يسمح لك هذا المعامل بتحديد المجال الذي ينتمي إليه الحقل.. فمثلا، يمكننا استخدام الجملـة التالية للحصول على الكتب التي ينحصر أسعارها بين ٣ و ١٠ جنيهات:

```
SELECT Book, Price
FROM Books
WHERE Price BETWEEN 3 AND 10
```

كما يمكننا استخدام NOT قبل هذا المعامل للحصول على قيم الحقل التي لا تنتمـي إلـى المجال المحدّد.. والجملة التالية تعيد الكتب التي لا تنحصر أسعارها بين ٣ و ١٠ جنيهات:

```
SELECT Book, Price
FROM Books
WHERE Price NOT BETWEEN 3 AND 10
```

المعامل "في" IN Operator:

أحيانا لا يحل المعامل BETWEEN كل مشاكلنا، فماذا لو أردنا أن نختار قيمـا متفرّقـة للحقل؟.. في هذه الحالة سيؤدي استخدام المعامل AND إلى كتابة جملة طويلة جدا.

هنا تبرز أهميـة المعامـل IN، فبعـده توضـع كـل القـيم بـين قوسـين مفصـولةً بالعلامة , .. والجملة التالية تعيد إليك الكتب التي تحمل الأرقام ٢ و ٦ و ٧ و ١٠:

```
SELECT ID, Book
FROM Books
WHERE ID IN (2, 6, 7, 10)
```

ويمكن نفي الجملة السابقة للحصول على باقي الكتب (التي لا تحمـل الأرقـام المـذكورة) كالتالي:

```
SELECT ID, Book
FROM Books
WHERE ID NOT IN (2, 6, 7, 10)
```

ولا يقتصر الأمر على الأرقام، فالجملة التالية تعيد كل الكتب التي ألفها (توفيق الحكيم) و(نبيل فاروق) و(أحمد رجب):

```
SELECT Book, Author
FROM Books, Authors
WHERE AuthorID = Authors.ID
    AND Author IN ('أحمد رجب' , 'نبيل فاروق' ,'توفيق الحكيم')
```

المعامل "يشبه" LIKE Operator:

يستخدم هذا المعامل في الفقرة WHERE بنفس الطريقة التي يستخدمه بها مبرمجو فيجيوال بيزيك في مقارنة النصوص، ولكن مع بعض الاختلافات الطفيفة، ستدركها عند التعرّف على العلامات الخاصّة التي يستخدمها هذا المعامل، وهي:

الرمز	الوظيفة

%	تعبّر عن أيّ عدد من الحروف (يمكن أن يكون هذا العدد صفرا)، مهما كانت هذه الحروف.
	مثال: استخدم الجملة التالية للحصول على جميع أسماء المؤلفين التي تحتوي على حرف الميم في أيّ موضع:
	`SELECT Author`
	`FROM Authors`
	`WHERE Author LIKE '%م%'`
	ولو أردت أن تبحث عن المؤلفين الذين يبدأون بحرف الميم، فاستخدم الصيغة 'م%'.

_	تعبّر عن حرف واحد فقط، مهما كان هذا الحرف. مثال: استخدم الجملة التالية للحصول على جميع أسماء المؤلفين التي يكون حرف الميم فيها هو ثالث حرف:

SELECT Author **FROM Authors** **WHERE Author LIKE "__م%'** حيث استخدمنا علامتي "_" متتاليتين لتحفظا موضعي حرفين (أيّ حرفين) يليهما الحرف الثالث وهو الميم، يليه العلامة % لتدّل على أن أيّ عدد من الحروف مهما كانت يمكن أن يأتي بعد حرف الميم، بما في ذلك أن يكون حرف الميم هو آخر حرف في النصّ ولا تليه أية حروف.	
تعبّر عن رقم منفرد من ٠ إلى ٩.	#
تعبّر عن حرفٍ واحدٍ من الحروف الموجودة بين القوسين.. هذه الحروف إمّا أن تُكتب متتالية مثل: **[ACdF]** وإمّا أن تُكتب على صورة مجال، مثل: **[g-y]** حيث تعبّر هذه الصيغة عن الحروف من g إلى y. مثال: استخدم الجملة التالية للحصول على جميع أسماء المؤلفين التي يكون حرف الميم أو النون أو الواو فيها هو ثالث حرف: **SELECT Author** **FROM Authors** **WHERE Author LIKE "__[منو]%'** ولو أردت أن تبحث عن المؤلفين الذين تبدأ أسماؤهم بأحد الحروف المحصورة بين الفاء والياء وتنتهي بحرف السين، فاستخدم الصيغة: **'س%[ف-ي]'**	[ ]
تعبّر عن أيّ حرفٍ غير ذلك الموجود في القوسين (سواء من الحروف المذكورة صراحةً أو من الحروف التي تقع في المجال المحدّد).. باختصار:هذا هو نفي الصيغة السابقة. مثال: استخدم الجملة التالية للحصول على جميع أسماء المؤلفين التي لا يكون ثالث حرف فيها A أو B أو C:	[^]

SELECT Author **FROM Authors** **WHERE Author LIKE "__[^ABC]%'** ولو أردت أن تبحث عن السجلات التي لا تبدأ بأحد الحروف المحصورة بين T و Z وتنتهي بالحرف S، فاستخدم الصيغة: '[^T-Z]%S'	

ملحوظة:
إذا أردت البحث في النصّ عن أيّ علامة من هذه العلامات الخاصّة، فضعها بين قوسين مضلعين.. فمثلا، يمكنك استخدام الصيغة '%[%]50%' للبحث عن النصّ "50%" في أيّ موضع من النص.

المقارنة بناتج استعلام:

من الإمكانيّات التي تمنحها لك الفقرة WHERE، قدرتك على مقارنة قيمة أحد الحقول، بقيمة أيّ حقل من جدول آخر، ناتج عن جملة SELECT كاملة أخرى!

انظر للمثال التالي، وفيه نحصل على كل الكتب التي ألفها (عباس العقاد) و(توفيق الحكيم):

```
SELECT *
From Books
WHERE AuthorID = (SELECT ID
                  FROM Authors
                  WHERE Author = 'عباس العقاد'
                  OR Author = 'توفيق الحكيم')
```

حيث تُعيد SELECT الفرعيّة حقلا به رقمي هذين المؤلفين، لتقوم جملة SELECT الخارجيّة بإيجاد السجلات التي تحتوي على أيّ من هذين الرقمين في الحقل AuthorID.

لاحظ أنك تستطيع اختصار الجملة السابقة إلى ما يلي:

```
SELECT Books.*
From Authors , Books
WHERE AuthorID = Authors.ID
        AND Author = 'عباس العقاد' OR Author = 'توفيق الحكيم'
```

بعض الدوال المستخدمة في إنشاء الشرط:

يمكنك استخدام بعض الدوال في تكوين شرط WHERE، مثل دول أخذ جزء من النص SUBSTRING و LEFT و RIGHT، ودالة حذف المسافات الطرفية TRIM ودالة طول النص LEN.. والمثال التالي يعيد الناشرين الذين تنتهي أسماؤهم بالحرفين "ية":

```
SELECT Publisher
FROM   Publishers
WHERE (RIGHT(Publisher, 2) = 'ية')
```

أما المثال التالي فيعيد أطوال أسماء الناشرين:

```
SELECT LEN(Publisher)
FROM   Publishers
```

وضع اسم مستعار باستخدام الكلمة AS:

رأينا من قبل كيف يمكن استخدام الكلمة AS لوضع اسم مستعار للجدول.. مثلا:

```
SELECT A.*, B.ID, B.Book
FROM Authors As A, Books As B
```

ويمكنك أيضا استخدام الكلمة AS لوضع اسم مستعار للحقل.. مثال:

```
SELECT Book AS [اسم الكتاب], Author AS [مؤلفه]
FROM Books, Authors
WHERE AuthorID = Authors.ID
```

الآن ستظهر أسماء المؤلفين وكتبهم، ولكن عنواني العمودين سيكونان اسمين عربيّين هذه المرّة.

ومن الاستخدامات الطريفة للاسم المستعار، استخدامه لتسميّة أحد الحقول الناتجة عن دمج حقلين معا.. افترض أننا نريد عرض حقل يحتوي على اسم الكتاب متبوعا بشرطة متبوعةً

باسم المؤلف.. لفعل هذا، علينا تشبيك Concatenate الحقلين، تماما كما نفعل مع المتغيّرات النصيّة في لغات البرمجة العادية، مع منح الحقل الناتج اسما مستعارا باستخدام الكلمة AS، كالتالي:

```
SELECT Book + '_' + Author AS [اسم الكتاب]
FROM Books, Authors
WHERE AuthorID = Authors.ID
```

ملحوظة:
عند تنفيذ الجملة السابقة في باني الاستعلام سيتم وضع التعبير Book + '_' + Author كحقل في قسم المعايير Criteria Pane، مع منحه الاسم المستعار "اسم الكتاب".. هذه هي الطريقة التي تضيف بها الحقول المدمجة في باني الاستعلام.. اكتب العمليّة التي تجمع بين الحقلين في أيّ خانة فارغة في العمود Column، وامنح هذا الحقل الجديد الاسم المستعار المناسب.

وليست الحقول النصيّة فقط هي ما نستطيع دمجه، حيث يمكننا أن نجرى عمليّات الجمع والطرح والضرب والقسمة على الحقول الرقميّة كما يحلو لنا.

الكلمة TOP:

يمكنك استخدام هذه الكلمة بعد SELECT مباشرة، وذلك إذا كان عدد السجلات الناتجة عن تنفيذ الاستعلام كبيرا، بينما ينحصر اهتمامك في مجموعة قليلة منها فقط.. فمثلا، لكي تعرض أوّل خمس سجلات من أسماء الكتب ومؤلفيها استخدم الجملة التالية:

```
SELECT TOP 5 Book, Author
FROM Books, Authors
WHERE AuthorID = Authors.ID
```

وإذا لم تكن تعرف بالضبط عدد السجلات المطلوبة، فيمكنك استخدام النسبة المئويّة PERCENT كالتالي:

```
SELECT TOP 5 PERCENT Book, Author
FROM Books, Authors
WHERE AuthorID = Authors.ID
```

الكلمة "منفصل" DISTINCT:

استخدم هذه الكلمة لإزالة السجلات المكرّرة من ناتج الاستعلام.. فمثلا، لو عرضنا جدولا بأسماء الكتب بدون مؤلفيها، فقد يتكرر اسم أحد الكتب أكثر من مرة (لمؤلفين مختلفين).. هنا يمكن تلافي هذا التكرار باستخدام الجملة التالية:

```
SELECT DISTINCT Book
FROM Books
```

القيم المنعدمة Null Values:

تشيرُ القيمة Null في قاعدة البيانات، إلى أن الخانة لم توضع بها أيّ قيمة من أساسِه.. لهذا تختلف الخانة التي بها نص فارغ "" عن الخانة التي ليس بها أي شيء على الإطلاق.

وعند تنفيذ الفقرة WHERE، فإنها لا تسترجع أيّ خانة فارغة، لأن القيمة NULL تنتج False بطريقة دائمة في أيّ عمليّة مقارنة.. هذا بالإضافة إلى أن دخول هذه القيمة في أيّ عمليّة حسابيّة يؤدّى إلى حدوث خطإ في البرنامج.

لهذا يجب أن تعامل هذه القيمة بحذر، حيث يمكنك أن تختبر وجودها باستخدام أحد التعبيرين IS NULL أو IS NOT NULL.. وكمثال، استخدم الاستعلام التالي للبحث عن أيّ خانة في الحقل Author تمّ تركها فارغة:

```
SELECT * FROM Authors
WHERE Author IS NULL
```

لاحظ أنك تستطيع تلافي الكثير من احتمالات الخطأ الناتجة عن القيمة Null، بإزالة علامة الاختيار من العمود Allow Nulls عند تصميم الحقول، بحيث لا يستطيع مدخل البيانات ترك خانة فارغة في هذا الحقل.

فقرة الترتيب ORDER BY:

يمكنك ترتيب السجلات الناتجة من الاستعلام تبعا لحقلٍ أو أكثر.. وفي المثال التالي سنعرض أسماء الكتب ومؤلفيها مرتبة باسم الكتاب:

```
SELECT Book, Author
FROM Books, Authors
ORDER BY Book
WHERE AuthorID = Authors.ID
```

ولكن لو كانت هناك كتب تحمل نفس الاسم، فكيف سيتمّ ترتيبها؟

في هذه الحالة يمكنك أن تحدّد حقل المؤلفين كمفتاح ثانٍ للترتيب، بحيث لو تشابهت أسماء الكتب، يتم ترتيب الكتب المتشابهة على حسب أسماء مؤلفيها:

```
SELECT Book, Author
FROM Books, Authors
ORDER BY Book, Author
WHERE AuthorID = Authors.ID
```

ويكون هذا الترتيب تصاعديّا في الوضع الافتراضيّ.. ولو أردت أن تغير طريقة الترتيب، فاستخدم الكلمة DESC بعد اسم أيّ حقلٍ تريد ترتيبه تنازليّا، والكلمة ASC بعد اسم الحقل الذي تريد ترتيبه تصاعديّا (وهي افتراضيّة ويمكن عدم كتابتها).. مثال:

```
SELECT Book, Author
FROM Books, Authors
ORDER BY Book DESC, Author ASC
WHERE AuthorID = Authors.ID
```

ملحوظة:

لوضع كيفيّة الترتيب في باني الاستعلام، حدد الحقل الذي تريد أن يتمّ الترتيب على أساسه في قسم المخطط Diagram Pane واضغطه بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Sort Ascending إذا كنت تريد الترتيب تصاعديّا، أو الأمر Sort Descending إذا كنت تريد الترتيب تنازليّا.. ستجد أن أيقونةً تمثل نوع الترتيب قد ظهرت بجوار اسم الحقل.. ولو أردت إزالة الترتيب، فاضغط نفس الأمر من القائمة الموضعيّة مرّة أخرى.. ويمكنك فعل ذلك مع أكثر من حقل ومن أيّ جدول، حيث ستظهر كلها في جملة SQL بالترتيب الذي أضفتها به.

كما يمكنك اختيار نوع الترتيب بطريقة أخرى، وذلك باستخدام قسم المعايير Criteria Pane، حيث يمكنك استخدام القائمة المنسدلة في العمود الذي يحمل العنوان Sort Type لتغيير طريقة ترتيب أي حقل.

وإذا أردت استخدام أكثر من حقل للترتيب على أساسها، فعليك استخدام العمود Sort Order للتحكم في أولوية هذه الحقول في عملية الترتيب، فالحقل الذي سيتم الترتيب على أساسه أولا أعطه الرقم ١، والحقل الذي سيتم الترتيب على أساسه في حالة تشابه قيم الحقل الأول أعطه الرقم ٢، وهكذا.

دوال التجميع Aggregate functions:

تمنحك SQL بعض الدوال الجاهزة لحساب بعض النتائج.. لاحظ ما يلي:

- هذه الدوال تقبل معاملا واحدا فقط، هو أحد أعمدة الجدول، أو أيّ عمود جديد ناتج عن إجراء عمليّة حسابيّة (كالجمع والطرح والضرب والقسمة) على واحد أو أكثر من الأعمدة، أو أيّ عمود ناتج من جملة SELECT فرعيّة.
- هذه الدوال تعيد قيمة واحدة فقط (رقما منفردا).. أيّ أن الناتج منها هو عمود يحتوي على خانة واحدة فقط.. ولو كنت ستعرض هذه النتيجة، فاستخدم التعبير AS لمنح هذا العمود الجديد اسما مناسبا، وإلا فإن SQL ستمنحه اسما افتراضيّا من لديها.
- لا مانع من استخدام الفقرة WHERE لتحديد السجلات التي ستتمّ العمليّة الحسابيّة عليها.

وهذه الدوال هي:

العدد COUNT:

تحسب عدد الخانات في العمود المرسل كمعامل.. ويمكن تطبيقها على أيّ نوع من البيانات.. والجملة التالية تحسب عدد الكتب التي ألفها توفيق الحكيم:

```
SELECT COUNT(Book) AS [عدد الكتب المتاحة]
FROM Books, Authors
WHERE Author = 'توفيق الحكيم'
        AND AuthorID = Authors.ID
```

لاحظ أن الحقول التي تحتوي على القيمة NULL لا يتمّ عدّها ضمنَ السجلات.. ولو أردت أن تفعل العكس، فعليك أن ترسل الرمز (*) كمعامل لهذه الدالة، حتى تأخذ هذه السجلات في اعتبارها:

```
SELECT COUNT(*) AS [عدد المؤلفين]
From Authors
```

العدد الكبير COUNT_BIG:

مماثلة للدالة COUNT في كل شيء، إلا أن ناتج الدالة COUNT يكون عددا صحيحا من النوع int، بينما ناتج الدالة COUNT_BIG يكون عددا صحيحا كبيرا bigint، لهذا عليك استخدامها عند التعامل مع جداول ضخمة يتجاوز عدد سجلاتها ٢ مليار سجل!

المجموع SUM:

تحسب مجموع الخانات في العمود المرسل كمعامل.. ويمكن تطبيقها على الأعمدة الرقميّة فقط.

مثال: الجملة التالية تحسب مجموع النسخ المتاحة من جميع الكتب:

```
SELECT SUM(Copies_No) AS [إجمالي النسخ]
From Books
```

المتوسط AVG:

تحسب المتوسّط الحسابيّ (مجموع العناصر ÷ عددها) لخانات العمود المرسل كمعامل.. ويمكن تطبيقها على الأعمدة الرقميّة فقط.. والمثال التالي يحسب متوسط عدد نسخ الكتب:

```
SELECT AVG(Copies_No) AS [متوسط النسخ]
From Books
```

الأصغر MIN:

تحسب أصغر قيمة في العمود المرسل كمعامل.. ويمكن تطبيقها على الأرقام والنصوص، وفي حالة النصوص ستعيد أصغر نصّ في الترتيب الأبجديّ.

الأكبر MAX:

تحسب أكبر قيمة في العمود المرسل كمعامل.. ويمكن تطبيقها على الأرقام والنصوص، وفي حالة النصوص ستعيد أكبر نصّ في الترتيب الأبجديّ.

المجموع التأكيدي CHECKSUM_AGG:

هذه الدالة خاصية بـ T-SQL فقط ولا تستطيع استخدامها مع قواعد بيانات آكسيس.. وهي تجري عملية حسابية على جميع قيم العمود، وتعيد المجموع التأكيدي Check Sum، وهو قيمة يمكنك استخدامها للتأكد من أن خانات العمود لم يحدث بها تغيير، فطالما ظلت ثابتة فهذا معناه أن قيم العمود لم تتغير.. هذا أفضل من حفظ جميع قيم العمود القديمة، ثم التأكد من أن كلا منها لم تتغير على حدة.. لاحظ أن هناك احتمالا صغيرا في أن تتغير بعض قيم العمود لكن تظل هذه الدالة تعطي نفس الناتج.. لاحظ أيضا أن هذه الدالة لا تهتم بكيفية ترتيب العمود.. مثال:

```
SELECT CHECKSUM_AGG(Copies_No)
FROM   Books
```

وتوجد دالة أخرى في T-SQL اسمها CHECKSUM، ولكنها ليست دالة تجميع، فهي تعيد عمودا جديدا وليس قيمة واحدة فقط.. وتنتج كل خانة في العمود العائد من حساب قيمة مختلطة Hash Value لكل خانة في العمود الأصلي.

التغير الإحصائي VAR:

تحسب التغير الإحصائي Statistical Variance للقيم الموجودة في العمود.. لو لم تدرس مادة الإحصاء من قبل، فلا تشغل بالك بهذه الدالة وكل الدوال التالية!

التغير الإحصائي السكاني VARP:

تحسب التغير الإحصائي السكاني Statistical variance for population للقيم الموجودة في العمود.

الانحراف الإحصائي المعياري STDEV:

تحسب الانحراف الإحصائي المعياري Statistical Standard Deviation للقيم الموجودة في العمود.

الانحراف الإحصائي المعياري السكاني STDEVP:

تحسب الانحراف الإحصائي المعياري السكاني Statistical Standard Deviation for Population لقيم العمود.

ويمكنك استخدام القيم المعادة من هذه الدوال في شروط الفقرة WHERE.. والمثال التالي يريك كيف نحسب عدد الكتب التي ألفها أوّل مؤلف في الترتيب الأبجديّ:

```
SELECT COUNT(Book) AS [عدد كتب المؤلف الأول]
FROM Books, Authors
WHERE AuthorID = Authors.ID
    AND Author = (SELECT MIN(Author)
                  From Authors)
```

حيث استخدمنا جملة SELECT فرعيّة لتعيد ناتج الدالة MIN في شرط الفقرة WHERE.. على كل حال، هناك صيغة ثانية للمثال الأخير، باستخدام الربط الداخليّ INNER JOIN الذي سنتعرّف عليه لاحقا.

ولكن ماذا لو أردت أن تطبّق هذه الدوال على حقل به قيم مكرّرة، وأردت ألا تأخذ التكرار في اعتبارك؟

في هذه الحالة يجب أن تستخدم الكلمة DISTINCT للحصول على حقل ليس به أيّ تكرار.. وتوضع هذه الكلمة قبل اسم الحقل مباشرة (داخل قوس الدالة).. والمثال التالي يريك كيف نحسب عدد المؤلفين بحساب عدد خانات الحقل AuthorID في جدول الكتب بدون تكرار:

```
SELECT COUNT (DISTINCT AuthorID) AS [عدد المؤلفين]
From Books
```

ملحوظة:
لاستخدام دوال التجميع في باني الاستعلام، اضغط اسم الحقل في قسم المعـــايير Criteria Pane بزرّ الفأرة الأيمن، ومن القائمة الموضـــعيّة اضـــغط الأمـــر Add Group By.. سيظهر عمود جديد في جدول الحقول اسمه Group By، وسيكون مكتوبا فيه مبديا النص Group By.. اضغط زرّ إسدال القائمة، حيث ستجد بها أسماء دوال التجميع التي يمكنـــك استخدامها.. اختر منها ما تريد لتطبيقه على هذا الحقل.

تجميع السجلات باستخدام التعبير GROUP BY:

رأينا كيف نجري بعض العمليّات الحسابيّة باستخدام دوال التجميع.. ولكن ماذا لـــو أردنـــا مثلا أن نحصل على سجل يحتوي على عدد الكتب التي ألفها كل مؤلف؟

في هذه الحالة لن تسعفنا التعبيرات التي تعلمناها حتى الآن.

هنا تبرز أهمّية التعبير GROUP BY، الذي يقوم بتقسيم سجلات الجدول إلى مجموعـــات فرعية، تمتلك كل مجموعة منها القيمة نفسها في حقل معين (كرقم المؤلف مثلا)، ومن ثـــم يتم تطبيق دوال التجميع على كل مجموعة فرعية بمفردها، وبهذا يكون ناتج دوال التجميـــع عمودا يحتوي على مجموعة من الصفوف، وليس صفا واحدا كما ألفنا من قبل.. هذه هـــي الجملة التي تحسب عدد كتب كل مؤلف:

```
SELECT AuthorID, COUNT(AuthorID) AS [عدد الكتب]
FROM Books
GROUP BY AuthorID
ORDER BY [عدد الكتب]
```

لاحظ قدرتنا على استخدام الاسم المستعار للعمود الناتج في باقي جملة SQL.

ولكن.. الجدول الناتج من الاستعلام السابق يحتوي على أرقام المؤلفين ولـــيس أســـماءهم.. لهذا لا ضير من استخدام الفقرة Where للربط بين الجدولين كالتالي:

```
SELECT Author, COUNT(AuthorID) AS [عدد الكتب]
FROM Authors, Books
Where  AuthorID = Authors.ID
GROUP BY Author
ORDER BY [عدد الكتب]
```

لاحظ أننا قمنا بعملية التجميع في الجملة الأخيرة باستخدام الحقل Author.. السبب في هذا، هو أن هناك قيدا صارما على الحقول المذكورة في المقطع SELECT، وهو أنها جميعا يجب أن تظهر إما في المقطع GROUP BY وإما في المقطع ORDER BY.. لهذا لو تركنا التجميع على الحقل AuthorID فستعترض سيكويل سيرفر على ظهور اسم الحقل Author في المقطع SELECT!

ملحوظة:
لاستخدام الفقرة Group By في باني الاستعلام، اضغط اسم الحقل في قسم المعايير Criteria Pane بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Add Group By.. سيظهر عمود جديد في جدول الحقول اسمه Group By، وسيكون مكتوبا فيه مبديا النص Group By.. هذا معناه تجميع القيم المتشابهة لهذا الحقل. ولو أردت إضافة دالة تجميع مع الفقرة Group By لنفس الحقل، فكرر اسم الحقل مرة أخرى في قسم المعايير بضغط صف فارغ بالفأرة واختيار اسم العمود من القائمة المنسدلة الموجودة في العمود Column، وبهذا تستطيع أن تضيف إلى هذا الحقل دالة التجميع التي تريدها.

استخدام الفقرة HAVING:

ماذا لو أردنا عرض أسماء المؤلفين الذين تزيد كتبهم عن كتاب واحد؟

في هذه الحالة يمكننا استخدام الفقرة HAVING، وهي فقرة شرطية تسمح باستخدام دوال التجميع في الشرط، كالتالي:

```
SELECT Author, COUNT(AuthorID) AS [عدد الكتب]
FROM Authors, Books
Where  AuthorID = Authors.ID
GROUP BY Authors.Author
HAVING COUNT(AuthorID) > 1
ORDER BY [عدد الكتب]
```

ملحوظة:
لبناء الفقرة HAVING السابقة في باني الاستعلام.. اذهب إلى الخانة Group By الخاصة بالعمود AuthorID في قسم المعايير، ومن القائمة المنسدلة اختر الدالة Count، ثم انتقل إلى الخانة Filter المجاورة، واكتب فيها الشرط 1<.. هذا معناه أن أي شرط تكتبه في الخانة Filter أثناء ظهور العمود Group By سيظهر كشرط في الفقرة Having.. لكن لو كنت تريد استخدامه كشرط في الفقرة Where، فأسدل القائمة المنسدلة في الخانة Group By واختر منها العنصر Where.

عمليات الربط SQL Joins:

تحدّد عمليّات ربط الجداول Joins كيفيّة استعادة البيانات من الجداول التي بينها علاقـــات.. وهناك خمسة أنواع من عمليّات الربط:

١- الربط المتقاطع Cross Join.

٢- الربط الأيسر Left Join.

٣- الربط الأيمن Right Join.

٤- الربط الكامل Full Join.

٥- الربط الداخليّ Inner Join.

دعنا نتعرف على هذه الانواع.

الربط المتقاطع Cross Join:

لقد تعرفنا على هذا النوع من الربط سابقا، فهو ينتج تلقائيا عند عرض حقلين من جـــدولين مختلفين بدون ذكر أي طريقة لربطهما معا.. في هذه الحالة، يتم ما يشبه عملية الضـــرب، حيث يتم تكوين أزواج تبادلية من الحقلين، بحيث تظهر كل خانة من الجدول الأول مع كل خانة من الجدول الثاني في هذه الأزواج، وبهذا يكون عدد السجلات الناتجة = عدد خانـــات الحقل الأول × عدد خانات الحقل الثاني.. مثال:

```
SELECT Author, Book
FROM   Authors, Books
```

عند تنفيذ هذا الاستعلام، سيظهر ناتج شبيه بما يلي (سنفترض أن هناك مؤلفين اثنين فقـــط على سبيل الاختصار):

Author	Book
توفيق الحكيم	الطعام لكل فم
توفيق الحكيم	شهرزاد
توفيق الحكيم	مهنتي القتل
توفيق الحكيم	الاحتلال

الطعام لكل فم	نبيل فاروق
شهرزاد	نبيل فاروق
مهنتي القتل	نبيل فاروق
الاحتلال	نبيل فاروق

طبعا هذا ناتج عجيب وليس عمليا هنا، ولكنه قد يبدو عمليا في بعض التطبيقات الرياضية التي تحتاج فيها إلى الحصول على تباديل قيمتين أو أكثر.

وتوجد فقرة خاصة بهذا النوع من الربط، وهي الفقرة CROSS JOIN، ولو كتبت الاستعلام السابق في باني الاستعلام، فسيتم تحويله تلقائيا إلى الصيغة التالية:

```
SELECT Author, Book
FROM   Authors CROSS JOIN Books
```

ملحوظة:
لإنشاء الجملة السابقة باستخدام باني الاستعلام، حدّد الحقل Book من جدول الكتب، والحقل Author من جدول المؤلفين، ثمّ اضغط بزرّ الفأرة الأيمن على الخط الواصل بينَ الجدولين، ومن القائمة الموضعيّة اضغط الأمر: Delete لحذف الربط بين الجدولين.. هذا سيجعل الربط بينهما تقاطعيا!.. لاحظ أن حذف هذا الخط لا يحذف العلاقة الموجودة بين الجدولين، ولكنه فقط يؤثر على كيفية بناء الاستعلام.. ولإعادة رسم الخط مرة أخرى، اسحب الحقل Authors.ID وأسقطه على الحقل Books.AuthorID.

الربط الأيسر Left Join:

تعرض هذه العمليّة كل سجلات الجدول الأيسر (الموجود في بداية الصيغة)، مع بعض سجلات الجدول الأيمن، التي تحقق شرط الربط (الذي يأتي بعد الكلمة ON).

ويمكننا أن نعرض كل أسماء المؤلفين وأسماء الكتب الخاصة بكل منهم، كالتالي:

```
SELECT Book, Author
FROM Authors LEFT JOIN Books
ON AuthorID = Authors.ID
```

ولكي تشعر بوجود اختلاف عن ناتج الجملة WHERE، يجب أن يكـــون هنـــاك بعـــض المؤلفين الذين لا توجد لهم كتب مناظرة في جدول الكتب، فاستخدام الربط الأيسر سيعرض أسماء كل المؤلفين مع ترك خانة الكتاب فارغة NULL للمؤلف الذي لا توجد له كتـــب.. لكن دون عرض الكتب التي لا يناظرها مؤلفون.. بينما الجملة WHERE لا تعـــرض إلا المؤلفين الذي لهم كتب.

ملحوظة:
لإنشاء الجملة السابقة باستخدام باني الاستعلام، حدّد الحقل Book مـــن جـــدول الكتـــب، والحقل Author من جدول المؤلفين، ثمّ اضغط بزرّ الفأرة الأيمن على علامة الربط فـــي منتصف الخط الواصل بينَ الجدولين، ومن القائمة الموضعيّة اضغط الأمـــر: Select All Rows From Authors.. هذا هو كل شيء!

<u>الربط الأيمن Right Join:</u>

مماثل للربط الأيسر، إلا إن كل سجلات الجدول الأيمن يتمّ عرضها بالكامل، مـــع عـــرض سجلات الجدول الأيسر التي تحقّق شرط الربط.. مثال:

```
SELECT Book, Author
FROM Authors RIGHT JOIN Books
ON AuthorID = Authors.ID
```

ولكي تشعر بتأثير الربط الأيمن، يجب أن يكون هناك بعض الكتب التي لا يوجد لها مؤلف في جدول المؤلفين، فاستخدام الربط الأيمن سيعرض أسماء كل الكتب، والكتاب الذي لـــيس له مؤلف سيتم ترك خانة المؤلف المناظرة له فارغة NULL.

تذكر أننا عندما أنشأنا الحقل AuthorID أزلنا علامة الاختيـــار مـــن الخاصـــيّة Allow Nulls لهذا لن يمكنك ترك خانة رقم المؤلف فارغة..ويمكنك تغيير قيمة هـــذه الخاصـــيّة وتجرّبة إدخال كتاب بدون مؤلف، لترى تأثير عمليّة الربط الأيمن.

ملحوظة:
لإنشاء الجملة السابقة باستخدام باني الاستعلام، حدّد الحقل Book من جدول الكتب، والحقل Author من جدول المؤلفين، ثمّ اضغط بزرّ الفأرة الأيمن على علامة الربط في منتصف الخط الواصل بينَ الجدولين، ومن القائمة الموضعيّة اضغط الأمر: Select All Rows From Books.. هذا هو كل شيء!

الربط الكامل Full Join:

هذا النوع هو مزيج من الربط الأيمن والأيسر، وفيه يتمّ عرض كل بيانات الجدولين التي تحقق شرط الربط.. مثال:

```
SELECT Book, Author
FROM Authors FULL JOIN Books
ON AuthorID = Authors.ID
```

ولكي تشعر بتأثير الربط الكامل، يجب أن يكون هناك بعض المؤلفين الذين لا توجد لهم كتب، وبعض الكتب التي لا يوجد لها مؤلف، حيث سيتمّ عرض أسماء هؤلاء المؤلفين وهذه الكتب، مع ترك الخانة المناظرة فارغة NULL.

ملحوظة:
لإنشاء الجملة السابقة باستخدام باني الاستعلام، اتبع نفس الخطوات التي اتبعناها في إنشاء الربط الأيسر، ولكن اختر كلا الاختيارين Select All Rows From Books، و Select All Rows From Authors.. لاحظ تغير شكل الأيقونة المجاورة عند اختيار أي من هذين الخيارين.. ولإلغاء أي اختيار اضغطه مرة أخرى.

الربط الداخليّ Inner Join:

هذه العمليّة مماثلة للفقرة WHERE، حيث يتمّ عرض السجلات المتوافقة فقط من الجدولين، مع تجاهل الخانات الفارغة:

```
SELECT Book, Author
FROM Authors INNER JOIN Books
ON AuthorID = Authors.ID
```

ملحوظة:
لإنشاء الجملة السابقة باستخدام باني الاستعلام، حدّد الحقل Book من جدول الكتب، والحقل Author من جدول المؤلفين.. هذا كل شيء! ولو كنت اخترت الربط الأيسر، أو الربط الأيمن، فاضغط رمز العلاقة بزر الفأرة الأيمن، ومن القائمة الموضعية أعد ضغط الأمر Select All Rows From Books، أو Select All Rows From لإزالة اختياره.

لاحظ أن الربط الأيسر والربط الأيمن والربط الكامل تعتبر جميعا أنواعا من الربط الخارجي Outer Join.. لهذا يمكنك أن تضيف الكلمة OUTER بعد كل من LEFT JOIN و RIGHT JOIN و FULL JOIN، وهو ما يفعله باني الاستعلام تلقائيا، فهو يضيف الكلمة OUTER حتى لو لم تكتبها أنت.

استعلامات الأداء Action Queries:

يعتبر هذا النوع من الاستعلامات أبسط من استعلامات التحديد، فهو لا يقوم باسترجاع أيّ سجلات.. ولكنه في المقابل يغير بعض بيانات الجدول، سواء بتحديث قيم السجلات أو بإضافة سجلات جديدة أو بحذف بعض السجلات الموجودة.

ويعيد استعلام الأداء عدد السجلات التي تأثرت بالعملية (وليس السجلات نفسها).

ملحوظة:
لن تستطيع تجربة هذه الاستعلامات في نافذة باني الاستعلام الخاصة بالعروض View، لأنها كما ذكرنا تستخدم استعلامات التحديد لعرض جزء من جدول أو أكثر.. وبدلا من هذا، يمكنك ضغط العنصر Views في متصفح الخوادم Server Explorer بزر الفأرة الأيمن وضغط الأمر New Query لفتح نافذة باني استعلام خاصة باستعلام عام.. في هذه الحالة يمكنك تجربة أمثلة استعلامات الأداء بلصقها في جزء الاستعلام SQL PANE في هذه النافذة، وضغط الأمر Execute SQL من القائمة الموضعية.

حذف الصفوف باستخدام الأمر DELETE:

تحذف هذه الجملة أيّ عدد تريده من الصفوف تبعا للشرط الذي تحدّده في المقطع WHERE.. والجملة التالية تريك كيف يمكن حذف كل الكتب نفدت:

```
DELETE FROM Books
WHERE Copies_No = 0
```

ملحوظة:
لبناء أمر الحذف السابق في باني الاستعلام، أضف الجدول Books إلى قسم المخطط، واضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمة الموضعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Delete.. ستجد أن جملة SQL المكتوبة قد تحولت إلى DELET بدلا من SELECT.

استخدم قسم المعايير Criteria Pane لإضافة اسم الحقل Copies_No، وضع في العمود Filter القيمة:

= 3

والجملة التالية تحذف كل الكتب التي كتبها (عباس العقاد):

```
DELETE FROM Books
WHERE AuthorID = (SELECT ID
                  FROM Authors
                  WHERE Author = 'عباس العقاد')
```

طبعا لاحظت استخدامنا لاستعلام التحديد في جملة الشرط.. هذه الإمكانيّة تمنحك قدرات بلا حدود، لحذف السجلات التي تنطبق عليها المواصفات التي تحدّدها.

وهناك طريقة أخرى لأداء نفس العملية، وذلك باستخدام الفقرة From لاختيار الجداول ووضع شروط الربط بينها باستخدام الفقرة WHERE أو طرق الربط المختلفة Joins.. دعنا نكتب المثال السابق باستخدام الربط الداخلي:

```
DELETE FROM Books
FROM Books INNER JOIN
     Authors ON Books.AuthorID = Authors.ID
WHERE (Authors.Author = 'عباس العقاد')
```

وهو نفس ما يمكنك فعله بالجملة WHERE كالتالي:

```
DELETE FROM Books
FROM   Books, Authors
WHERE AuthorID = Authors.ID AND
Authors.Author = 'عباس العقاد'
```

ملحوظة:

لبناء أمر الحذف السابق في باني الاستعلام، اضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمة الموضعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Delete.

أضف الجدول Books أولا (ليتم الحذف منه) ثم أضف الجدول Authors ليقوم باني

الاستعلام بتكوين فقرة الربط بينهما.
وفي قسم المعايير Criteria Pane اختر اسم الحقل Authors.Author من العمود Column وفي العمود Filter الخاص بهذا الحقل ضع الشرط:
'عباس العقاد' =

إدراج سجلات جديدة باستخدام الأمر INSERT:

يمكنك إضافة سجلات جديدة إلى الجدول، باستخدام أمر الإدراج INSERT على الصيغة التالية:

INSERT INTO اسم الجدول (أسماء الحقول) VALUES (قيم الحقول)

حيث ستضاف القيم إلى الحقول تبعا لترتيب كتابة أسمائها.. ويمكن ألا تكتب كل أسماء الحقول، وفي هذه الحالة ستترك الخانات المناظرة للحقول فارغة.. ويجب أن يكون عدد القيم المرسلة مساويا لعدد أسماء الحقول المكتوبة.. والمثال التالي يضيف سجلا جديدا إلى جدول المؤلفين:

```
INSERT INTO Authors (Author, CountryID, About)
VALUES ('فاروق جويدة', 21, 'شاعر مصري معاصر')
```

لاحظ أننا لم نرسل قيمة إلى الحقل ID لأنه ترقيم تلقائي، ولن يقبل منك أي قيمة.. كما أننا لم نرسل قيمة حقل رقم الهاتف، لأننا صممناه بحيث يقبل القيمة الفارغة NULL، ولا مانع من إرسال قيمة له لو أردت.. لاحظ أيضا أن الرقم ٢١ هو رقم مصر في جدول الدول.

وهناك تسهيل آخر، يتيح لك إدخال قيم كل الحقول دون أن تكتب أسماءها، لكن في هذه الحالة عليك أن تكتب القيم مرتبة تبعا لترتيب الحقول الأصليّ في الجدول، وذلك على الصيغة التالية:

INSERT INTO اسم الجدول VALUES (قيم الحقول)

فمثلا: لإضافة حقل جديد إلى جدول الكتب، استخدم الجملة التالية:

```
INSERT INTO Books
VALUES ('كانت لنا أوطان', 14, 6, 7, '1/8/2000', 3, 2, 1000)
```

لاحظ أننا لم نضع قيمة للحقل ID لأنه يُولد تلقائيّا، لهذا ستوضع القيم بالترتيب في الحقول التالية لهذا الحقل.. لكن المشكلة أنك لو لصقت الجملة السابقة في باني الاستعلام، فسيضيف أسماء الحقول تلقائيا إليها، ومن ضمنها الحقل ID، لهذا عليك أن تحذفه وإلا حدث خطأ عند تنفيذ الاستعلام.

لاحظ أيضا ضرورة وضع التاريخ بين العلامتين ' ' عند كتابة جمل SQL، تماما كما نفعل مع النصوص.. كما أنك تستطيع استخدام الكلمة NULL كقيمة، إذا أردت ترك الخانة فارغة، وإذا كانت الخانة تحتوي على بيانات ثنائية (كالصورة image) فأرسل إليها القيمة 0x0 وهي تعادل العدم NULL.

ملحوظة:

لبناء أمر الإدراج في باني الاستعلام، أضف الجدول الذي ستقوم بالإدراج فيه إلى قسم المخطط، واضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمة الموضعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Insert Values.. ستجد أن جملة SQL المكتوبة قد تحولت إلى INSERT بدلا من SELECT، وستكون على هذه الصورة:

INSERT INTO اسم الجدول ( )
VALUES ( )

اختر أسماء الحقول التي تريد الإدراج فيها بوضع علامة الاختيار بجوارها في مخطط الجدول، لتظهر أسماؤها في جملة SQL داخل القوسين الفارغين التاليين لاسم الجدول.. بعد هذا عليك أن تضيف القيم بكتابتها في العمود New Value في قسم المعايير Criteria Pane، حيث ستكتب القيمة الجديدة في الخانة المجاورة لاسم كل حقل.

ولو أردت العودة إلى استخدام الأمر SELECT، فاضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمة الموضعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Select.

ويتيح لك الأمر INSERT نسخ مجموعة من السجلات من جدول إلى آخر، بشرط أن يكون لحقولهما نفس نوع البيانات، وذلك باستبدال الفقرة VALUES بالجملة SELECT.

افترض أن عندنا جدولا اسمه TempBooks مماثلا في تركيبه لجدول الكتب، ونريد أن نضع فيه بعض الكتب مؤقتا لأيّ سبب.. في هذه الحالة يمكن أن ننسخ فيه كتب (توفيق الحكيم) بالجملة التالية:

```
INSERT INTO TempBooks
SELECT Books.*
FROM Authors, Books
WHERE Author = 'توفيق الحكيم' AND AuthorID = Authors.ID
```

ملحوظة:

لبناء الأمر الأخير في باني الاستعلام، أضف الجدولين Authors و Books إلى قسم المخطط، وحدد كل الحقول من جدول الكتب، وحدد الحقل Author من جدول المؤلفين، وفي قسم المعايير Criteria Pane أزل علامة الاختيار من العمود Output في صف الحقل Author لكي لا يظهر في النتائج، وفي الخانة Filter الخاصة بهذا الحقل أضف الشرط: ('توفيق الحكيم' =)

الآن لديك جملة التحديد SELECT التي تحصل على النتائج التي تريد إدراجها في الجدول TempBooks.. كل ما عليك فعله الآن، هو الضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمة الموضعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Insert Results.. ستظهر لك نافذة تتيح لك اختيار الجدول الذي تريد إدراج النتائج فيه.. اختر الجدول TempBooks وأغلق النافذة.. الآن ستجد جملة الإدراج كاملة في قسم الاستعلام SQL Pane!

تحرير السجلات الموجودة باستخدام الأمر UPDATE:

لتغيير قيم بعض ــ أو كل ــ سجلات أحد الجداول، استخدم الجملة UPDATE، التي لهـا الصيغة التالية:

اسم الجدول UPDATE
القيمة٢ ,الحقل٢ = القيمة١ ,الحقل١ = ... SET
شرط WHERE

مثال: الجملة التالية تغير اسم المؤلف (نبيل فاروق) إلى (نبيل فاروق رمضان):

```
UPDATE Authors
SET Author = 'نبيل فاروق رمضان'
WHERE Author = 'نبيل فاروق'
```

ملحوظة:
لبناء الأمر الأخير في باني الاستعلام، أضف الجدول Authors إلى قسم المخطط، وحـدد الحقل Author. اضغط بزر الفأرة الأيمن في أي موضع خال من قسم المخطط، ومن القائمـة الموضـعية اضغط Change Type ومن القائمة الفرعية اضغط الأمر Update، وفي قسم المعـايير Criteria Pane أضف الشرط التالي في الخانة Filter الخاصة بالحقل Author: ('نبيـل فاروق' =) وفي العمود New Value السابق للعمود Filter اكتب: 'نبيل فاروق رمضان' الآن ستجد الأمر UPDATE مكتوبا في قسم SQL كاملا.

التحرير المتتابع للسجلات، باستخدام الأمر UPDATE .WRITE:

عند التعامل مع أعمدة تحتوي على بيانـات مـن النـوع text أو ntext أو image فـي سيكيويل سيرفر ٢٠٠٠، كانت تواجهنا مشكلة خطيرة، وهي كيفية إرسال كم ضـخم مـن البيانات من العميل إلى الخادم لحفظه في هذه الأعمدة.. افرض أنك تريـد وضـع صـورة حجمها ١٠ ميجا في خانة من النوع image بالاستعلام التالي:

```
UPDATE Publishers
SET Logo = @Logo
WHERE ID = 1
```

حيث Logo@ هو معامل من النوع image يحمل بيانات الصورة.. لو حاولت تنفيذ هذا الاستعلام، فسيستغرق إرسال الصورة إلى الخادم وقتا ملموسا (تبعا لسرعة الاتصال، وحجم الضغط على الخادم في تلك اللحظة)، وقد يتعطل برنامجك عن الاستجابة، وقد ينفد وقت الانتظار Timeout قبل إتمام العملية.

ويمكنك أن تجرب الاستعلام السابق بضغط الزر Add Logo في المشروع WriteLargeData.. جرب اختيار صورة حجمها كبير لترى تأثير هذا.

وللأسف، لا يفيدك استخدام استعلام كالتالي:

```
UPDATE Publishers
SET Logo = Logo + @Logo
WHERE ID = 1
```

هذا الاستعلام يحاول إضافة جزء من بيانات الصورة موضوع في المعامل Logo@ إلى البيانات الموجودة فعليا في عمود الصورة Logo.. لكن للأسف، سيؤدي هذا إلى حدوث خطأ، لأن سيكويل سيرفر لا يعرف كيف يجمع بيانات من النوع image !

ولحل هذه المشكلة، قدمت سيكويل سيرفر ٢٠٠٥ الأنواع القصوى الجديدة varchar(max), nvarchar(max), varbinary(max) وهي مخصصة للتعامل مع الكائنات الثنائية الضخمة Binary Large Objects (BLOBs)، فهي تستطيع استقبال بيانات يصل حجمها إلى حوالي ٢ جيجا بايت، كما يمكنها التعامل مع البيانات الثنائية والحروف بطريقة تتابعية Sequential، أي أنك تستطيع الكتابة فيها في أي موضع، أو القراءة منها من أي موضع.

لهذا إذا كنت تنوي التعامل مع بيانات ضخمة BLOBs، فالأفضل أن تستخدم:

- النوع varchar(max) بدلا من النوع Text.
- والنوع nvarchar(max) بدلا من النوع ntext.
- والنوع varbinary(max) بدلا من النوع image.

وللكتابة التتابعية في هذه الأنواع الجديدة، قدمت T-SQL الصيغة الجديدة التالية لأمر التحديث:

UPDATE اسم_الجدول

SET اسم_العمود .WRITE (@Value, @Offset, @Length)

Where شرط

فالدالة الداخلية Write تكتب البيانات المرسلة إلى المعامل @Value في الخانة المحددة في العمود، بدءا من الموضع @Offset، وبحيث يكون طول البيانات المكتوبة @Length.. لاحظ ما يلي:

- هذه الصيغة ستسبب خطأ إذا حاولت التعامل مع أي نوع بيانات غير الأنواع القصوى التي تنتهي بـ (MAX).
- هذه الصيغة ستسبب خطأ إذا حاولت إضافة بيانات في خانة قيمتها Null.. لهذا أمامك حلان:

١- فإما أن تضيف الجزء الأول من البيانات باستخدام الصيغة العادية للأمر Update، ثم تضيف باقي أجزاء البيانات باستخدام الصيغة Update .Write.

٢- وإما أن تمنع استخدام القيمة Null في خانات العمود، وفي هذه الحالة عليك استخدام قيمة افتراضية لوضعها في الخانات الفارغة، وذلك بوضع القيمة صفر في الخاصية Default Value Or Binding في خصائص العمود.. ولكي لا تؤثر هذه القيمة الافتراضية على القيمة التي ستضعها في خانات العمود، يجب أن تكون للمعامل @Offset القيمة صفر عند كتابة أول جزء من البيانات، لتوضع في بداية الخانة بدلا من أية بيانات موجودة.

- إذا أرسلت إلى المعامل @Value القيمة Null، فسيتم تجاهل المعامل @Length، وسيتم حذف القيمة الموجودة في الخانة في الموضع المحدد في المعامل @Offset.

- إذا أرسلت إلى المعامل @Offset القيمة Null، فسيتم تجاهل المعامل @Length، وستضاف قيمة المعامل @Value إلى نهاية البيانات الموجودة حاليا في الخانة.
- إذا أرسلت إلى المعامل @Offset قيمة أكبر من طول البيانات الموجودة في الخانة، فسيحدث خطأ.. ولكي تكتب بعد نهاية البيانات الموجودة، يجب أن ترسل إلى المعامل @Offset قيمة تساوي طول البيانات الموجودة في الخانة، أو ترسل إليه القيمة Null كما أوضحنا في الملحوظة السابقة.
- إذا أرسلت إلى المعامل @Length القيمة Null، فسيتم حذف جميع البيانات التالية للموضع المحدد في المعامل @Offset.

وستجد مثالا لاستخدام هذه الصيغة لكتابة بيانات صورة أول ناشر في العمود Logo2 في الجدول Publishers بطريقة تتابعية، وذلك في الزر Write. Update في المشروع WriteLargeData، وفيه نستخدم الاستعلام:

```
UPDATE Publishers
SET Logo2 .WRITE (@Logo, @Offset , @Length)
WHERE ID = 1
```

لاحظ أن العمود Logo2 من النوع varbinary(MAX)، وأن قيمته الافتراضية هي ٠٠. ولكتابة بيانات الصورة في الخانة الأولى من هذا العمود تتابعيا، سنرسل أول ١٠٠ وحدة ثنائية Byte لحفظها في الخانة بدءا من الموضع رقم صفر، ثم نرسل ١٠٠ وحدة تالية لحفظها في الخانة بدءا من الموضع رقم ١٠٠، ونستمر في فعل هذا إلى أن ننتهي من كتابة بيانات الصورة.. هذا معناه أننا سنستخدم الاستعلام Update. Write عدة مرات (وذلك من خلال حلقة تكرار Loop)، لكن مع تغيير المعاملات المرسلة إلى الدالة الداخلية Write. في كل مرة.. طبعا إرسال ١٠٠ وحدة ثنائية في كل مرة، أفضل بكثير من إرسال ١٠ ميجا أو أكثر دفعة واحدة.. لكن هذا قد يصير عبئا خطيرا على البرنامج إذا كانت الصورة ضخمة جدا، بسبب زيادة عدد مرات إرسال البيانات من العميل إلى الخادم.. لهذا عليك اختيار حجم مناسب لأجزاء البيانات التي ترسلها، بحيث لا يكون كبيرا جدا فيدمر الذاكرة ويبطئ عملية الإرسال، ولا يكون صغيرا جدا فيؤدي إلى إضاعة وقت كبير من

الخادم بسبب كثرة عدد الأوامر المرسلة إليه من عميل واحد.. ربما يكون الأنسب مثلا أن تستخدم ١٠٢٤ وحدة ثنائية (١ كيلو بايت) لكل جزء.. لكن هذا سيجعلك ترسل الصورة التي حجمها ١ ميجا فقط على حوالي ١٠٠٠ مرة.. ما زال هذا يبدو كثيرا.. أليس كذلك؟.. لو شئت رأيي، فإن إرسال ١٠٠ كيلو في كل مرة سيكون مناسبا، فهذا سيرسل الصورة التي حجمها ١ ميجا باستخدام ١٠ أوامر فقط وهذا ليس كثيرا، كما أن ١٠٠ كيلو ليس بالحجم المقلق الذي يستغرق وقتا ملموسا عند إرساله إلى الخادم.. لكن عليك في هذه الحالة أن تجعل للخاصية Timeout الخاصة بالاتصال الذي تستخدمه قيمة أكبر قليلا، ولتكن ١٢٠ ثانية على سبيل الاحتياط.. وسنعرف كيف نفعل هذا لاحقا عند دراسة كائن الاتصال وكائن الأمر.

ملحوظة:
الإصدارات القديمة من T-SQL، كانت تستخدم الأوامر UPDATETEXT و READTEXT و WRITETEXT للتعامل مع البيانات الضخمة الموجودة في الأعمدة من النوع image أو text أو ntext.. لكن لا ينصح باستخدام هذه الأوامر الآن لأنها ستزال من لغة الاستعلام، وبدلا من هذا عليك استخدام الأنواع القصوى، والصيغة .Update .Write

لغة تعريف البيانات Data Definition Language (DDL):

تختصّ أوامر هذه اللغة بإنشاء أو حذف كائنات قاعدة البيانات، مثل الجداول وما تحتويه من أعمدة وفهارس، ومثل العلاقات Relations والقيود Constrains.

ولن نتوسّع في شرح هذه الأوامر في هذا الكتاب، وسنتركها إلى الكتاب الذي نشرح فيه T-SQL بإذن الله، ويمكنك أن تبحث في ملفات استعلام اللغة عن كيفيّة استخدام الأوامر التالية، وهي تستخدم لإنشاء عناصر قاعدة البيانات:

CREATE DATABASE
CREATE TABLE
CREATE PROCEDURE
CREATE FUNCTION
CREATE VIEW
CREATE INDEX

ولتعديل عناصر قاعدة البيانات، تستخدم الكلمة ALTER بدلا من الكلمة CREATE..

ولحذف أحد العناصر، تستخدم الكلمة DROP بدلا من الكلمة CREATE.

ويمكنك الاسترشاد بهذه الأمثلة (وهي خاصة بقواعد بيانات Access):

الجملة التالية تنشئ جدولا للمؤلفين، به عمودان يستوعبان نصوصا لا يزيد طولها عن ٣٠ حرفا، مع ملاحظة أن العمود الثاني مفهرس (بإنشاء فهرس اسمه X) مع عدم السماح بتكرار القيم:

```
Create Table Authors
      ([Author] Text  (30), [City] Text (20) Constraint X Unique)
```

أمّا الجدول التالي، فله عمود رقميّ، والآخر ترقيم تلقائيّ:

```
Create Table Numbers
      ([No] Integer,[ID] Counter)
```

ولإنشاء فهرس اسمه X على العمود ID في الجدول السابق:

```
Create Unique Index X On Numbers (No)
```

ولإضافة عمود اسمه Y للجدول السابق:

```
ALTER TABLE  Numbers
      ADD COLUMN Y Long
```

ولحذفه مرّة أخرى:

```
ALTER TABLE  Numbers DROP COLUMN Y
```

الإجراءات المخزّنة Stored Procedures:

الإجراءات المخزّنة هي برامج قصيرة مكتوبة بلغة T-SQL محفوظة في قاعدة البيانات، ويتم تنفيذها على الخادم Server، لأداء وظيفة معيّنة على قاعدة البيانات.

ويمتاز الإجراء المخزن بالمميزات التالية:

١- يقوم SQL Server بترجمة Compile الإجراءات المخزّنة مرّة واحدة عند إنشائها، وبالتالي فهي تعمل أسرع من جمل SQL العاديّة.

٢- اختصار الكود، بسبب قدرتك على استدعاء نفس الإجراء المخزن من برامج مختلفة أو من أكثر من موضع من نفس البرنامج.

٣- سهولة التعديل في الإجراء المخزن بدون تغيير أكواد البرامج التي تستخدمه، وذلك لأنه موجود في قاعدة البيانات.

٤- الإجراءات المخزنة أكثر مناعة ضد دس الاستعلامات SQL Injection كما سنرى فيما بعد.

وتسمّى اللغة التي تكتب بها الإجراءات المخزّنة T-SQL اختصارا لـ (Transact-SQL)، وهي نسخة مطوّرة من SQL خاصة بسيكيول سيرفر كما ذكرنا سابقا.

ويبدأ الإجراء المخزن بجملة التعريف التالية:

CREATE PROCEDURE dbo.SP1

حيث إن SP1 هو اسم الإجراء، ويمكنك تغييره إلى أيّ اسم يناسبك.

لاحظ أن الجملة CREATE PROCEDURE تستخدم عند إنشاء الإجراء المخزن للمرة الأولى، لكن بمجرد حفظه تتحول إلى ALTER PROCEDURE، لأنك ستعدل فيه بعد ذلك ولن تعيد إنشاءه من جديد.. ولو حاولت استخدام CREATE PROCEDURE مع نفس اسم الإجراء المخزن مرة أخرى فستحصل على خطأ يخبرك أن هذا الإجراء موجود من قبل، لهذا لو كنت تقصد إعادة حفظ نفس الإجراء بعد التعديل، فتأكد من أنه يبدأ بالفقرة ALTER PROCEDURE.

بعد هذا يأتي تعريف معاملات الإجراء.. وتبدأ المتغيّرات في لغة T-SQL بالرمز @..

والجملة التالية تعرّف معاملا نصّيا طوله ثلاثة أحرف:

@Str1 nvarchar(3)

كما ترى: تكتب اسم المعمل، يليه نوعه.

وعند وجود أكثر من معامل، توضع العلامة "," للفصل بينها.

ويمكن أن تضع للمعامل قيمة افتراضيّة، بحيث لو لم يُرسل هذا المعامل، يقــوم الإجــراء باستخدام هذه القيمة:

@Str2 nvarchar(7) = 'افتراضي'

وتوضع معاملات الإجراء بينَ قوسين ()، ولا يشترط أن تكتب في سطر واحد.

ويبدأ الإجراء المخزّن بالكلمة AS، وينتهي بالكلمة RETURN، وبينهما الكود:

```
AS
……….
RETURN
```

ويمكنك أن تعرّف المتغيّرات داخل الإجراء، باستخدام الجملة DECLARE:

```
DECLARE @Str3 nvarchar(7)
```

ويمكنك كتابة جمل الشرط كالتالي:

```
IF @Str3 = ''
BEGIN
    SELECT @Str3 = 'ABC'
END
```

وإذا كان مقطع الشرط جملة واحدة، فلسنا بحاجة إلى BEGIN و END:

```
IF @Str3 = ''
   SELECT @Str3 = 'ABC'
```

لاحظ استخدامنا للتعبير SELECT لتغيير قيمة المتغيّر @Str3.. يمكنك كــذلك اســتخدام التعبير SET لتغيير قيمة المتغيّر، وهو ليس مربكا مثل SELECT:

```
SET @Str3 = 'ABC'
```

ويمكن أن تضع في المتغيّر ناتج جملة SELECT ــ طبعا إذا كانت تعيد قيمة واحدة، كأن تكون بها دالة من دوال التجميع Aggregate Functions أو يكون الجدول الناتج عبــارة عن خانة واحدة ــ كالتالي:

```
SET @Str3 = (SELECT Book FROM Books WHERE ID = 3)
```

والآن، ما رأيك أن نكتب إجراء مخزّنا نرسل إليه اسم المؤلف، فيعيد إلينا الكتب التي ألفها؟

في متصفّح خوادم الإنترنت Server Explorer، أسدل العناصر التي تنتمي لاتصال قاعدة البيانات Books.mdf التي أنشأناها بـ SQL Sever.. من هذه العناصر اضغط بزرّ الفأرة الأيمن على "الإجراءات المخزّنة" Stored Procedures، ومن القائمة الموضعيّة اضغط الأمر New Stored Procedure.. ستظهر لك نافذة تحرير الإجراء المخزّن.. اكتب بها هذا الإجراء:

```
CREATE PROCEDURE dbo.GetAuthorBooks
        (
                لدينا معامل واحد فقط نستقبل فيه اسم المؤلف*/
                /* وقد جعلناه ٣٠ حرفا ليتلاءم مع طول حقل المؤلفين
              @Author char(30)
        )
AS
IF @Author <>''      /*تأكّد أن اسم المؤلف غير فارغ */
BEGIN
        /*هذه جملة استعلام عاديّة تماما*/
        SELECT Book
        FROM Books, Authors
        WHERE Author = @Author
                AND AuthorID = Authors.ID
END
RETURN /*وجود هذه الكلمة في أيّ موضع يؤدّي إلى إنهاء الإجراء*/
```

أعتقد أن الأمر في غاية البساطة.

ولا حاجة بي لألفت نظرك إلى أن التعليق في لغة T-SQL يوضع بين العلامات /* */.. هذه الرموز تتيح لك كتابة التعليق على أكثر من سطر.. فإذا أردت أن تكتب سطرا واحدا كتعليق، فيمكنك أن تضع في بدايته العلامتين --.. مثال:

```
/* هذا تعليق
/*على سطرين
--بينما هذا تعليق في سطر واحد
```

ولا بدّ أنك لاحظت ظهور الجملة SELECT في نافذة الإجراء السابق في مستطيل.. اضغط بزرّ الفأرة الأيمن داخل هذا المستطيل، ومن القائمة الموضعيّة اختر الأمر "تصميم مقطع الاستعلام" Design SQL Block.. ستظهر لك نافذة باني الاستعلام Query Builder، حيث يمكنك استغلاله لتطوير الجملة واختبارها.

كما يمكنك استخدام باني الاستعلام لتصميم جملة SQL مباشرةً منذ البداية.. اضغط بزرّ الفأرة الأيمن في الموضع الذي تريد كتابة الجملة فيه في الإجراء المخزّن، ومن القائمة الموضعيّة اضغط الأمر Insert SQL.. ستظهر لك نافذة باني الاستعلام.. صمّم جملة SQL التي تريد، وقم بحفظ العمل.. الآن لو عدت إلى نافذة الإجراء المخزّن، فستجد أن جملة SQL قد أضيفت للموضع الذي ضغطت فيه الفأرة.

نريد الآن اختبار الإجراء الذي كتبناه.. اضغط بزرّ الفأرة الأيمن في أيّ موضع من نافذة الإجراء، ومن القائمة الموضعيّة اضغط الأمر Run Stored Procedure.. ستظهر لك نافذة تسألك عن قيمة معامل الإجراء.. اكتب اسم (توفيق الحكيم) واضغط OK.. سيتمّ تنفيذ الإجراء، حيث ستظهر أسماء كتب (توفيق الحكيم) في نافذة المخرجات Output Window.

إنشاء الإجراءات المخزنة في قواعد بيانات Access:

على عكس ما يظنه الكثيرون، يمكن استخدام الإجراءات المخزنة مع قواعد بيانات آكسيس.. لكن المشكلة أنك لن تجد طريقة لإنشائها في آكسيس نفسه، كما أنك لا تستطيع إنشائها في متصفح الخوادم Server Explorer في دوت نت، رغم أنك ستجد عنصرا اسمه Stored Procedures ضمن عناصر قاعدة بيانات الكتب Books.mdb!

فما هو حل هذا اللغز العجيب يا ترى؟

الحل هو تنفيذ الجملة التي تنشئ الإجراء المخزن باستخدام الكود في أحد برامجك لمرة واحدة، وبهذا يتم إنشاء الإجراء المخزن في قاعدة بيانات آكسيس، ويظهر في متصفح الخوادم تحت العنصر Stored Procedures.. طبعا لا نستطيع فعل هذا الآن، لهذا

سنؤجله إلى فصل لاحق.. لكن علينا هنا أن نوضح بعض الاختلافات الجوهرية بين الإجراء المخزن الخاص بآكسيس والإجراء المخزن الخاص بسيكويل سيرفر:

١- يستخدم آكسيس التعبير CREATE PROC بدلا من التعبير CREATE PROCEDURE.

٢- الإجراء المخزن في آكسيس يتكون من جملة SQL واحدة فقط، لأن آكسيس لا يقبل تنفيذ أكثر من جملة مفصولة بالعلامة ;.

٣- لا يمكن تعريف متغيرات في الإجراء المخزن في آكسيس ولا يمكن استخدام جمل الشرط.. السبب ببساطة أن آكسيس لا يتعامل مع T-SQL.

٤- لا توضع العلامة @ قبل أسماء المعاملات الخاصة بالإجراء المخزن في آكسيس.. وطبعا أنواع هذه المعاملات هي الأنواع الخاصة بآكسيس.

٥- لا ينتهي الإجراء المخزن في آكسيس بالكلمة RETURN.

باختصار: الإجراء المخزن في آكسيس، هو مجرد استعلام سيكويل مخزن!

هكذا مثلا سيكون الإجراء المخزن الذي يحذف من جدول المؤلفين، المؤلف الذي ترسل اسمه إليه:

```
CREATE PROC DeleteAuthor(
                AuthorName VARCHAR(20))
AS
DELETE FROM Authors
        WHERE (Authors.Author = AuthorName);
```

ولكي تضيف هذا الإجراء إلى قاعدة بيانات الكتب، افتح المشروع AccessStoredProcedure، وشغل البرنامج، واضغط الزر "DeleteAuthor".. كما يمكنك إنشاء إجراء مخزن للحصول على كتب أحد المؤلفين، بضغط الزر "GetAuhorBooks".

لاحظ أن هذا البرنامج يفترض وجود الملف Books.mdb على المسار C:\.. وسنفهم الكود الذي يستخدمه هذا البرنامج بالتفصيل لاحقا.

بعد أن تنفذ هذا البرنامج، أضف اتصالا بقاعدة بيانات الكتب الموجودة على المسار C:\ في متصفح الخوادم، وأسدل عناصرها.. ستجد أن الإجراءين DeleteAuthor و GetAuhorBooks قد ظهرا تحت العنصر Stored Procedures.. لكن للأسف، لا يمنحك متصفح الخوادم أية طريقة لعرض كود هذين الإجراءين أو تعديلهما أو تشغيلهما.
ويريك الزر "الكتب١" في المشروع Factories كيف يمكن استدعاء الإجراء GetAuhorBooks من برنامجك، وسنفهم كيف يفعل هذا لاحقا.

دوال SQL التي يعرفها المستخدم User Defined SQL Functions:

يمكنك تعريف دوال خاصة، تستخدمها داخل استعلامات SQL، بنفس الطريقة التي استخدمنا بها الدالة RIGHT.. هذه الدوال تحفظ في قاعدة البيانات تماما كالإجراءات المخزنة، وتظهر في متصفح الخوادم تحت العنصر Functions.
ويمكنك إنشاء دالة جديدة بإسدال عناصر قاعدة البيانات وضغط العنصر Functions بزر الفأرة الأيمن، وضغط القائمة الفرعية New، حيث ستظهر لك ثلاثة أنواع من الدوال التي تستطيع إنشاءها.. لكن نظرا لأن هذا الموضوع يحتاج إلى تفاصيل أعمق في T-SQL فسنؤجله إلى الوقت المناسب بإذن الله.

أنواع الجداول التي يعرفها المستخدم User-Defined Table Types:

قدم سكيويل سيرفر ٢٠٠٨ نوعا جديدا من المعاملات، قادرا على استقبال جدول كامل يحتوي على مجموعة صفوف، بدلا من المعاملات العادية التي تستقبل قيمة خانة منفردة في أحد الصفوف.. هذا يغنيك عن استخدام عدد كبير من المعاملات عند تعريف الإجراء المخزن، كما يغنيك عن استدعاء نفس الإجراء المخزن أكثر من مرة.. فمثلا: لو أردت كتابة إجراء مخزن يقوم بتحديث قيم ٥ صفوف بها عشرة أعمدة في أحد الجداول، فإن استخدام المعاملات التقليدية سيجبرك على تعريف ١٠ معاملات للإجراء المخزن (واحد لكل عمود)، واستخدام واستدعاء الإجراء المخزن ٥ مرات لإرسال القيم إلى هذه الصفوف.. بينما إذا استخدمت معاملا جدوليا Table-Valued Parameter، فستعرف معاملا واحدا فقط للإجراء المخزن، وستستدعيه مرة واحدة فقط لتحديث الصفوف الخمسة.

يبدو هذا مريحا للغاية، حيث سيوفر عليك كتابة الكثير من الكود في الإجراء المخزن وفي برنامجك أيضا، وسيجعل نقل كميات كبيرة من البيانات أمرا في غاية البساطة.

ولاستخدام هذا النوع من المعاملات، يجب أن تقوم بتعريفه أولا في قاعدة البيانات.. ويشبه تصميم النوع تصميم الجدول، لكن للأسف، لا توجد طريقة مرئية لتصميم النوع، لهذا عليك كتابة كود T-SQL لفعل هذا، وتنفيذه من نافذة تنفيذ الاستعلامات في مدير سيكويل.. ويستخدم الأمر CREATE TYPE لإنشاء نوع جديد في T-SQL، وذلك على الصيغة:

CREATE TYPE اسم_النوع AS TABLE
(تعريف_الأعمدة)

دعنا ننشئ نوعا اسمه AuthorType نستطيع من خلاله تحديث صفوف جدول المؤلفين.. لفعل هذا في مدير سيكويل، اضغط اسم قاعدة البيانات Books.mdf بزر الفأرة الايمن، ومن القائمة الموضعية اضغط الأمر New Query، وفي نافذة الاستعلام اكتب الكود التالي:

```
CREATE TYPE AuthorType AS  TABLE
   ( ID int,
     Author nvarchar(50),
     CountryID smallint,
     Phone varchar(20),
     About nvarchar(MAX) )
```

من القائمة الموضعية اضغط Execute.. الآن سيظهر النوع AuthorType تحت العنصر Programmability\Types\User defined Table Types في متصفح الكائنات في مدير سيكويل سيرفر، حيث يمكنك تغيير اسمه لو أردت، كما يمكنك حذفه في أي وقت.

الآن يمكنك تعريف إجراء مخزن جديد يستخدم معاملا من هذا النوع.. لاحظ أنك لا تستطيع استخدام هذا المعامل للإخراج، كما أنك مجبر على تعريفه للقراءة فقط باستخدام الكلمة READONLY كالتالي:

```
CREATE PROCEDURE UpdateAuthors
   (@Rows AuthorType READONLY)
```

لكن كيف نتعامل مع المعامل @Rows في كود الإجراء المخزن؟

الأمر بسيط للغاية، فهذا المعامل هو جدول، ويمكنك التعامل معه في جمل SQL كأي جدول عادي.. مثلا: يمكننا تحديث الجدول Authors من الجدول @Rows باستخدام الأمر UPDATE والربط الداخلي INNER JOIN كالتالي:

```
CREATE PROCEDURE UpdateAuthors
   (@Rows AuthorType READONLY)
AS
UPDATE Authors
   SET Authors.Author = R.Author,
         Authors.CountryID = R.CountryID,
         Authors.Phone = R.Phone,
         Authors.About = R.About
   FROM Authors INNER JOIN @Rows AS R
   ON Authors.ID = R.ID;
RETURN
```

لاحظ ضرورة تعريف اسم مستعار للجدول Rows@ في المقطع FROM لتسهيل كتابة الاستعلام.. لقد منحناه هنا الاسم المستعار R.

بهذا يمكنك استخدام هذا الإجراء المخزن لتحديث قيم سجلات مؤلفين موجودين بالفعل في قاعدة البيانات، بناء على تماثل الرقم ID في السجل المرسل والسجل الأصلي.

كما يمكنك تعريف إجراء لإدراج سجلات الدول المرسل كمعامل، في جدول المؤلفين، وذلك باستخدام الأمر INSERT.. هذا هو:

```
CREATE PROCEDURE InsertAuthors
   (@Rows AuthorType READONLY)
AS
INSERT INTO  Authors
   (Author, CountryID, Phone, About)
SELECT R.Author, R.CountryID, R.Phone, R.About
FROM @Rows AS R
RETURN
```

لاحظ أنك لا تستطيع اختصار جملة الإدراج السابقة كالتالي:

```
INSERT INTO  Authors
SELECT R.*
FROM @Rows AS R
```

السبب في هذا هو أنك لا تستطيع نسخ قيمة الحقل ID من سجلات المعامل إلى سجلات الجدول الأصلي، لأن الحقل Author.ID حقل يولد تلقائيا، بينما نحن لم نعرف الحقل AuthorType.ID على أنه يولد تلقائيا.

لاحظ أن تعريف المعامل الجدول بالكلمة ReadOnly يمنعك عن إجراء أية تعديلات على قيم سجلاته، كما يمنعك من حذف أي من هذه السجلات أو الإضافة إليها.. لهذا إذا كنت مضطرا إلى فعل هذا، فعليك أن تعرف متغيرا من نفس نوع الجدول، وتدرج فيه سجلات المعامل باستخدام الأمر INSERT.. مثال:

```
DECLARE @TEMP AuthorType;
INSERT INTO  @TEMP
SELECT R.*
FROM  @Rows AS R;
```

لاحظ أننا استخدمنا صيغة الإدراج المختصرة (لم نكتب أسماء الأعمدة) لأن الحقــل ID لا يولد تلقائيا هنا.

بعد هذا يمكنك أن تضيف إلى الجدول TEMP@ سجلات جديدة أو تعدل قــيم الســجلات الموجودة، ومن ثم تستخدمه في الكود كما تريد.

والمشروع TableValuedParameters يريك كيف تستخدم كائن الأمر والمعاملات لتنفيذ هذا الإجراء المخزن لإدراج عدد من المؤلفين في جدول المؤلفين.. وسنتعرف على كــائن الأمر Command ومعاملاته في فصل لاحق.

هذا الكتاب صدقة جارية على روح والدي:

أ. حمدي كامل الحديدي غانم

رحمه الله وغفر له وجعل مثواه الجنة

لهذا أرجو من كل من يستفيد به أن يتذكر أن أبي هو الـــذي ربـــاني وعلمنـــي ولولاه بعد توفيق الله ما خرج إلى الوجود هذا الكتاب وغيره من الكتب.

فادعوا له بالرحمة والمغفرة

ومن كان منكم في الحرمين الشريفين وكان قادرا على عمل عمرة له، فجزاه الله خيرا.

أدعو الله أن يكون هذا الكتاب وباقي كتبي من العلم الذي ينتفع به، وأن يجعل الله لأبي نصيبا من ثوابه، فيكون من عمله الذي لا ينقطع بموته.

اللهم ارحم أبي واغفر له وكفر عنه سيئاته وقه من عذاب القبر وقه من عذاب النار،
وأدخله الجنة وأعلِ منزلته فيها
واحفظ والدتي وبارك في عمرها
اللهم ارحم والديّ كما ربياني صغيرا
آمين يا رب العالمين