**الفصل السادس عشر**

**لغة الاستعلام المركّبة**

**Structured Query Language (SQL)**

* [**ما هي لغة الاستعلام المركّبة SQL؟**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\1-%20ما%20هي%20لغة%20الاستعلام%20المركّبة.htm)
* [**باني الاستعلام Query Builder:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام.htm)
* [**استعلامات التحديد Selection Queries:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد.htm)
* [**استعلامات الأداء Action Queries:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\4-%20استعلامات%20الأداء.htm)
* [**لغة تعريف البيانات Data Definition Language (DDL):**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\4-%20لغة%20تعريف%20البيانات.htm)
* [**الإجراءات المخزّنة Stored Procedures:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\16-%20SQL%20لغة%20الاستعلام%20المركّبة\5-%20الإجراءات%20المخزّنة.htm)

**ما هي لغة الاستعلام المركّبة؟**

مهما كان نوع تطبيق قاعدة البيانات الذي تتعامل معه، فإنّه يستخدم لغة خاصّة بقواعد البيانات، متّفق عليها دوليّا، هي SQL.. صحيح أنّ هناك اختلافات في تركيب هذه اللغة ما بينَ تطبيق وآخر، ولكنّها اختلافات طفيفة لا تكفي للادعاء بأنّها نسخ مختلفة تمام الاختلاف.

وبالمقارنة بلغات البرمجة المألوفة، فإنّ SQL لغة غير إجرائيّة Nonprocedural، بمعنى أنّها لا تحتوي على تركيبات لغويةٍ مثل جمل الشرط وجمل التكرار وما شابهها.. إنّ SQL تعتبر لغة برمجة عالية المستوي، حيث يمكنها في سطر واحد إجراء عمليّات بليغة التعقيد على قاعدة البيانات.

وتنقسم جمل لغة SQL إلى طائفتين رئيسيّتين، تعتبر كلّ منهما لغة منفصلة:

**لغة التعامل مع البيانات Data Manipulation Language (DML):**

وتختصّ باسترجاع أو تحديث أو إضافة أو حذف السجلات التي تحقّق شروطا معيّنة، لهذا تسمّى جمل هذه اللغة "استعلامات" Queries.. وتسمّى جمل استرجاع البيانات بـ "استعلامات التحديد" Selection Queries، بينما تسمّى جمل الحذف والتعديل والإضافة بـ "استعلامات الفعل" Action Queries.

**لغة تعريف البيانات Data Definition Language (DDL):**

وتختصّ بإنشاء أو حذف كائنات قاعدة البيانات، مثل الجداول وما تحتويه من أعمدة وفهارس، ومثل العلاقات والقيود Constrains.

**باني الاستعلام**

* [**استخدام باني الاستعلام لتنفيذ جمل SQL:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\1-%20استخدام%20باني%20الاستعلام.htm)
* [**قسم المخطّط Diagram Pane:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\2-%20قسم%20المخطّط.htm)
* [**قسم جدول الحقول Grid Pane:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\3-%20قسم%20جدول%20الحقول.htm)
* [**قسم SQL:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\4-%20قسم%20الأوامر.htm)
* [**قسم النتائج Results Pane:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\5-%20قسم%20النتائج.htm)
* [**مثال:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\6-%20مثال.htm)
* [**إنشاء استعلامات Access:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20باني%20الاستعلام\7-%20إنشاء%20استعلامات.htm)

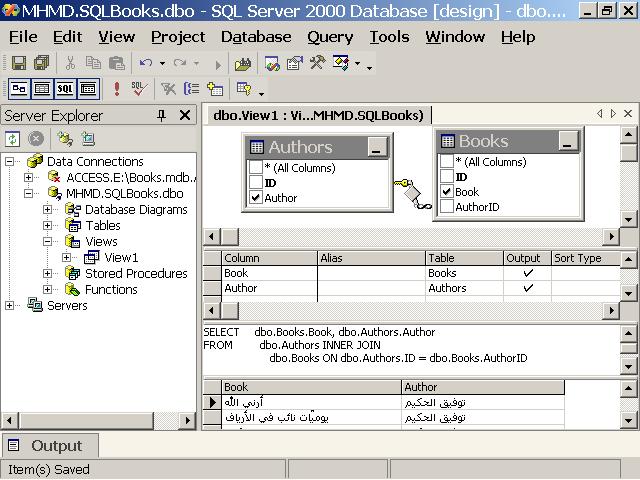
**استخدام باني الاستعلام لتنفيذ جمل SQL:**

لكيّ تجرّب الأمثلة التي سنشرحها في هذا الفصل، يمكنك استخدام "باني الاستعلام" Query Builder، حيث ستساعدك هذه الأداة على إنشاء جمل SQL بطريقة مرئيّة Visual.

اضغط العنصر Views في متصفّح خوادم الإنترنت بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر New View.

في البداية ستظهر لك نافذة "إضافة جدول"، حيث ستحتوي القائمة على جدولي المؤلّفين والكتب.. حدّد الاثنين واضغط زرّ إضافة.

بعد إغلاق النافذة سيظهر لك نافذة باني الاستعلام، التي تتكوّن من أربعة أجزاء: المخطّط Diagram، جدول الحقول Grid، ونصّ جملة SQL التي سيتمّ تنفيذها، ونتائج التنفيذ Results.



**قسم المخطّط Diagram Pane:**

هنا ستظهر الجداول التي ستدخل في بناء الاستعلام (الجداول التي سنحصل على البيانات منها)، بصورة مشابهة للتي رأيناها في مخطّطات قاعدة البيانات [Database Diagrams](file:///C:\\Users\\Administrator\\Desktop\\vb.net\\15-%20تركيب%20قواعد%20البيانات\\4-%20إنشاء%20قواعد%20بيانات\\3-%20أهمّ%20عناصر%20الرابطة\\1-%20مخطّطات%20قاعدة%20البيانات.files).. ويمكنك أن تحذف أيّ جدول بضغطه بالفأرة، وضغط زرّ Delete من لوحة المفاتيح.. ولإضافة الجدول مرّة أخرى، اضغط بزرّ الفأرة الأيمن في أيّ موضع من المخطّط، ومن القائمة الموضعيّة اضغط Add Table، حيث سيظهر لك مربّع حوار إضافة جدول.

وإذا كانت هناك علاقات قد تمّ إنشاؤها سابقا فستجد الخطوط التي توضّحها مرسومةً ما بين الجداول.. ويمكنك أن تحدّد العلاقات بين الجداول، بسحب أسماء الحقول من الجدول الرئيسيّ وإسقاطها على أسماء الحقول في الجداول التابعة، بنفس الطريقة التي تعلّمناها عند التعامل مع مخطّطات قواعد البيانات.

وفي حالتنا هذه سيظهر خطّ بين الحقل ID في جدول المؤلّفين، والحقل AuthorID في جدول الكتب.. على منتصف هذا الخطّ سيظهر رمز يمثّل نوع الربط الذي سيتمّ تنفيذه في جملة SQL ما بين الجدولين.. لهذا الربط أنواع عديدة، يمكن اختيارها بضغط الرمز بزرّ الفأرة الأيمن، والاختيار من القائمة الموضعيّة.

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

وتتمثّل أوّل خطوة في بناء جملة SQL، في اختيار الحقول التي ستظهر في النتيجة.. ولفعل ذلك اضغط بالفأرة مربّع الاختيار على يسار كلّ حقل تريد إضافته.. فإذا كنت تريد إضافة كلّ حقول الجدول مرّة واحدة، فاضغط مربّع اختيار العنصر الأوّل (All Columns).

**قسم جدول الحقول Grid Pane:**

تعرض جدولا يحتوي على أسماء الحقول التي تمّ اختيارها في المخطّط.. بعض هذه الحقول قد تستخدم فقط لقراءة البيانات منها، ولكن دونَ أن تظهر في النتيجة.. لفعل ذلك مع مثل هذه الحقول، أزل علامة (✓) من خانة العمود Output المناظرة لها.

وتمثّل خانة "الاسم المستعار" Alias الاسم الجديد الذي تريد عرض العمود به.. فمثلا، لن تعرض لمستخدم برنامجك أسماء المؤلّفين العرب تحت عمود اسمه Author.. لا ريب أنّك ستجعل لهذا العمود الاسم المستعار "المؤلّف".

حالة أخرى ستحتاج فيها لاستخدام الاسم المستعار، إذا كانت النتيجة تحتوي على عمودين أو أكثر من جداول مختلفة لها نفس الاسم.. بالطبع ستحتاج للتفريق بينها قبل عرضهما للمستخدم.

بخلاف هذا اترك هذه الخانة فارغة.

**قسم SQL:**

في هذا الجزء يظهر نصّ جملة SQL، ومع كلّ اختلاف تُحدثه في المخطّط أو جدول الحقول، يحدث اختلاف في هذا النصّ.. أعتقد أنّك بقليل من الملاحظة يمكنك تعلّم لغة SQL، وذلك بتجريب بعض التغييرات في المخطّط وجدول الحقول وملاحظة تأثيرها على جملة SQL.

ليست هذا هو كلّ دور هذا القسم، فبخلاف أنّ بإمكانك كتابة أيّ جملة SQL (أو لصقها) في هذا القسم لتجريبها، فإنّ بإمكانك كذلك أن تنسخ جملة SQL التي تمّ بناؤها بالطريقة المرئيّة لتستخدمها في أيّ موضعٍ آخر من تطبيقك.

ولكن.. إلى الآن لم يتمّ تنفيذ الاستعلام!

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

**قسم النتائج Results Pane:**

هذا هو الموضع الذي تظهر به نتائج تنفيذ الاستعلام.. ولمحو النتائج، يمكنك أن تضغط قسم النتائج بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة تختار الأمر Clear Results.

بعد أن تنتهي من تصميم عرض البيانات View، احفظه باسم View1، ثم أغلق نافذة باني الاستعلام.

الآن يظهر الاسم View1 تحت العنصر Views في متصفّح خوادم الإنترنت.. اضغطه مرّتين بالفأرة.. سيعرض لك جدولا يحتوي على نتائج تنفيذ الاستعلام.

ولو أردت إعادة تصميم هذا العرض، فاضغطه بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Design View.

**مثال:**

في قسم المخطّط في باني الاستعلام أضف الحقل Author من جدول المؤلّفين، والحقل Book من جدول الكتب.. وفي قسم جدول الحقول، غيّر خاصيّة Alias لكلّ منهما إلى "المؤلّف" و"الكتاب" على الترتيب.. نفّذ الاستعلام.. ستجد أنّ قسم النتائج يعرض جدولا يحتوي على أسماء الكتب وأسماء مؤلّفيها.

**تحذير:**

لو قمت بحذف أيّ صفّ في جدول العرض فسيتمّ حذف الخانات المناظرة في الأعمدة التي يعرضها من الجداول الأصليّة حذفا نهائيّا.. ليس هذا فحسب، فنظرًا لأنّنا أنشأنا تكاملا مرجعّيا بين الجدولين، فإنّ محاولة حذف الصفّ الذي يحتوي على اسم (توفيق الحكيم) مثلا، ستؤدّي إلى حذف اسم (توفيق الحكيم) من جدول المؤلفين، ممّا سيعمل آليّا على حذف كلّ كتبه من جدول الكتب!.. عامّة ستظهر رسالة تأكيد الحذف أولا، فلا تتسرّع في عمليّة الحذف، فقد تؤدّي لضياع كمّ ضخمٍ من البيانات على عكس رغبتك، ممّا قد يرهقك في إعادة إدخالها.. لهذا فكن على حذر، ومن الأفضل الاحتفاظ بنسخة احتياطيّة من قاعدة البيانات.

كانت هذه فكرةً سريعةً عن باني الاستعلام، حتّى تستخدمه في تجربة جمل SQL التي سنشرحها في المقاطع التالية.. كلّ 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 بناء كلّ جملة عن طريقه.

**إنشاء استعلامات Access:**

لن تستطيع استخدام العروض Views لإنشاء الاستعلامات في قواعد البيانات المنشئة في Access، فإنشاؤها من متصفح الخوادم غير متاح إلا مع قواعد بيانات SQL Server!!.. يا للحسرة!

لا تبتئس.. ما زال بإمكانك إنشاء هذه الاستعلامات، سواء من خلال كتابة الكود في VB أو من خلال Access نفسه.

وعموما: يمكن استخدام باني الاستعلام مع قواعد بيانات Access، ولكن بطرق أخرى، سنتعرّف عليها في الفصل القادم بإذن الله.

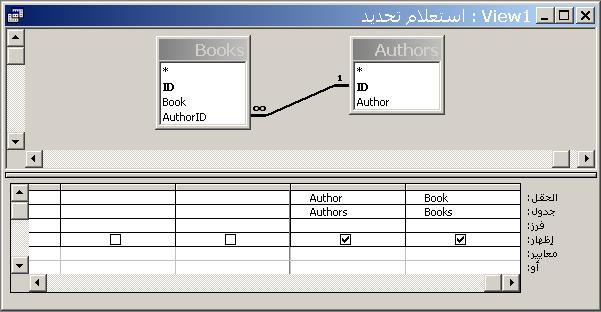
ولك، تعال نرى كيف ننشئ عرضا من Access:

لإضافة استعلام لقاعدة بيانات الكتب، يعرض أسماء الكتب ومؤلّفيها، افتح هذه القاعدة في Access، ومن الشرائط الجانبية اضغط الشريط "استعلامات" Queries.. سيختفي اسما الجدولين، وستظهر نافذة الاستعلامات.



اضغط الأمر: "إنشاء استعلام في طريقة عرض التصميم".. سيظهر لك مربّع حوار "إضافة جدول".. استخدمه لإضافة الجدولين وأغلقه.. الآن تبدو أمامك نافذة شديدة الشبه بنافذة باني الاستعلام، وإن كانت لا تحتوي على المنطقة التي تظهر بها جمل SQL ولا على المنطقة التي تعرض نتائج الاستعلام.. يمكنك استخدام هذه النافذة لبناء الاستعلام، مع ملاحظة أنّ هناك اختلافات طفيفة.. فمثلا جدول الحقول يعرض أسماء الحقول كأعمدة وخصائصها كصفوف على عكس باني الاستعلام.. كما أنّ إضافة حقل إلى جدول الحقول يتمّ عن طريق نقره مرّتين بالفأرة، حيث لا توجد مرّبعات اختيار بجوار أسماء الحقول في منطقة الجداول.

أضف حقلي الكتب والمؤلّفين، كما في الصورة التالية:



اضغط زرّ الحفظ.. سمّ الاستعلام View1، وأغلق عرض التصميم.

الآن سيظهر اسم الاستعلام في نافذة الاستعلامات.. انقره مرّتين بالفأرة.. سيظهر لك جدول النتائج.. أغلقه.. لو أردت الآن العودة إلى عرض الجداول، فاضغط شريط "جداول".

ستجد اسم هذا الاستعلام الآن تحت العنصر Views في متصفّح خوادم الإنترنت في VS.NET.

**استعلامات التحديد Selection Queries**

* [**جملة التحديد SELECT Statement:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\1-%20جملة%20لتحديد.htm)
* [**فقرة "حيثُ" WHERE Clause:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\3-%20حيث.htm)
* [**استخدام كلمةAS :**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\4-%20تغيير%20اسم%20الحقل.htm)
* [**الكلمة TOP:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\5-%20بالأكثريّة.htm)
* [**الكلمة "منفصل" DISTINCT:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\6-%20بدون%20تكرار.htm)
* [**القيم المنعدمة Null Values:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\7-%20القيم%20المنعدمة.htm)
* [**فقرة الترتيب ORDER BY:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\8-%20فقرة%20الترتيب.htm)
* [**إجراء العمليّات على خانات الحقول:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\9-%20إجراء%20العمليّات%20على%20خانات%20الحقول.htm)
* [**ربط الجداول SQL Joins:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20استعلامات%20التحديد\12-%20ربط%20الجداول.htm)

**جملة التحديد SELECT Statement:**

تمكّنك جملة SELECT من استرجاع جزء من الجدول.. وأبسط صور هذه الجملة كالتالي:

**SELECT أسماء الأعمدة**

**FROM أسماء الجداول**

**ملاحظات:**

1- أسماء الأعمدة والجداول يفصل بينها علامة ",".

2- يمكن كتابة اسم مختصر للجدول بعد اسمه الحقيقيّ مباشرةً، واستخدامه في باقي الجملة للتسهيل.

3- إذا كان اسم الجدول يحتوي على أيّ رموز غير مقبولة برمجيّا ـ مثل المسافات أو العلامات ":" أو "+" أو "=" ... إلخ ـ فيجب وضعه بين قوسين مضلّعين [ ].

4- إذا تشابهت بعض أسماء الحقول في أكثر من جدول، فيجب التمييز بينها بكتابة اسم كلّ حقل على الصيغة: (اسم الجدول.اسم الحقل).. وإذا كان اسم الحقل يحتوي على رموز غير مقبولة، فيجب استخدام الصيغة: (اسم الجدول.[اسم الحقل]).

5- ستظهر الحقول في السجلات المعادة بالترتيب الذي كتبت أسماءها به في جملة SELECT.

6- جمل SQL غير حسّاسة لحالة الأحرف، ولكن تُعُورِفَ على كتابة الكلمات الرئيسيّة فيها بحروف كبيرة لسهولة تمييزها.

7- يمكن كتابة جملة SQL على أكثر من سطر، فهي ليست كأوامر VB تنتهي بنهاية السطر، ولكنّها كأوامر C++ تنتهي بالفاصلة المنقوطة ";"، وإن كان من الممكن عدم كتابتها كاختصار، لأنّك في الغالب لن تكتب إلا جملة واحدة.

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

فمثلا: لتحديد كلّ سجلات جدول المؤلّفين يمكنك استخدام جملة كالتالية:

**SELECT ID, Author**

**FROM Authors**

ولكن ماذا لو كان الجدول يحتوي على عشرين عمودا، نريد استرجاعها كلّها.. هل سنكتب أسماء عشرين عمودا؟؟!!

طبعا لا.. بل إنّ الأمر سيكونُ أسهل من كتابة اسم عمود واحد، حيث سنستخدم العلامة "\*" (أو كلمة ALL) كالتالي:

**SELECT \***

**FROM Authors**

ولو أردت استرجاع كلّ حقول جدول ما، مع حقل أو اثنين من جدول آخر، فاستخدم الصيغة التالية:

**SELECT Authors.\*, Books.Book**

**FROM Authors, Books**

ويمكن اختصار أسماء الجداول في المثال السابق كالتالي:

**SELECT A.\*, B.Book**

**FROM Authors A, Books B**

لو جرّبت هذا المثال فستفاجأ بنتائج غريبة، حيث سيتم تكرار كلّ سجلّ من سجلات المؤلّفين مع كلّ أسماء الكتب، ممّا يبدو معه أنّ كلّ مؤلّف قد ألّف كلّ الكتب!.. إنّ هذا خطأ متوقّع، لأنّنا لم نضع أيّ شرط يربط بين حقول الجدولين، وهو ما سنتعلّمه حالا.

**استخلاص السجلات التي تحقّق شروطا معيّنة**

* [**فقرة "حيثُ" WHERE Clause:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20حيث\1-%20حيث.htm)
* [**معاملات المقارنة:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20حيث\2-%20معاملات%20المقارنة.htm)
* [**المقارنة بحقل من جدول:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\3-%20حيث\3-%20المقارنة%20بحقل%20من%20جدول.htm)

**فقرة "حيثُ" WHERE Clause:**

لم تخترع قواعد البيانات لكي تقرأ كلّ سجلات الجدول كما أدخلتها!!.. إنّ الأهمّ من ذلك قدرتك على حساب بعض النتائج واستخلاص المعلومات من هذه السجلات.. هنا يبرز الدور الحيويّ لفقرة WHERE، فهي تسمح لك بتحديد الشروط التي سيتمّ على أساسها استرجاع السجلات، بحيث لا تسترجع منها إلا ما يحقّق هذه الشروط.

ولن تكون صيغة هذه الفقرة صعبةً عليك، فستكب صيغة الشرط مماثلةً للشروط التي تعوّدت كتابتها في جملة If في VB.

تعال نسترجع كلّ الكتب التي يحمل مؤلّفها رقم 1 أو 3:

**SELECT Book**

**FROM Books**

**WHERE AuthorID = 1 OR AuthorID = 3**

ولكنّ من هو المؤلّف رقم 1؟

إنّ هذه الطريقة قد تكون عديمة الجدوى، فهي غير عمليّة في البرمجة.

إنّك تريد أن تعرف الكتب التي ألّفها (توفيق الحكيم) مباشرةً، دونَ أن تبحث عن رقمه.. بسيطة.. استخدم الجملة التالية:

**SELECT Book**

**FROM Books, Authors**

**WHERE Author = 'توفيق الحكيم' AND AuthorID = Authors.ID**

لاحظّ أنّنا كتبنا اسم جدول المؤلّفين في فقرة FROM رغم أنّ أيا من حقوله لن يظهر في النتيجة، وذلك لأنّنا سنستخدم حقول هذا الجدول في فقرة WHERE.

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

وكان من الممكن تبديل ترتيب جملتي الشرط، فالترتيب ليس مهمّا، حيث يقوم تطبيق قاعدة البيانات بتنفيذ الشروط على حسب أولويّة تنفيذها، وليس على حسب ترتيب كتابتها.

والآن تعال نطوّر المثال السابق، لنحصل على كلّ الكتب التي ألّفها (توفيق الحكيم) أو (عباس العقاد).. في هذه الحالة لا بدّ من عرض اسم المؤلّف في النتيجة، وإلا لتعذّر معرفة مؤّلف كلّ كتاب:

**SELECT Book, Author**

**FROM Books, Authors**

**WHERE (Author = 'توفيق الحكيم' OR Author = 'عباس العقاد')**

**AND AuthorID = Authors.ID**

لاحظ استخدام كلمة OR بين اسمي المؤلّفين، ولو استخدمت كلمة AND لما حصلت على أيّ نتائج، فلا يمكن أن يوجد جدول به سجلّ يحتوي على خانة واحدة لها قيمتان!!.. ولكن يمكن أن يكون للخانة القيمة كذا **أو** القيمة كذا.. يمكن كذلك أن تكون لإحدى خانات السجلّ القيمة كذا **و** لخانة أخرى القيمة كذا.. فمثلا، لو كان لدينا في جدول الكتب كتاب اسمه "حياتي"، وحدث أن تكرّر اسمه في أكثر من سجلّ، نتيجة لأنّ هناك أكثر من مؤلّف له كتاب بهذا الاسم، فإن بإمكانك استرجاع الكتاب الذي يحقّق الشرط التالي:

**SELECT ID**

**FROM Books, Authors**

**WHERE Author = 'عقل'**

**AND Book = 'حياتي'**

**AND AuthorID = Authors.ID**

تلاحظ طبعا أنّ الهدف من الاستعلام قد اختلف، فعندما تبحث عن كتاب تعرف اسمه واسم مؤلّفه، فلا بدّ أنّك لسبب أو لآخر تريد أن تعرف رقم الكتاب ID.

ولكن ماذا لو أردنا أن نعرض كلّ أسماء الكتب مع ما يناظرها من مؤلّفين؟

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

**SELECT Book, Author**

**FROM Books, Authors**

**WHERE AuthorID = Authors.ID**

**ملحوظة1:**

يمكنك عرض الجدول فارغا بوضع شرط خاطئ أو مستحيل في فقرة WHERE (مثل 1 = 2)!!، كالتالي:

**SELECT \***

**FROM Books**

**WHERE 1 = 2**

يفيد هذا عندما تريد أن تسمح للمستخدم إدخال سجلّ جديد، دون العبث بالسجلات القديمة.

**ملحوظة2:**

لكي تنشئ فقرة WHERE بباني الاستعلام، أضف للاستعلام الحقل الذي سيظهر في هذه الفقرة (بضغط خانة الاختيار المجاورة لاسمه في المخطّط).. سيضاف هذا الحقل إلى جدول الحقول Grid Pane.. اضغط هذا الحقل ـ في جدول الحقول ـ بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Group By.. سيظهر عمود جديد يحمل اسم Group By.. اضغط الخانة المناظرة للحقل في هذا العمود.. سيظهر لك زرّ إسدال القائمة.. اضغطه بالفأرة، ومن القائمة الموضعيّة اختر WHERE.. وفي العمود التالي مباشرة ـ ذلك الذي يحمل اسم Criteria ـ اكتب الشرط الذي تريده، بدون كتابة اسم الحقل (كأن تكتب مثلا > 5).. فإذا أردت أن تكتب شرطا آخر مرتبطا بالشرط الأوّل بالمعامل OR، فانتقل إلى العمود التالي.. ستجده يحمل عنوان OR.. اكتب الشرط الذي تريده في هذا العمود.. فإذا كان هناك المزيد من الشروط، فلديك المزيد من الأعمدة التي تحمل العنوان OR.

ولكن ماذا لو أردت أن تكتب شرطا يرتبط بالشروط السابقة بالمعامل AND؟

في هذه الحالة يجب أن يظهر هذا الشرط في صفّ جديد.. طبعا لن تجد مشكلةً إذا كان الشرط الجديد ينطبق على حقل مختلف.. ولكنّ المشكلة هي أنّ يظهر نفس الحقل في أكثر من شرط يربطها المعامل AND!

بسيطة.. اضغط بالفأرة في أيّ خانة فارغة في العمود المسمّى Column، ومن القائمة المنسدلة اختر نفس اسم الحقل الذي تريد تكراره.. هذه الطريقة تمكّنك من تكرار اسم الحقل أكثر من مرّة في جدول الحقول.. في الصفّ الجديد اختر كلمة WHERE في العمود Group By، واكتب الشرط في خانة Criteria.

ولكن ماذا لو كان هناك شرطان كلّ منهما على حقل مختلف، ويربطهما المعامل OR؟

في هذه الحالة يجب أن تكتب الشرط الثاني في خانة OR، مع ترك خانة Criteria للحق الثاني فارغة.

وكتلخيص: الشروط التي تظهر في خانة Criteria في صفوف مختلفة يربطها المعامل AND، بينما الشروط التي تظهر في خانة OR، حتّى لو كانت في صفوف مختلفة، يربطها المعامل OR.

**معاملات المقارنة:**

رأينا كيف نستخدم المعاملات المنطقيّة AND و OR و NOT في ربط شروط فقرة WHERE.. ليس هذا هو كلّ ما لدينا، فكثيرا ما يكون استخدام المعاملين AND و OR في تكوين جملة الشرط مرهقا، خاصّة عندما يزداد عدد الشروط.. لهذا فإنّ لدينا المعاملات التالية:

* [**المعاملات >, <, =, < >:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20معاملات%20المقارنة\1-%20المعاملات.htm)
* [**بين BETWEEN:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20معاملات%20المقارنة\2-%20بين.htm)
* [**في IN:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20معاملات%20المقارنة\3-%20في.htm)
* [**المعامل "يشبه" LIKE Operator:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\2-%20معاملات%20المقارنة\4-%20يشبه.htm)

**المعاملات >, <, =, < >:**

كأيّ جملة شرط عاديّة، يمكنك استخدام معاملات المقارنة التقليديّة >, <, =, < >.. فمثلا للحصول على كلّ الكتب ما عدا ذلك الذي يحمل رقم 9، استخدم الجملة التالية:

**SELECT ID, Book**

**FROM Books**

**WHERE ID <> 9**

وللحصول على الكتب التي تسبق كتاب "حائرة في الحب" في الترتيب الهجائيّ، استخدم الجملة التالية:

**SELECT ID, Book**

**FROM Books**

**WHERE Book < 'حائرة في الحب'**

وللحصول على الكتب التي تحمل أرقاما أكبر من 12 بالإضافة إلى الكتاب رقم 2 استخدم الجملة التالية:

**SELECT ID, Book**

**FROM Books**

**WHERE ID = 2 OR ID > 12**

**بين BETWEEN:**

يسمح لك هذا المعامل بتحديد المجال الذي ينتمي إليه الحقل.. فمثلا، يمكننا استخدام الجملة التالية للحصول على الكتب التي ينحصر رقمها بين 3 و 10:

**SELECT ID, Book**

**FROM Books**

**WHERE ID BETWEEN 3 AND 10**

كما يمكننا استخدام NOT قبل هذا المعامل للحصول على قيم الحقل التي لا تنتمي للمجال المحدّد.. والجملة التالية تعيد الكتب التي لا تنحصر بين 3 و 10:

**SELECT ID, Book**

**FROM Books**

**WHERE ID NOT BETWEEN 3 AND 10**

**في IN:**

أحيانا لا يحلّ المعامل BETWEEN كلّ مشاكلنا، فماذا لو أردنا أن نختار قيما متفرّقة للحقل؟.. في هذه الحالة ستكون جملة AND طويلة وبطيئة التنفيذ.

في هذه الحالة يمكن استخدام المعامل IN ووضع كلّ القيم بين قوسين مفصولةً بعلامة , .. والجملة التالية تعيد لك الكتب التي تحمل الأرقام 2 و 6 و 7 و 10:

**SELECT ID, Book**

**FROM Books**

**WHERE ID IN (2,6,7,10)**

ويمكن نفي الجملة السابقة للحصول على باقي الكتب (التي لا تحمل الأرقام المذكورة):

**SELECT ID, Book**

**FROM Books**

**WHERE ID NOT IN (2,6,7,10)**

ولا يقتصر الأمر على الأرقام، فالجملة التالية تعيد كلّ الكتب التي ألّفها (توفيق الحكيم) و(نبيل فاروق) و(محمد حمدي غانم):

**SELECT Book, Author**

**FROM Books, Authors**

**WHERE AuthorID = Authors.ID**

**AND Author IN ('توفيق الحكيم', 'نبيل فاروق' , 'محمد حمدي غانم')**

ولا مانع من تكوين جملة شرط معقّدة تحتوي على مزيج من كلّ المعاملات التي شرحناها معا.. المهمّ أن تصل لهدفك من أقصر طريق.

**المعامل "يشبه" LIKE Operator:**

استخدم هذا المعامل في جملة WHERE بنفس الطريقة التي استخدمته بها في مقارنة النصوص في VB ([راجع الفصل الثالث](file:///C:\\Users\\01-%20أساسيّات%20اللغة\\03-%20المتغيرات\\3-%20أنواع%20المتغيرات\\5-%20المتغيّرات%20النّصّيّة\\3-%20مقارنة%20النصوص.htm))، ولكن مع بعض الاختلافات الطفيفة، ستدركها عند التعرّف على العلامات الخاصّة التي يستخدمها هذا المعامل، وهي:

|  |  |  |
| --- | --- | --- |
| علامة  SQL | ما يناظرهـا في VB | 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 المؤلّفين التي تحتوي على حرف الميم في أيّ موضع:  **SELECT Author**  **FROM Authors**  **WHERE Author LIKE '%م%'**  ولو أردت أن تبحث عن السجلات التي تبدأ بحرف الميم، فاستخدم الصيغة 'م%'. |
| \_  (الشرطة المنخفضة Underscore) | ? | تعبّر عن حرف واحد فقط ـ مهما كان هذ1 الحرف ـ ويجب أن يكون موجودا في النصّ في الموضع المناظر لهذا العلامة في الصيغة،.  مثال: استخدم الجملة التالية للحصول على جميع أسماء المؤلّفين 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 أنّ 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 يليه أية حروف. |
| # | # | تعبّر عن رقم واحد فقط من 0 إلى 9. |
| [ ] | [ ] | تعبّر عن حرفٍ واحد 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 المؤلّفين 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 السجلات التي تبدأ بأحد الحروف المحصورة بين الفاء والياء وتنتهي بحرف السين، فاستخدم الصيغة:  **'[ف-ي]%س'** |
| [^ ] | [! ] | 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. |

**ملحوظة:**

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

**المقارنة بحقل من جدول:**

من الإمكانيّات التي تمنحها لك فقرة 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 كلّ الكتب التي ألّفها (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 المؤلّفين، لتقوم جملة SELECT الخارجيّة بعرض الحقول التي تحتوي على أيّ من هذين الرقمين في حقل AuthorID.

**استخدام كلمةAS :**

يمكنك استخدام كلمة AS لوضع الاسم المستعار للحقول التي تريدها، وذلك كالتالي:

**SELECT Book AS [اسم الكتاب], Author AS [مؤلّفه]**

**FROM Books, Authors**

**WHERE AuthorID = Authors.ID**

الآن ستظهر نفس النتائج السابقة، ولكنّ العمودين سيحملان اسمين عربيّين هذه المرّة.

ومن الاستخدامات الطريفة للاسم المستعار، استخدامه لتسميّة أحد الحقول الناتجة عن دمج حقلين معا.. ما موضوع هذا الدمج؟

افترض أنّنا نريد عرض حقل يحتوي على اسم الكتاب متبوعا بشرطة متبوعةً باسم المؤلّف.. كلّ ما علينا فعله إذ ذاك، هو توصيل concatenate الحقلين، تماما كما نفعل مع المتغيّرات النصّيّة في VB، مع منح الحقل الناتج اسما مستعارا باستخدام كلمة 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 جدول الحقول Grid 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 النصّيّة 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 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 ومؤلّفيها 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 كالتالي:

**SELECT TOP 5 PERCENT Book, Author**

**FROM Books, Authors**

**WHERE AuthorID = Authors.ID**

**الكلمة "منفصل" DISTINCT:**

استخدم هذه الكلمة لإزالة السجلات المكرّرة.. فمثلا لو عرضنا جدولا بأسماء الكتب بدون مؤلّفيها، فإنّ كتاب "حياتي" سيتكرّر 3 مرّات.. هنا يمكن تلافي هذا التكرار باستخدام الجملة التالية:

**SELECT DISTINCT Book**

**FROM Books**

**القيم المنعدمة Null Values:**

هناك اختلاف جوهريٌّ بينَ أن يكونَ النص فارغا "" أو أن تكون العدديّة مساوي للصفر، وبينَ أن تكون لهما القيمة Null، حيثُ إنّها تشيرُ في هذه الحالة إلى أنّ هذه الخانة لم توضع بها أيّ قيمة من أساسِه.. إنّ مثل هذه الحالة تؤدّي إلى تبعاتٍ غير مرغوبٍ فيها، ففقرة WHERE مثلا لا تسترجع أيّ خانة فارغة، لأنّ القيمة NULL تنتج False بطريقة دائمة في أيّ عمليّة مقارنة.. هذا بالإضافة إلى أنّ دخول هذه القيمة في أيّ عمليّة حسابيّة يؤدّى لحدوث خطإ في البرنامج.

إذا ما الحل؟

الحلّ هو أن تعامل هذه القيمة بحذر، حيث يمكنك أن تختبر وجودها باستخدام التعبير IS NULL أو IS NOT NULL.. وكمثال، استخدم الاستعلام التالي للبحث عن أيّ خانة في حقل المؤلّفين تمّ تركها فارغة:

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

عامّةً يمكن تلافي الكثير من احتمالات الخطأ بوضع القيمة False في خاصيّة "السماح بطول صفريّ" Allow Nulls عند تصميم الحقول.

**فقرة الترتيب ORDER BY:**

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

**SELECT Book, Author**

**FROM Books, Authors**

**ORDER BY Book**

**WHERE AuthorID = Authors.ID**

ولكنّ هناك ثلاث كتب تحمل اسم "حياتي" فكيف سيتمّ ترتيبها؟

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

**SELECT Book, Author**

**FROM Books, Authors**

**ORDER BY Book, Author**

**WHERE AuthorID = Authors.ID**

ولكنّ هذا الترتيب تنازليّ في الوضع الافتراضيّ.. ماذا لو أردت أن يتمّ الترتيب تصاعديّا؟

في هذه الحالة أضف الكلمة DESC بعد اسم أيّ حقلٍ تريد ترتيبه تنازليّا، والكلمة ASC بعد ذلك الذي تريد ترتيبه تصاعديّا (وهي افتراضيّة ويمكن عدم كتابتها)، كالتالي:

**SELECT Book, Author**

**FROM Books, Authors**

**ORDER BY Book DESC, Author ASC**

**WHERE AuthorID = Authors.ID**

**ملحوظة:**

لتحديد كيفيّة الترتيب في باني الاستعلام، اضغط الحقل الذي تريد أن يتمّ الترتيب على أساسه 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 أنّ أيقونةً تمثّل نوع الترتيب قد ظهرت بجوار اسم الحقل.. ولو أردت إزالة الترتيب، فاضغط نفس الأمر من القائمة الموضعيّة مرّةً أخرى.. كما أنّ بإمكانك فعل ذلك مع أكثر من حقل ومن أيّ جدول، حيث ستظهر في جملة SQL بالترتيب الذي أضفتها به.

كما يمكن اختيار نوع الترتيب بطريقة أخرى، وذلك باستخدام جدول الحقول Grid Pane، حيث يمكنك تغيير قيمة هذه الخاصيّة لأيّ حقل من العمود الذي يحمل العنوان Sort Order.

**إجراء العمليّات على خانات الحقول**

* [**دوال التجميع Aggregate Functions:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\9-%20إجراء%20العمليّات.htm\1-%20دوال%20التجميع.htm)
* [**تجميع السجلات باستخدام تعبير GROUP BY:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\9-%20إجراء%20العمليّات.htm\2-%20تجميع%20السجلات.htm)
* [**استخدام فقرة HAVING:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\9-%20إجراء%20العمليّات.htm\3-%20استخدام%20فقرة.htm)

**دوال التجميع Aggregate functions:**

تمنحك SQL بعض الدوال الجاهزة لحساب وتلخيص بعض النتائج.. ويجب أن تلاحظ هنا ما يلي:

- أنّ هذه الدوال تقبل معاملا واحدا فقط، هو أحد أعمدة الجدول، أو أيّ عمود جديد ناتج عن إجراء إحدى العمليّات الحسابيّة (كالجمع والطرح والضرب والقسمة) على واحد أو أكثر من الأعمدة، أو أيّ عمود ناتج من جملة SELECT فرعيّة.

- أنّ هذه الدوال تعيد قيمة واحدة فقط (رقم).. أيّ أنّ الناتج منها هو عمود يحتوي على خانة واحدة فقط.. ولو كنت ستعرض هذه النتيجة، فاستخدم تعبير AS لمنح هذا العمود الجديد اسما مناسبا، وإلا فإنّ SQL ستمنحه اسما افتراضيّا من لديها.

- لا مانع من استخدام فقرة WHERE لتحديد السجلات التي سيتمّ عليها العمليّة الحسابيّة.

وهذه الدوال هي:

|  |  |
| --- | --- |
| COUNT() | لحساب عدد الخانات في العمود المرسل كمعامل.. ويمكن تطبيقها على أيّ نوع من البيانات.  مثال: الجملة التالية تحسب عدد الكتب التي ألّفها توفيق الحكيم:  SELECT COUNT(Book) AS [عدد الكتب المتاحة]  FROM Books, Authors  WHERE Author = 'توفيق الحكيم'  AND AuthorID = Authors.ID  لاحظّ أن الحقول التي تحتوي على القيمة NULL لا يتمّ عدّها ضمنَ السجلات.. ولو أردت أن تفعل العكس، فعليك أن ترسل الرمز (\*) كمعامل لهذه الدالة، حتّى تأخذ هذه السجلات في اعتبارها:  SELECT COUNT(\*) AS [عدد المؤلّفين]  From Authors |
| SUM() | لحساب مجموع الخانات في العمود المرسل كمعامل.. ويمكن تطبيقها على الأعمدة الرقميّة فقط.  مثال: لسبب غير واضح، قرر شخصٌ ما مهووسٌ بالأرقام، أن يجمع عمود رقم الكتاب على عمود رقم المؤلّف في جدول الكتب، وأن يحسب المجموع الكلّي للعمود الجديد.. هذا هو ما فعله بالضبط:  **SELECT SUM(ID + AuthorID) AS [مجموع كلّ أرقام الجدول]**  **From Books** |
| AVG() | لحساب المتوسّط الحسابيّ ( المجموع ÷ العدد) لخانات العمود المرسل كمعامل.. ويمكن تطبيقها على الأعمدة الرقميّة فقط.  وقد قام نفس الشخص المهووس بحساب متوسّط الأرقام، بوضع الدالة AVG بدلا من SUM في المثال السابق! |
| MIN() | لحساب أصغر قيمة في العمود المرسل كمعامل.. ويمكن تطبيقها على الأرقام والنصوص، حيث في حالة النصوص تعيد أصغر نصّ في الترتيب الأبجديّ. |
| MAX() | لحساب أكبر قيمة في العمود المرسل كمعامل.. ويمكن تطبيقها على الأرقام والنصوص، حيث في حالة النصوص تعيد أكبر نصّ في الترتيب الأبجديّ. |

الجدير بالذكر أنّ بإمكانك استخدام القيم المعادة من هذه الدوال في شروط فقرة WHERE.. والمثال التالي يريك كيف نحسب عدد الكتب التي ألّفها أوّل مؤلّف في الترتيب الأبجديّ:

**SELECT COUNT(Book) AS [عدد كتب المؤلّف الأول في الترتيب الهجائيّ]**

**FROM Books, Authors**

**WHERE AuthorID = Authors.ID**

**AND Author = (SELECT MIN(Author) From Authors)**

حيث استخدمنا جملة SELECT فرعيّة لتعيد ناتج دالة القيمة الصغرى في شرط فقرة WHERE.

عامّة نفس هذه الجملة لها صيغة ثانيةٌ، 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 كلمة DISTINCT للحصول على حقل ليس به أيّ تكرار.. ولكن أين سيكون موضع هذه الكلمة؟

ستكون بالطبع قبل اسم الحقل مباشرةً، أي داخل قوس الدالة.. والمثال التالي يريك كيف نحسب عدد المؤلّفين بحساب عدد خانات الحقل AuthorID في جدول الكتب بدون تكرار:

**SELECT COUNT (DISTINCT AuthorID) AS [عدد المؤلّفين]**

**From Books**

**ملحوظة:**

لاستخدام دوال التجميع في باني الاستعلام، أضف للاستعلام الحقل الذي تريد تطبيق الدالة عليه (بضغط مربّع الاختيار المجاور له في الجدول).. سيظهر اسم هذا الحقل في جدول الحقول.. اضغطه بزرّ الفأرة الأيمن، ومن القائمة الموضعيّة اضغط الأمر Group By.. سيظهر عمود جديد في جدول الحقول اسمه Group By.. اضغط الخانة المناظرة للحقل في هذا العمود بالفأرة.. سيظهر لك زرّ إسدال القائمة.. اضغطه بالفأرة.. ستظهر لك قائمة بها أسماء الدوال التي يمكنك استخدامها.. اختر منها ما تريد.

**تجميع السجلات باستخدام تعبير GROUP BY:**

رأينا كيف نجري بعض العمليّات الحسابيّة باستخدام دوال التجميع.. ولكن ماذا لو أردنا مثلا أن نحصل على سجلّ يحتوي على عدد الكتب التي ألّفها كلّ مؤلّف؟

في هذه الحالة لن تسعفنا التعبيرات التي تعلّمناها حتّى الآن.

هنا ستبرز أهمّية التعبير GROUP BY، حيث يؤدّي إلى ضمّ كلّ الصفوف ذات القيم المتشابهة في الحقول المحدّدة معا، بحيث تعمل دوال التجميع على كلّ منها كأنّها جدول مستقلّ، وبهذا يكون ناتج دوال التجميع في هذه الحالة عمودا يحتوي على مجموعة من الصفوف، وليس صفّا واحدا كما ألفنا من قبل.

هذه هي الجملة التي تحسب عدد كتب كلّ مؤلّف:

**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 جملة SQL.

ولكن.. الجدول الناتج من الاستعلام السابق يحتوي على أرقام المؤلّفين وليس أسماءهم.. فما العمل؟

إنّ هناك قاعدةً تقضي بأنّه عند استخدام فقرة GROUB BY يجب أن تكون كلّ الحقول التي تظهر في فقرة SELECT مستخدمة إمّا في دالة التجميع أو في فقرة GROUB BY.. إنّ هذا يعني أنّنا لا نستطيع استخدام الحقل Author مباشرةً.. إذن ما العمل؟

بسيطة.. أليس ناتج الاستعلام السابق جدولا؟.. تعال نعامله كما عاملنا جدول الكتب من قبل، حيث سنستخدم جملة WHERE لربط رقم المؤلّف باسمه بين هذا الجدول المشتقّ، وبين جدول المؤلّفين.

ولكن كيف سنتعامل مع هذا الجدول الوهميّ الجديد؟

ربما ستستغرب ذلك، ولكنّه غاية في البساطة.. ضع كلّ جملة الاستعلام السابقة التي تُنتج هذا الجدول بين قوسين، وضعها في جملة FROM كأنّها اسم جدول!

شيءٌ واحد فقط سنحتاج لتعديله قبل أن يعمل هذا الأمر، هو حتميّة كتابة فقرة TOP في تعبير هذا الجدول المشتقّ.. وكحيلة بسيطة، دعنا نعرض 100% من الجدول.. انظر للجملة التالية:

**SELECT Author, [عدد الكتب]**

**FROM Authors, (**

**SELECT TOP 100 PERCENT**

**AuthorID, COUNT(AuthorID) AS [عدد الكتب]**

**FROM Books**

**GROUP BY AuthorID**

**ORDER BY [عدد الكتب]**

**)**

**WHERE AuthorID = Authors.ID**

إنّ هذه الإمكانيّة الرائعة تسمح لك بكتابة جمل SELECT متداخلة، لأداء استعلامات غاية في التعقيد.

عامّةً هناك طريقة أخرى لحلّ المشكلة السابقة، تعتبر أبسط وأسهل، وذلك بوضع الحقل Author في فقرة GROUP BY ولكن بعد إضافة فقرة WHERE، كالتالي:

**SELECT Author, COUNT(AuthorID) AS [عدد الكتب]**

**FROM Authors, Books**

**WHERE AuthorID = Authors.ID**

**GROUP BY Author**

**ORDER BY [عدد الكتب]**

فكأنّنا كوّنا جدول الكتب وأسماء مؤلّفيها أولا، ثمّ استخدمنا فقرة GROUP BY على الحقل Author لحساب عدد مرات تكراره لكلّ مؤلّف.

ولا غبار عليك لو استخدمت أكثر من دالة تجميع مع فقرة GROUP BY.

**ملحوظة:**

عرفنا كيف نستخدم دوال التجميع في باني الاستعلام.. ورغم أنّنا اخترنا الدالة من العمود الذي يحمل العنوان Group By، إلا إنّ فقرة Group By لم تظهر في جملة SQL.. كيف إذن نضيف هذه الفقرة؟

يمكن أن تفعل ذلك بنفس طريقة إضافة دوال التجميع، ولكن مع ترك قيمة الخانة كما هي Group By.

ولكن ماذا لو أردنا أن نضيف دالة تجميع وفقرة Group By لنفس الحقل؟

بسيطة.. بمجرّد اختيار دالة التجميع للحقل، ستجد أنّ علامة الاختيار المجاورة لاسمه في جدوله الأصليّ قد اختفت.. ضعها مرّة أخرى.. ستجد أنّ اسم الحقل قد ظهر مرّة أخرى (مكرّرا) في جدول الحقول، وستجد أنّ قيمة العمود Group By هي Group By.. اتركها كما هي، وبهذا تحلّ المشكلة!.. ويمكنك أن تكرّر اسم نفس الحقل في جدول الحقول بطريقة أخرى.. اضغط خانة فارغة في عمود الحقول.. سيظهر لك زرّ إسدال القائمة.. اضغطه بالفأرة، ومن القائمة اختر اسم الحقل.

**استخدام فقرة HAVING:**

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

في هذه الحالة يمكننا استخدام فقرة HAVING التي تسمح باستخدام دوال التجميع، كالتالي:

**SELECT Author, COUNT(AuthorID) AS [عدد الكتب]**

**FROM Authors, Books**

**WHERE AuthorID = Authors.ID**

**GROUP BY Author**

**HAVING COUNT(AuthorID) > 1**

**ORDER BY [عدد الكتب]**

**الربط الأيسر Left Join:**

هذه العمليّة تعرض كلّ سجلات الجدول الأيسر (الموجود في بداية الصيغة)، مع بعض سجلات الجدول الأيمن، التي تحقّق شرطا معيّنا.

وكمثال: يمكننا أن نعرض كلّ الكتب من جدول الكتب، مع أسماء المؤلّفين المناظرة لهذه الكتب، كالتالي:

**SELECT Book, Author**

**FROM Authors LEFT JOIN Books**

**ON AuthorID = Authors.ID**

**ملحوظة1:**

لكي تشعر بوجود اختلاف عن ناتج جملة WHERE، يجب أن يكون هناك بعض المؤلّفين الذين لا توجد لهم كتب مناظرة في جدول الكتب، حيث سيتمّ عرض أسمائهم أيضا في هذه الحالة، مع ترك خانة الكتب فارغة NULL.. هذا بالإضافة لعرض باقي الكتب وما يناظرها من مؤلّفين (لكن دون عرض الكتب التي لا يناظرها مؤلفون).

**ملحوظة2:**

لإنشاء الجملة السابقة باستخدام باني الاستعلام، حدّد الحقل Book من جدول الكتب، والحقل Author من جدول المؤلّفين، ثمّ اضغط بزرّ الفأرة الأيمن على علامة الربط في منتصف الخطّ الواصل بينَ الجدولين، ومن القائمة الموضعيّة اضغط Property Pages.. ومن مربّع الحوار، اختر الاختيار All Rows From Authors، ولا تختر الاختيار All Rows From Books.. أغلق النافذة.. انتهى!

**الربط الأيمن Right Joins:**

هذا النوع مماثل للنوع السابق، إلا إنّ كلّ سجلات الجدول الأيمن سيتمّ عرضها بالكامل، مع عرض سجلات الجدول الأيسر التي تحقّق شرطا معيّنا.

**SELECT Book, Author**

**FROM Authors RIGHT JOIN Books**

**ON AuthorID = Authors.ID**

**ملحوظة1:**

لكي تشعر بوجود اختلاف عن ناتج جملة WHERE، يجب أن يكون هناك بعض الكتب التي لا يوجد لها مؤلّف مناظر في جدول المؤلّفين، حيث سيتمّ عرض أسمائها في هذه الحالة، مع ترك خانة المؤلّف فارغة NULL.. هذا بالإضافة لعرض باقي الكتب وما يناظرها من مؤلّفين.

حريٌّ بي أن أذكّرك، أنّنا عندما أنشأنا الحقل AuthorID أزلنا علامة الاختيار من خاصيّة Allow Nulls ممّا لن يسمح لك بترك خانة رقم المؤلّف فارغة.. عامّة يمكنك أن تغيّر قيمة هذه الخاصيّة وتجرّب إدخال كتاب بدون مؤلّف، لترى تأثير عمليّة الربط الأيمن.

**ملحوظة2:**

لإنشاء الجملة السابقة باستخدام باني الاستعلام، اتبع نفس الخطوات التي اتبعناها في إنشاء الربط الأيسر، ولكن حدّد الاختيار All Rows From Books، وأزل الاختيار All Rows From Authors.

**الربط الكامل Full Joins:**

هذا النوع هو مزيج من النوعين السابقين، وفيه يتمّ عرض كلّ بيانات الجدولين:

**SELECT Book, Author**

**FROM Authors FULL JOIN Books**

**ON AuthorID = Authors.ID**

**ملحوظة1:**

لكي تشعر بوجود اختلاف عن ناتج جملة WHERE وعن ناتج العمليّتين السابقتين، يجب أن يكون هناك بعض المؤلّفين الذين لا توجد لهم كتب، وبعض الكتب التي لا يوجد لها مؤلّف، حيث سيتمّ عرض أسماء هؤلاء المؤلّفين وهذه الكتب، مع ترك الخانة المناظرة فارغة NULL.. هذا بالإضافة لعرض باقي الكتب وما يناظرها من مؤلّفين.

**ملحوظة2:**

لإنشاء الجملة السابقة باستخدام باني الاستعلام، اتبع نفس الخطوات التي اتبعناها في إنشاء الربط الأيسر، ولكن اختر كلا الاختيارين All Rows From Books، وAll Rows From Authors.

**الربط الداخليّ Inner Joins:**

هذه العمليّة مماثلة لفقرة WHERE، حيث يتمّ عرض السجلات المتوافقة فقط من الجدولين.

**SELECT Book, Author**

**FROM Authors INNER JOIN Books**

**ON AuthorID = Authors.ID**

**ملحوظة:**

لإنشاء الجملة السابقة باستخدام باني الاستعلام، اتبع نفس الخطوات التي اتبعناها في إنشاء الربط الأيسر، ولكن أزل كلا الاختيارين All Rows From Books، وAll Rows From Authors.

**استعلامات الأداء Action Queries**

يعتبر هذا النوع من الاستعلامات أبسط من استعلامات التحديد، حيث لا يقوم باسترجاع أيّ سجلات.. ولكنّه في المقابل يقوم بالتغيير في محتويات الجدول، سواء بتحديث قيم السجلات أو إضافة سجلات جديدة أو حذف بعض السجلات الموجودة.

هذا وتعيد هذه الاستعلامات عدد السجلات التي تأثّرت (وليس السجلات نفسها).

**ملحوظة:**

لن يمكنك تجريب هذه الاستعلامات في باني الاستعلام الخاص بالعروض Views، فهو لا يتعامل إلا مع استعلامات التحديد.. فإذا أردت تجربتها، فعليك بإضافة أداة أمر DataCommand للنموذج (لا بد أن تضع أداة من النوع DataConnection أولا، وتربطها بقاعدة البيانات، وتربط أداة الأمر بها).. عندئذ يمكنك وضع استعلامات الأداء في خاصية ComandText الخاصة به.. وعلى فكرة: لو ضغطت زرّ الانتقال المجاور لخانة قيمة هذه الخاصيّة في نافذة الخصائص، فستجد أنّه يفتح لك نافذة باني الاستعلام.. هنا تستطيع تجربة هذه الأوامر.

طبعا لا تشعر أنّك منسجم مع كلامي هذا.. شيء طبيعيّ، فكلّ هذا سيُشرح بالتفصيل في الفصل القادم بإذن الله!

ويمكنك أن تؤدّى بهذه الاستعلامات العمليّات التالية:

* [**حذف الصفوف Deleting Rows:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\4-%20استعلامات%20الأداء\1-%20حذف%20الصفوف.htm)
* [**إدراج سجلات جديدة Inserting New Rows:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\4-%20استعلامات%20الأداء\2-%20إدراج%20سجلات%20جديدة.htm)
* [**تحرير السجلات الموجودة Editing Existing Rows:**](file:///C:\Users\Administrator\Desktop\vb.net\e-%20%20Databases%20قواعد%20البيانات\16-%20SQL%20لغة%20الاستعلام%20المركّبة\4-%20استعلامات%20الأداء\3-%20تحرير%20السجلات.htm)

**حذف الصفوف Deleting Rows:**

يمكنك أن تحذف أيّ عدد تريده من الصفوف تبعا للشرط الذي تحدّده في مقطع WHERE.. والجملة التالية تريك كيف يمكن حذف كلّ الكتب التي تحمل رقم 3:

**DELETE Books**

**WHERE ID = 3**

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

**DELETE 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.

**إدراج سجلات جديدة Inserting New Rows:**

لجملة الإدراج الصيغة التالية:

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

حيث ستضاف القيم للحقول تبعا لترتيب كتابة أسمائها.. ويمكن ألا تكتب كلّ أسماء الحقول (لو أردت أن تترك بعض الحقول فارغة)، ولكن يجب أن يكون عدد القيم مساويا لعدد أسماء الحقول.

فإذا أردت أن تُدخل قيما لكلّ الحقول، فيمكن ألا تكتب أسماءها، على أن تكتب القيم مرتّبة تبعا لترتيب الحقول الأصليّ في الجدول، تبعا للصيغة التالية:

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

مثال: لإضافة حقل جديد لجدول الكتب، استخدم الجملة التالية:

**INSERT INTO Books (Book, AuthorID)**

**VALUES ('نجم الخيال الضاحك', 6)**

لاحظ أنّنا لم نضع قيمة للحقل ID لأنّه يُولّد تلقائيّا، حيث إنّنا جعلناه ترقيما تلقائيّا عند إنشائه.

ويمكن أن تنسخ مجموعة من السجلات من جدول لآخر (على أن يكون لحقولهما نفس نوع البيانات)، باستبدال فقرة VALUES بجملة SELECT.

افترض أنّ عندنا جدولا مماثلا لجدول الكتب، سنضع فيه بعض الكتب مؤقّتا لأيّ سبب، هذا الجدول اسمه TempBooks.. 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**

وقديما في VB6 كان بإمكانك إضافة السجلات لجدول من قاعدة بيانات أخرى (ولتكن C:\Db1.mdb)، بالجملة التالية:

**INSERT INTO TempBooks IN 'C:\Db1.mdb'**

**SELECT Books.\***

**FROM Authors, Books**

**WHERE Author = 'توفيق الحكيم' AND AuthorID = Authors.ID**

إنّ هذه الجملة ما زالت متاحة ـ على الأقل مع قواعد بيانات Access، وذلك بوضعها في خاصية CommandText لأمر البيانات من النوع OleDbCommand (سنتعرّف على هذه الأمور في الفصل القادم بإذن الله).

**سؤال:**

هل توجد وسيلة مباشرة لنقل البيانات بين نوعين مختلفين من قواعد البيانات؟

**الإجابة:**

لا أعرف صراحة إن كانت جمل SQL (مثل الجملة السابقة) قادرة على نقل البيانات بين نوعين مختلفين من قواعد البيانات أم لا.

ولكنّي أقترح حلاًّ لهذه المشكلة:

حيث يمكنك تحميل البيانات من أحد النوعين من قواعد البيانات في مجموعة بيانات DataSet، ثم تقوم بحفظها منها في قاعدة البيانات الأخرى (يمكن أن تتعامل مجموعة مع أكثر من موصل DataAdapter من أنواع مختلفة).

لم أجرّب ذلك.. عليك بتجربته.. طبعا بعد أن تنتهي من دراسة الفصل القادم بإذن الله!

**تحرير السجلات الموجودة Editing Existing Rows:**

لتغيير قيم بعض ـ أو كلّ ـ سجلات أحد الجداول، استخدم جملة UPDATE، التي لها الصيغة التالية:

**UPDATE اسم الجدول SET الحقل1 = القيمة1, الحقل2 = القيمة2, ……..**

**WHERE شرط**

مثال: استخدم الجملة التالية لتغيير اسم المؤلّف (منصور) إلى (عايش):

**UPDATE Authors SET Author = 'عايش'**

**WHERE Author = 'منصور'**

**لغة تعريف البيانات Data Definition Language (DDL):**

قلنا إنّ هذه الاستعلامات تختصّ بإنشاء أو حذف كائنات قاعدة البيانات، مثل الجداول وما تحتويه من أعمدة وفهارس، ومثل العلاقات والقيود Constrains.

ولن نتوسّع في شرح هذه الأوامر في هذا المرجع، 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**

والأوامر التالية، وهي تستخدم لتعديل عناصر قاعدة البيانات:

**ALTER DATABASE**

**ALTER TABLE**

**ALTER PROCEDURE**

**ALTER FUNCTION**

**ALTER VIEW**

ويمكنك الاسترشاد بهذه الأمثلة (وهي خاصة بالتعامل مع قواعد بيانات Access):

الجملة التالية تنشئ جدولا للمؤلفين، به عمودان يستوعبان نصوصا لا يزيد طولها عن 30 حرفا، مع ملاحظة أنّ العمود الثاني مفهرس (بإنشاء الفهرس الذي اسمه X عليه) 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 يتمّ حفظها في قاعدة البيانات، وتنفيذها على خادم الإنترنت Sserver، لأداء وظيفة معيّنة على قاعدة البيانات.

إنّ أيّ حدثٍ تقوم به دائما عند التعامل مع قاعدة البيانات يجب أن تكتبه كإجراء مخزّن، حتّى تستطيع استدعاءه من تطبيقات مختلفة أو من أكثر من موضع من نفس التطبيق.. إنّ ذلك لن يوفّر لك الجهد فحسب، ولكنّه سيوفّر كذلك وقت التنفيذ، حيث إنّ SQL Server يقوم بترجمة Compile الإجراءات المخزّنة مرّ واحدة عند إنشائها، وبالتالي فهي تعمل أسرع من جمل SQL العاديّة.

وتحتوي الإجراءات المخزّنة على جمل البرمجة العاديّة للتحقّق من صحّة المعاملات وغير ذلك من المهامّ.. وتسمّى اللغة التي تكتب بها الإجراءات المخزّنة T-SQL اختصارا لـ (Transact-SQL)، وهي تعتبر نسخة مطوّرة من SQL.

ويمكنك أن تعتمد على برامج المعالجات السحرية Wizards لتحويل جمل SQL إلى إجراءات مخزّنة، وإن كان من الأفضل أن تفهم كيفية عملها، حتّى تستطيع تطويعها لخدمة أغراضك.

يبدأ الإجراء بجملة التعريف التالية:

**ALTER PROCEDURE dbo.StoredProcedure1**

حيث إنّ StoredProcedure1 هو اسم الإجراء.. طبعا يمكنك تغييره إلى أيّ اسم يناسبك.

بعد ذلك يأتي تعريف معاملات الإجراء.. إنّ كلّ المتغيّرات في لغة T-SQL تبدأ بالرمز @.. والجملة التالية تعرّف معاملا نصّيا طوله ثلاثة أحرف:

**@Str1 nvarchar(3)**

كما ترى: تكتب اسم المعمل، يليه نوعه مشفوعا بطوله.

وكأيّ إجراء، تفصل بين المعاملات علامة ",".

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

**@Str2 nvarchar(7) = 'افتراضي'**

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

ويبدأ الإجراء المخزّن بكلمة AS، وينتهي بكلمة RETURN، وبينهما يكون الكود:

**AS**

**………..**

**RETURN**

ويمكن أن تعرّف المتغيّرات داخل الإجراء، عن طريق جملة DECLARE.. فمثلا:

**DECLARE @Str3 nvarchar(7)**

ويمكن كتابة جمل الشرط كالتالي:

**IF @Str3 = ''**

**BEGIN**

**SELECT @Str3 = 'ABC'**

**END**

لاحظ استخدامنا لتعبير SELECT لتغيير قيمة المتغيّر @Str3.. يمكنك كذلك استخدام تعبير SET لتغيير قيمة المتغيّر، وأعتقد أنّه لن يكون مربكا لك مثل تعبير SELECT:

**SET @Str3 = 'ABC'**

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

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

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

في متصفّح خوادم الإنترنت Server Explorer، أسدل العناصر التي تنتمي لرابطة قاعدة البيانات Books 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:

**ALTER PROCEDURE dbo.GetAuthorBooks**

**(**

**/\*لدينا معامل واحد فقط نستقبل فيه اسم المؤلّف**

**وقد جعلناه 30 حرفا ليتلاءم مع طول حقل المؤلّفين \*/**

**@Author char(30)**

**)**

**AS**

**IF @Author <>'' /\* تأكّد أنّ اسم المؤلّف 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 أنّ الأمر في غاية البساطة.

ولا حاجة بي لألفت نظرك إلى أنّ 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 --:

**/\* هذا تعليق**

**على سطرين\*/**

**--بينما هذا تعليق في سطر واحد**

ولا بدّ أنّك لاحظت ظهور جملة 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.. هنا سيظهر لك نافذة باني الاستعلام.. صمّم جملة SQL التي تريد، وقم بحفظ العمل.. الآن لو عدت لنافذة الإجراء المخزّن، فستجد أنّ جملة SQL قد أضيفت للموضع الذي ضغطت فيه الفأرة.

ما علينا.. 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 تستعلمك عن قيمة معامل الإجراء.. اكتب اسم (توفيق الحكيم) واضغط OK.. سيتمّ تنفيذ الإجراء، حيث ستظهر أسماء كتب (توفيق الحكيم) في نافذة المخرجات Output Window.

كانت هذه نبذة سريعة عن الإجراءات المخزّنة.. وفي الفصل القادم بإذن الله سنرى كيف نستخدم هذه الإجراءات في البرمجة.

أمّا بخصوص T-SQL فإنّها تحتاج بمفردها لفصل مستقلّ، ليس مجاله الآن.. ويمكنك أن تستعين بملفّات إرشادات اللغة للحصول على معلومات تفصيليّة حول هذه اللغة.. من قائمة Help اضغط الأمر Contents، ستظهر لك نافذة شجرة المحتويات.. تأكّد أن القائمة المركّبة التي تحمل عنوان Filtered By في أعلى النافذة تحتوي على القيمة No Filter، ثمّ تتبّع الشجرة التالية:

**MSDN Library\ Enterprise Development\ .Net Enterprise Servers\ Microsoft SQL Server\ Microsoft SQL Server 2000\ Transact-SQL Reference**

عِش حياتك!