

سلسلة دروس تطبيقية حول SQL Server 2008



دروس مبسطة ومشرحة بالصور
وتطبيقات عملية مفصلة

الطبعة الأولى

إعداد

منتدى دلفي للعرب

بسم الله الرحمن الرحيم

السلام عليكم ورحمة الله...

الحمد لله رب العالمين والصلاة والسلام على أشرف المرسلين، أما بعد:

فقد يسر الله عز وجل لي جمع سلسلة من الدروس المتعلقة بخادم قواعد بيانات Microsoft SQL Server 2008 من بعض المراجع الموثوقة، ثم ترتيبها وتنسيقها في هذا الكتيب بحيث يسهل على المتبع الرجوع إليها في أي وقت، إسهاما لتغطية النقص الملموس في الكتاب العربي في مجال علوم الحاسب عموما وحول هذا الموضوع خصوصا...

فالحمد لله الذي وفقني لإتمام هذا العمل، وجزى الله خيرا كل من أسهم فيه من قريب أو بعيد، وبارك الله في أساتذتنا وإخواننا الذين لم ييخلوا علينا بما علموه ونفع الله بعلمهم، ونخص بالذكر منهم القائمين على المواقع والمنتديات العربية المتخصصة في البرمجة ومنتدى دلفي للعرب.

أسأل الله عز وجل أن يجعل هذا العمل خالصا لوجهه الكريم وأن يجعله زاد يوم نلقاه وألا يجعل فيه حظا للنفس ولا للشيطان، ولا تنسوا إخوانكم بالدعاء.

مدخل إلى Microsoft SQL Server

في هذه السلسلة سنقدم دروس تطبيقية ومواضيع أخرى حول تهيئة وكيفية استخدام Microsoft SQL Server 2008 من خلال Microsoft SQL Server Management Studio.

1. إعداد Microsoft SQL Server:

Microsoft SQL Server هو برنامج مخصص لنظام التشغيل Microsoft Windows يستخدم لإنشاء وإدارة قواعد بيانات الحاسب، وهو يوفر بيئة تستخدم لإدارة قواعد البيانات التي يمكن الوصول إليها من محطات العمل، شبكة الإنترنت، وغيرها من وسائل الإعلام مثل المساعد الرقمي الشخصي.

يكفي من أجل متابعة هذه السلسلة وتطبيق الأمثلة، نسخة من SQL Server 2008 وجهاز كمبيوتر مزود بأحد أنظمة التشغيل التالية¹:

- ❖ Microsoft Windows XP Professional
- ❖ Microsoft Windows Vista
- ❖ Microsoft Windows Server 2003
- ❖ Microsoft Windows Server 2008

أولا عليك تثبيت Microsoft SQL Server 2008 على حاسوبك، يمكنك تثبيته على أي من أنظمة التشغيل سألفة الذكر، وهذا يعني أن عليك أن تراعي نظام التشغيل المناسب.

قبل تثبيت Microsoft SQL Server 2008، تأكد من تثبيت كافة الحزم الخدمية الضرورية (Service packs) على نظام التشغيل الخاص بك، لو كنت تخطط لاستخدام Microsoft Visual Studio 2008 لإنشاء قواعد بيانات SQL Server 2008، يجب تثبيت Microsoft SQL Server 2008 أولاً، ثم تثبيت Microsoft Visual Studio 2008.

2. الأمن في قواعد البيانات:

أحد أهم المشاغل الرئيسية لقاعدة البيانات هو الأمن، وهو أمر أساسي عند Microsoft SQL Server، تطبق الحماية في قواعد البيانات على مستويات مختلفة: في الحجرة حيث يوجد

الحاسب، في الحاسب الذي يحوي البرنامج الخادم (SQL Server)، والحاسب العميل الذي سيستخدم قواعد البيانات (Clients)، والأشخاص الذين يستخدمون جميع هذه الموارد.¹

عندما يتعلق الأمر بالغرفة التي تضم الحواسيب حيث تم تركيب Microsoft SQL Server، يجب عليك التأكد من سلامة وأمن الغرفة؛ حيث يمكن أن تغلق في أي وقت ولا يصل إليها إلا فئة قليلة محددة من الأشخاص المخول لهم بذلك، ولتحقيق أقصى درجة أمان، ينبغي أيضا أن يوجد الحاسب في حجرة مقفلة!²

3. إنشاء حساب مشرف في نظام التشغيل Windows:

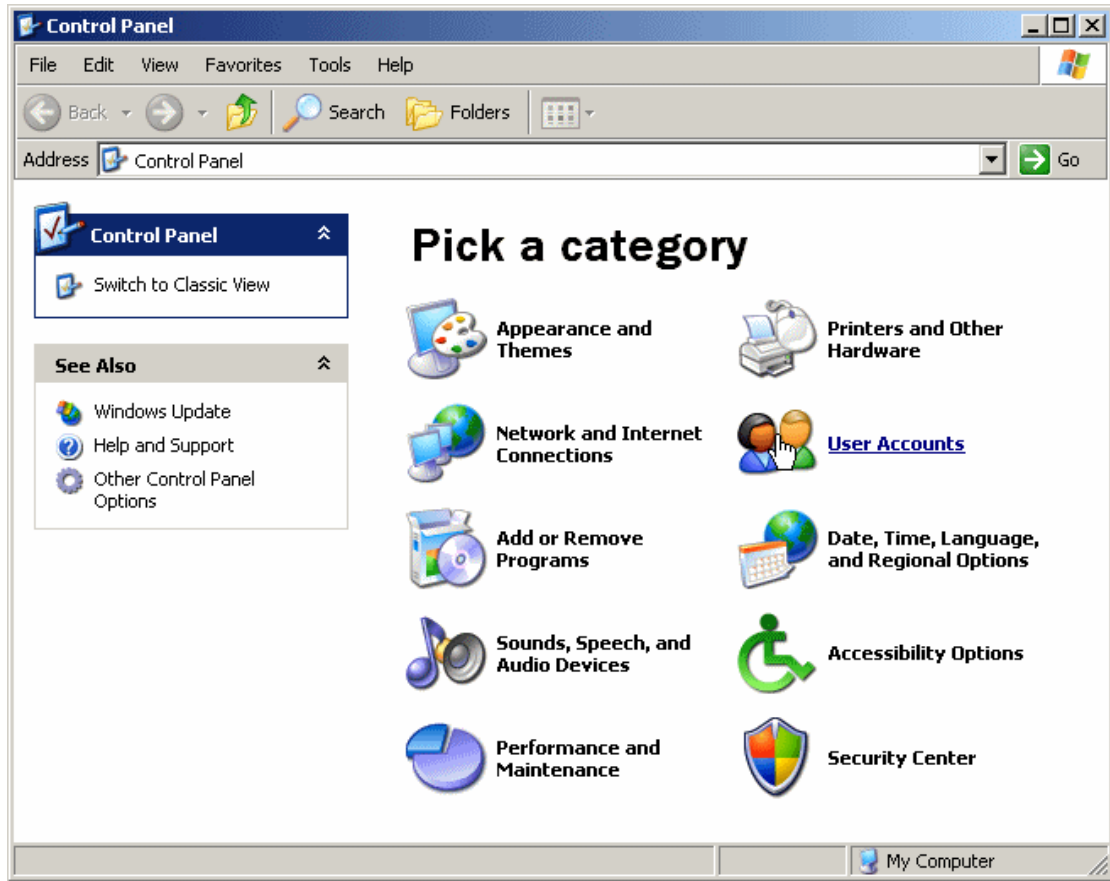
لتنصيب Microsoft SQL Server 2008 يجب استخدام حساب المستخدم الذي يمتلك القدرة على تنصيبه، إذا كنت تستخدم أحد إصدارات Microsoft Windows XP أو Vista، يجب عليك استخدام حساب المشرف (Administrator account) أو إنشاء حساب جديد يملك صلاحيات المشرف، أيضا يجب أن يتوفر الحساب الذي سنستخدمه على كلمة سر، يجب تعيين كلمة السر في نظام Microsoft Windows XP Professional أو Vista لاستخدام حساب المدير (المشرف)، بدلا من ذلك، يمكنك تسجيل الدخول للنظام كمدير، ثم إنشاء حساب يمتلك صلاحيات المدير.

تطبيق عملي: إنشاء حساب في نظام التشغيل XP أو Vista

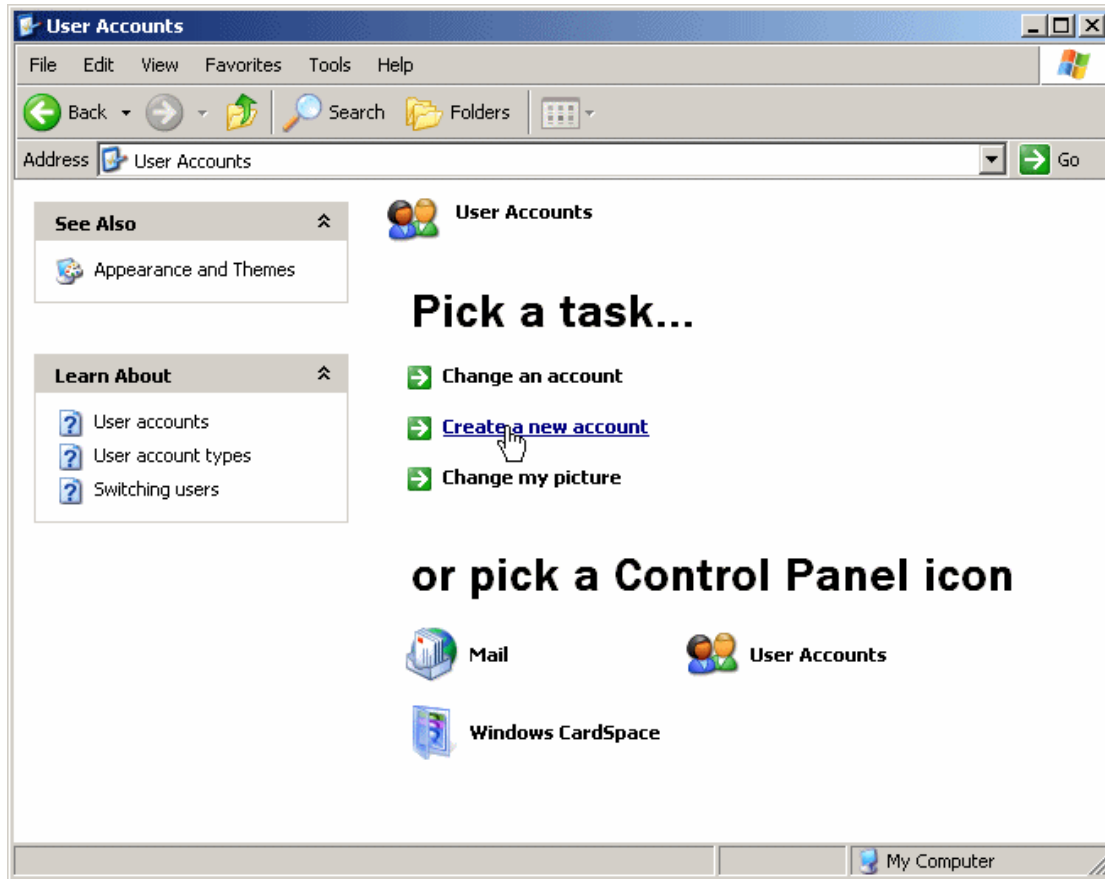
1- من Start حدد Control Panel وانقر على User Accounts:

¹ ذلك لأن الحماية في SQL Server تطبق أساسا على مستوى الخادم لا على مستوى ملف قاعدة البيانات، بعبارة أخرى: إمكانية الوصول إلى ملف قاعدة البيانات يعني إمكانية نسخه إلى جهاز آخر واستعراض كل أنواع الكائنات والبيانات.

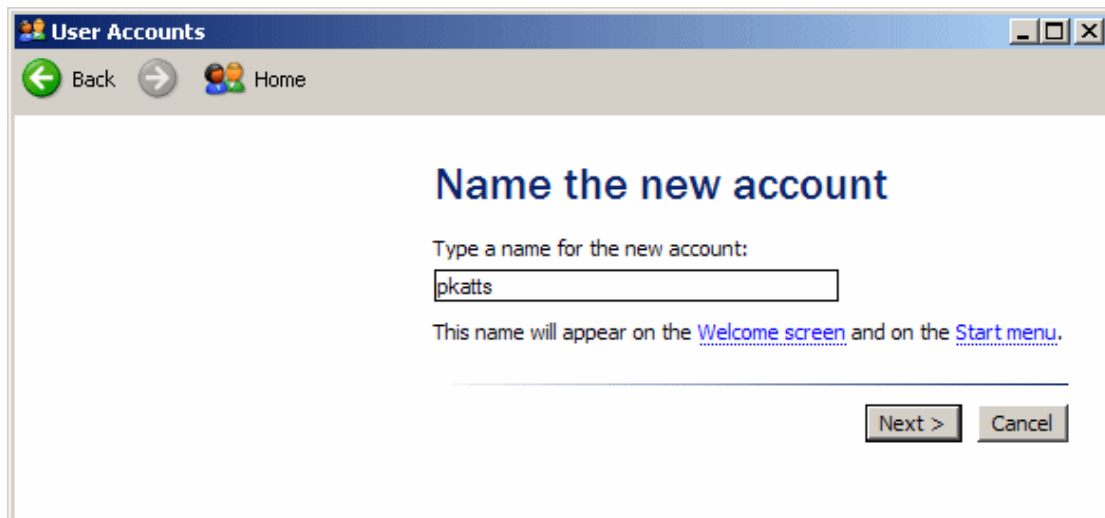
² هناك خيار متقدم يسمح بتشفير (Encryption) ملف قاعدة البيانات وبالتالي حمايته من النسخ. هذا الخيار غير متوفر في النسخة Express Edition.



2- انقر على Create a new account:

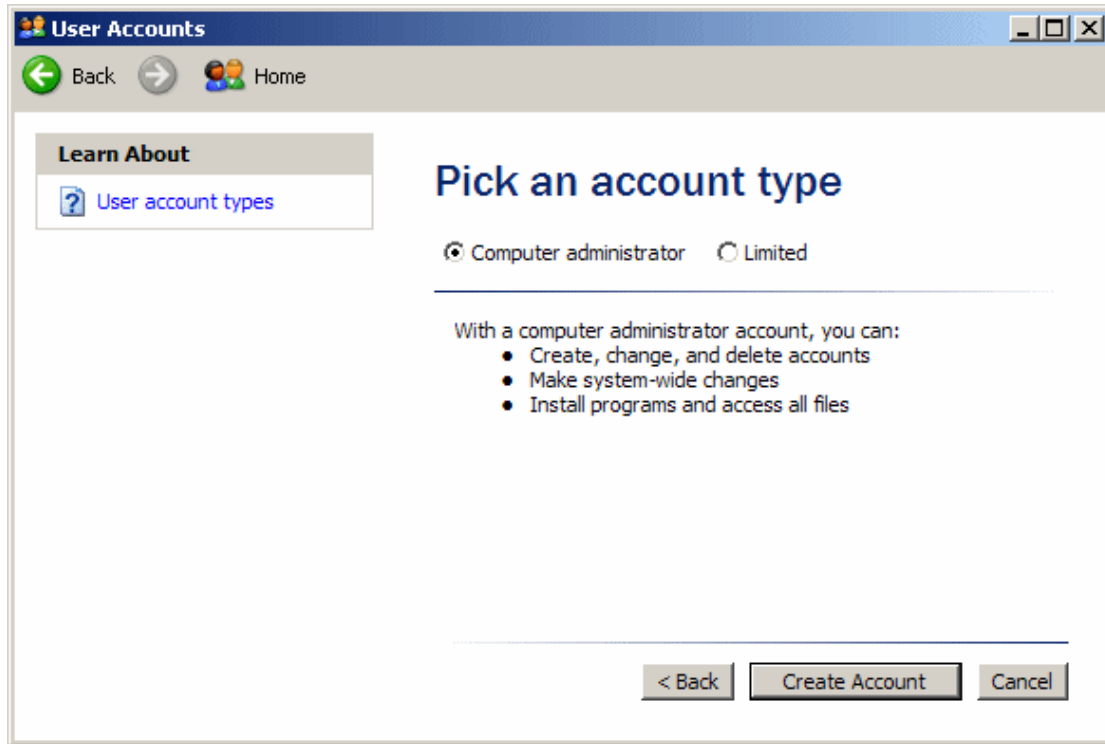


3- اكتب الاسم الجديد للحساب pkatts:



4- انقر على Next:

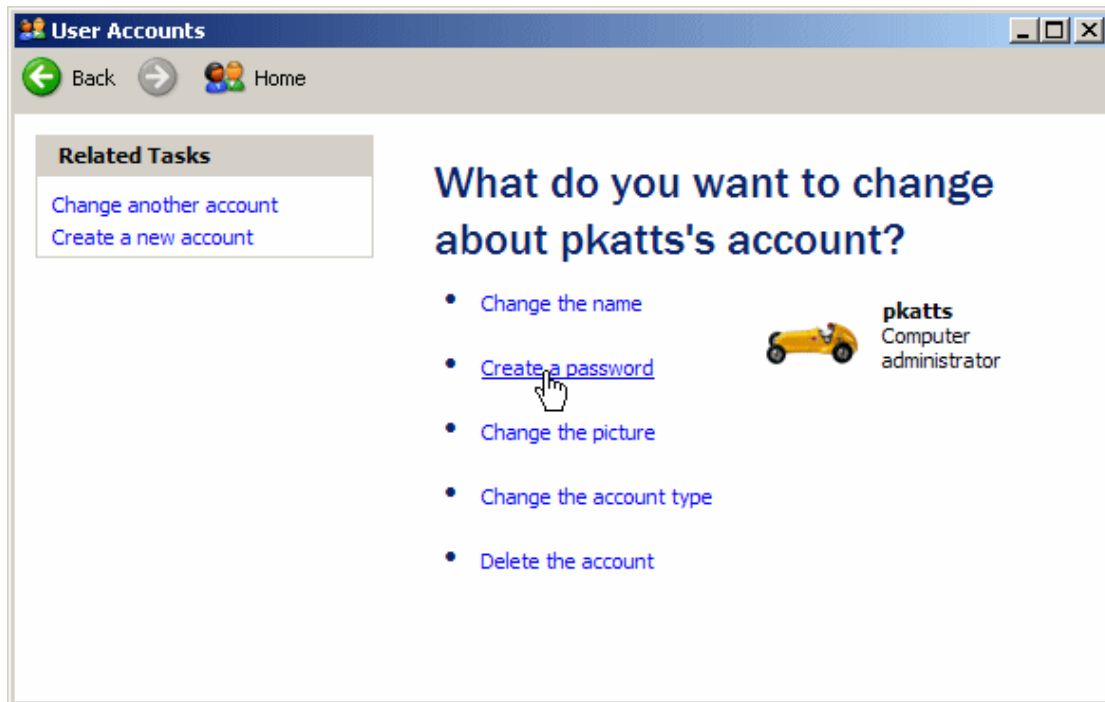
تأكد من تحديد خيار مدير الحاسب (Computer Administrator).



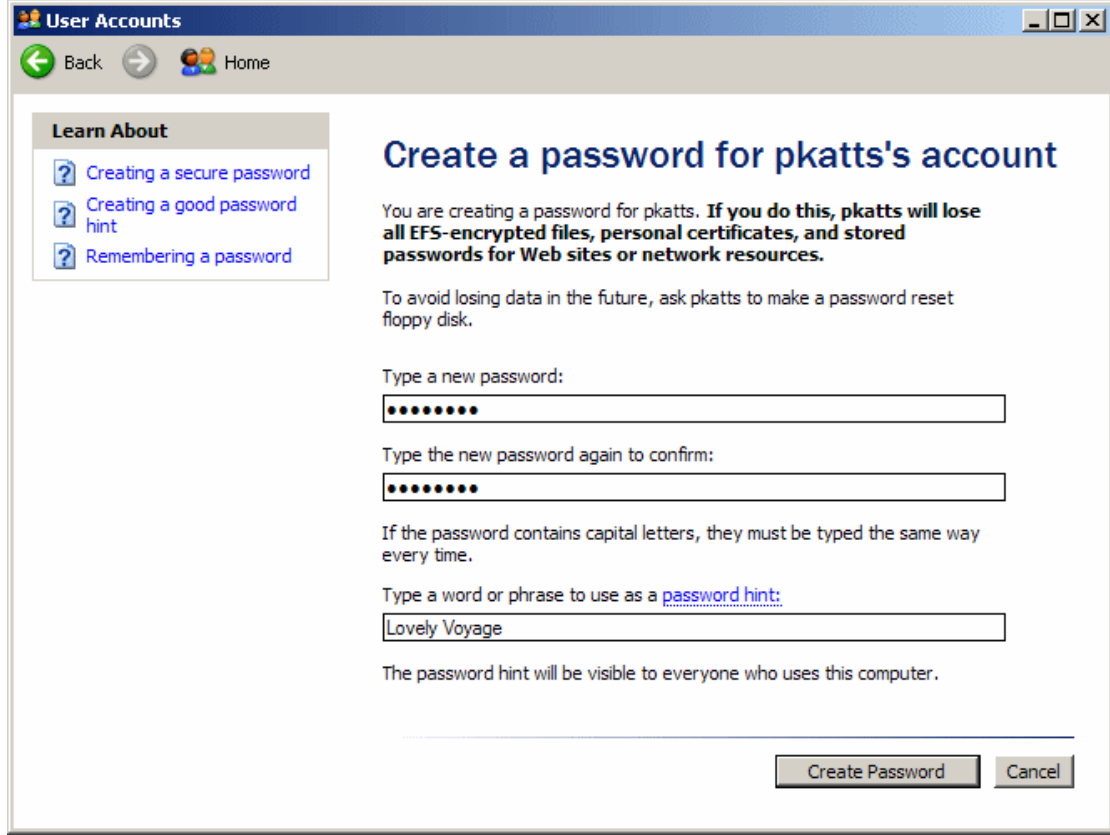
5- انقر Create Account

6- لتعيين كلمة المرور، في حساب المستخدم انقر على حساب pkatts

7- انقر على إنشاء كلمة السر (Create a password).



8- في خانة Type a new password اكتب Password1 واضغط مفتاح Tab ثم أدخل Password1 مرة أخرى للتأكيد، ثم أدخل أي شيء في الخانة الأخرى:



9- انقر فوق إنشاء Create Password

10- بنفس الطريقة أنشئ حساب آخر باسم gmonay بصلاحيّة مدير مع كلمة السر Password1

تطبيق عملي: إنشاء حساب في الخادم على Microsoft Windows Server

1- سجل دخولك إلى Microsoft Windows Server 2003 أو 2008، بحساب يسمح لك بإنشاء حساب مدير.

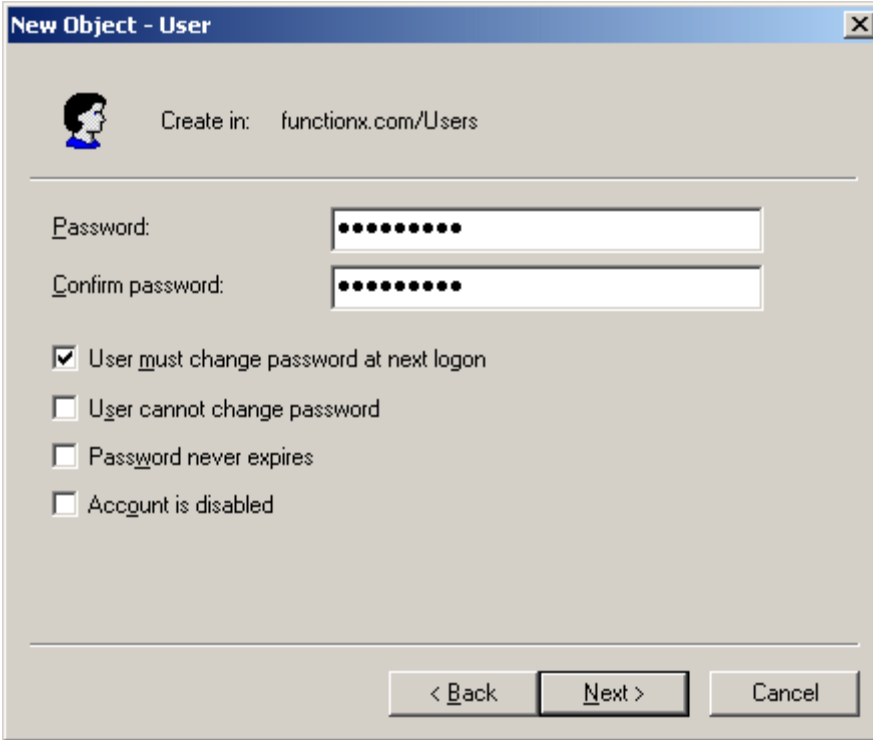
2- لإنشاء حساب في Microsoft Windows Server 2003 أو 2008، انقر على: Start ->

.Administrative Tools -> Active Directory Users and Computers

3- في الإطار الأيسر، انقر بالزر الأيمن Users -> New User

6- اكتب كلمة السر Password1

7- اكتب تأكيد كلمة السر Password1، واقبل الخيارات في مربعات الاختيار كما هي:



New Object - User

Create in: functionx.com/Users

Password: [dots]

Confirm password: [dots]

User must change password at next logon

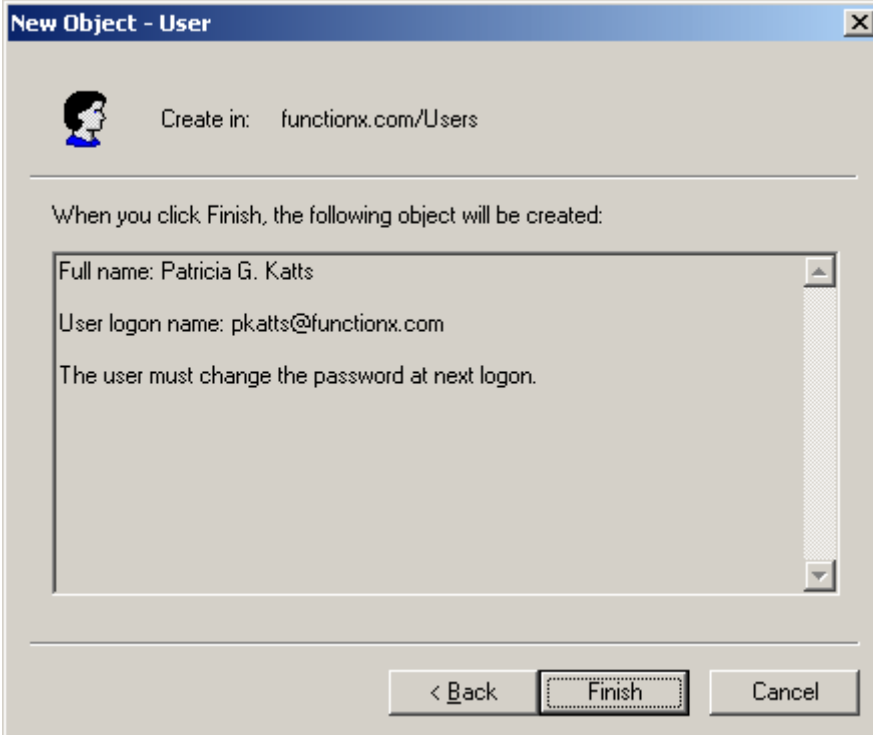
User cannot change password

Password never expires

Account is disabled

< Back Next > Cancel

8- انقر على Next (في النهاية، اسم التسجيل سيكون مختلفا عن الخاص بك)



New Object - User

Create in: functionx.com/Users

When you click Finish, the following object will be created:

Full name: Patricia G. Katts

User logon name: pkatts@functionx.com

The user must change the password at next logon.

< Back Finish Cancel

9- اقبل الخيارات في الصفحة الأخيرة من المعالج وانقر على Finish

بنفس الطريقة، قم بإنشاء حساب آخر وفق المعطيات التالية:

First Name: **Gertrude**

LastName: **Monay**

User Logon Name: **gmonay**

Password: **Password1**

4. الأمين الرئيسي:

الأمين الرئيسي هو الشخص أو الغرض¹ الذي يريد إلى الولوج إلى قاعدة بيانات، يمكن اعتبار الأمين الرئيسي كيان واحد أو مجموعة من الكيانات:



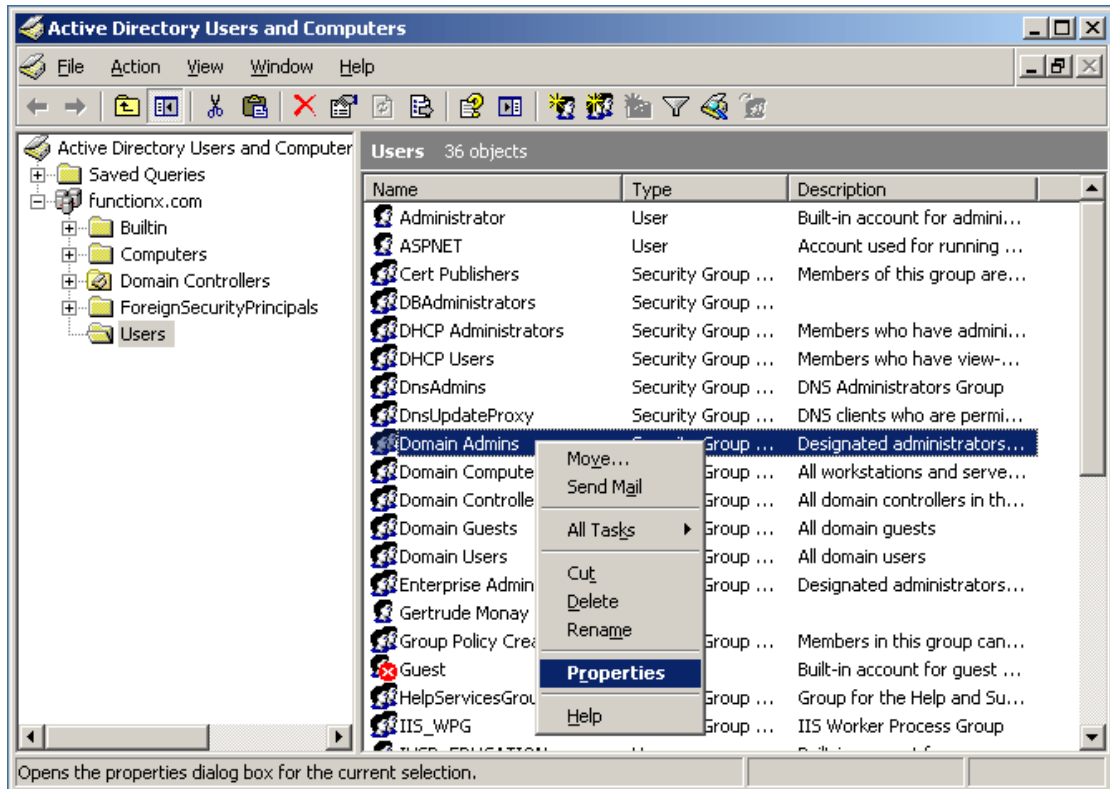
يعني هذا، أن الأمين الرئيسي يمكن أن يعتبر كيان وحيد أو طرف من المجموعة، مثلا الشخص الذي يلج قاعدة بيانات يعتبر كيانا، والفريق الذي يمارس نفس الوظيفة ويقدم نفس الوظيفة على قاعدة بيانات يعتبر مجموعة.

لتشكيل فريق في Microsoft Windows Server 2003 أو 2008، افتح: Active Directory

Users and Computers, بزر اليمن Group -> New -> Users node.

¹ يعتبر SQL Server كل من يقوم بأي عمل على قاعدة البيانات أن يكون مخلولا له بذلك، سواء كان مستعمل بشري (موظف، عامل...) أو كائن برمجي (تابع، قادح (Trigger)، استعلام (SQL)...).

أو انقر بالزر الأيمن على المجموعة وانقر **Properties**:



أضف المعلومات اللازمة.

للتحكم في طريقة دخول الأمين الرئيسي إلى جهاز أو قاعدة بيانات تستخدم المستويات، حيث هناك ثلاثة مستويات: Windows، وخادم، وقاعدة بيانات:



استنادا إلى وظيفة الأمين الرئيسي يمكن اعتباره منتمي إلى المجموعة.

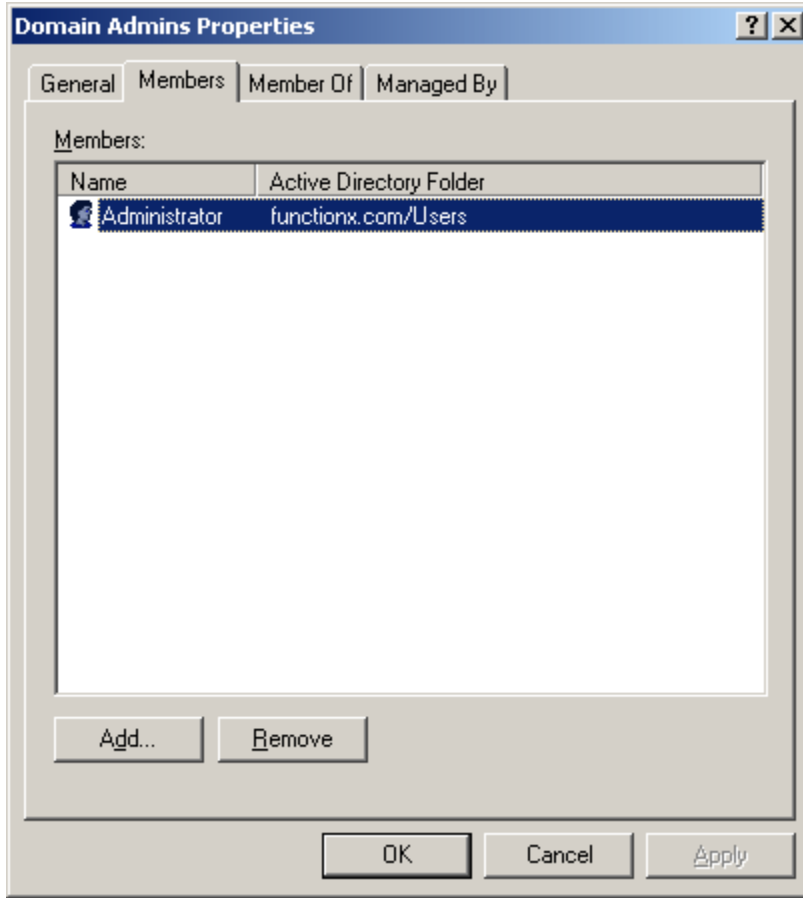
لمساعدة نظام التشغيل في التعرف عليهم، كل طرف رئيسي يستخدم رقم وحيد يدعى Security Identifier (SID).

تطبيق عملي: إضافة حساب إلى مجموعة

1- في الإطار الأيسر، انقر على Users إذا لزم الأمر.

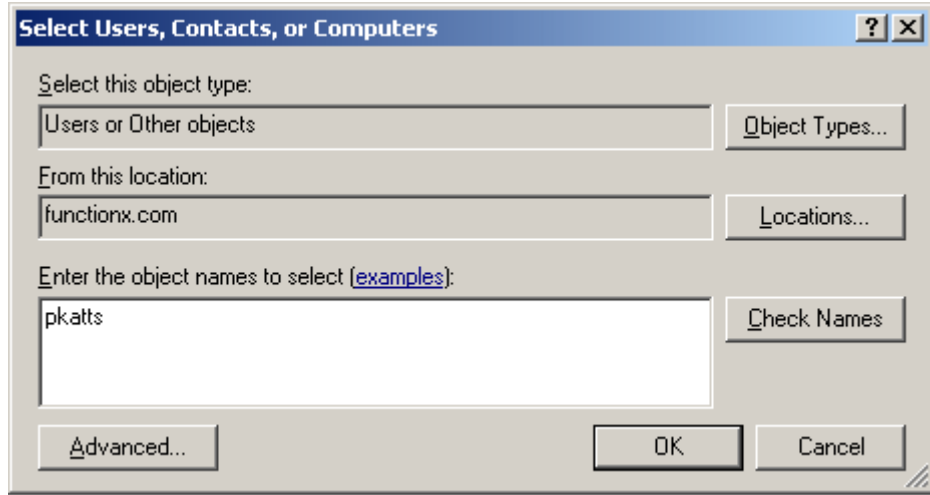
في الجانب الأيمن انقر بالزر الأيمن على الإطار، ثم انقر نقرا مزدوجا على Domain Admins

2- في Domain Admins Properties، حدد التويب Members (اسم المجال (Domain) سيكون مختلفا عن الخاص بك)



3- انقر للإضافة Add...

4- في مربع النص Enter the object names to select للإطار Select Users, Contacts, or Computers، أكتب .pkatts.



5- انقر على Check Names للتحقق من الاسم (وفي النتيجة اسم الحقل سيكون مختلفا عن الخاص بك).

6- انقر على OK ثم OK للموافقة.

5. الأدوار وتسجيل الدخول

الأدور هي تصرف أو مجموعة من التصرفات التي تمنح للأمين الرئيسي. مثلا، الشخص A يمكنه أن ينشئ ويستخدم قاعدة بيانات، يعبر عن الصلاحية للقيام بمثل هذا العمل بالدور، شخص آخر B يمكنه فقط استخدام قاعدة البيانات دون إنشاء قاعدة جديدة، وهذا نوع آخر من الأدوار.

من أجل الحصول على هدف ما (مثل الكمبيوتر أو قاعدة بيانات)، لا بد أن يكون الغرض مخولا له بذلك، مثل غرض يمكنه استخدام اسم مستخدم وكلمة سر واستخدام المجموعة التي ينتمي إليها، لهذا السبب سميت القدرة على اكتساب صلاحية الدخول لقاعدة بيانات أو مورد بتسجيل اسم الدخول (Login).

سنتطرق إلى إنشاء وكيفية تسجيل الدخول في درس لاحق إن شاء الله.

6. للحصول على Microsoft SQL Server

تصدر شركة Microsoft، لتمكينك وطلاب المعاهد من دراسة قواعد البيانات، نسخة تجريبية من Microsoft SQL Server 2008. هذه النسخة المتكاملة متوفرة للتحميل من الموقع

الرسمي لشركة Microsoft، للحصول على النسخة التجريبية من Microsoft SQL Server 2008، انتقل إلى موقع شركة Microsoft، ابحث عن عبارة "trial software"، وتتبع الوصلات لتحديد مكان النسخة التجريبية Microsoft SQL Server 2008 Enterprise Edition (وهي في ملف كبير نسبياً بحجم GB3 وبلاحقة ISO)¹.

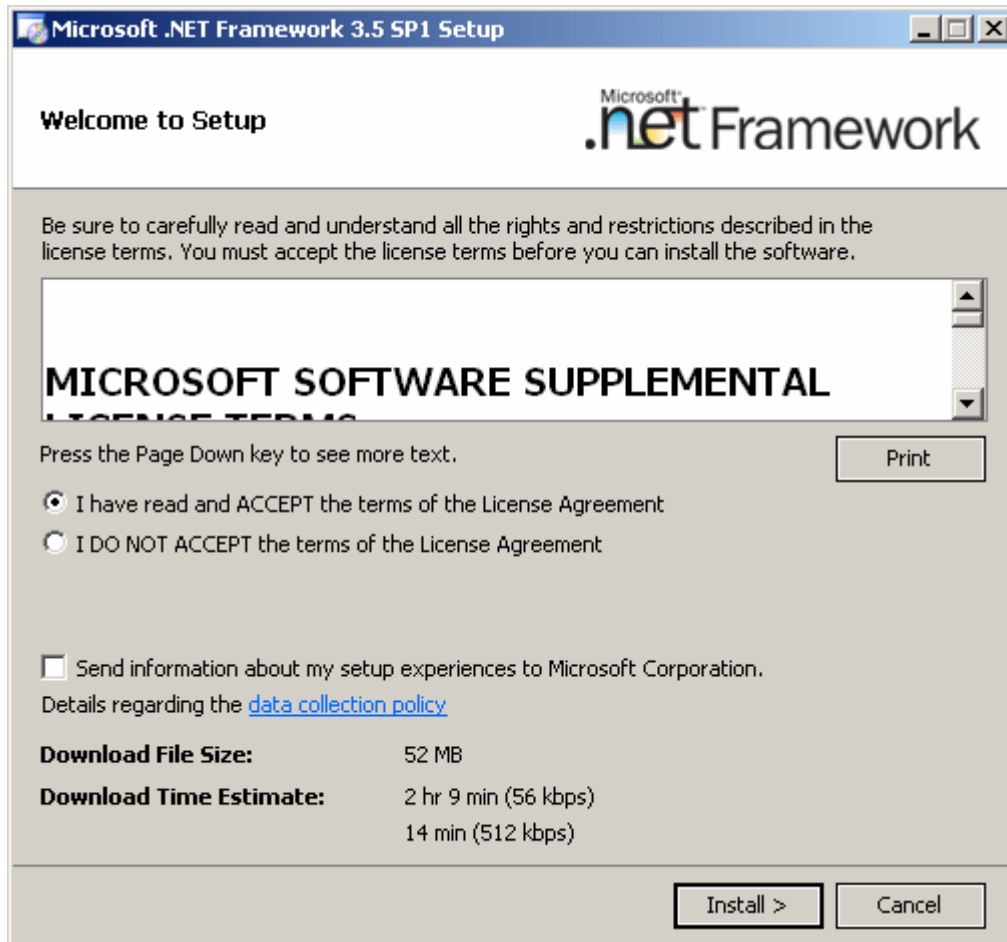
يمكنك التحميل من أي جهاز يتوفر على مساحة كافية، بعد التحميل انسخ الملف في قرص DVD ثم ركبه على محرك الأقراص لديك.

تطبيق عملي: تثبيت Microsoft SQL Server 2008

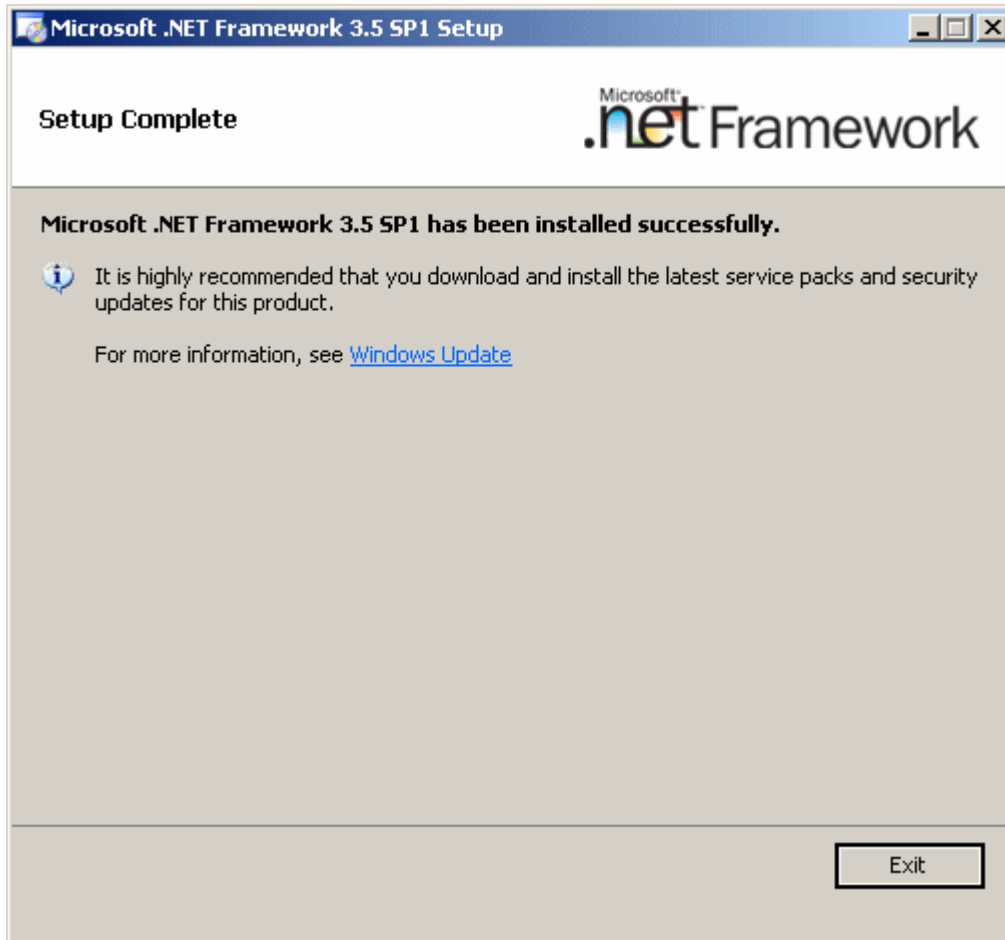
- 1- شغل الحاسب باستخدام حساب المدير، سجل باسم مستخدم وكلمة سر.
- 2- ضع قرص DVD الخاص ببرنامج Microsoft SQL Server 2008 وشغل برنامج التركيب، يمكن أن تشاهد رسالة مفادها أن حزمة .NET Framework لا بد من تثبيتها، انقر على موافق²، سيطلب منك بدء التثبيت:

¹ لمتابعة دروس هذه السلسلة يكفي تحميل واستخدام نسخة MS SQL Server Express Edition المجانية بحجم أقل من 100 Mb، ونسخة من MS SQL Server Management Studio 2008 Express المجانية بحجم يقارب 200 Mb.

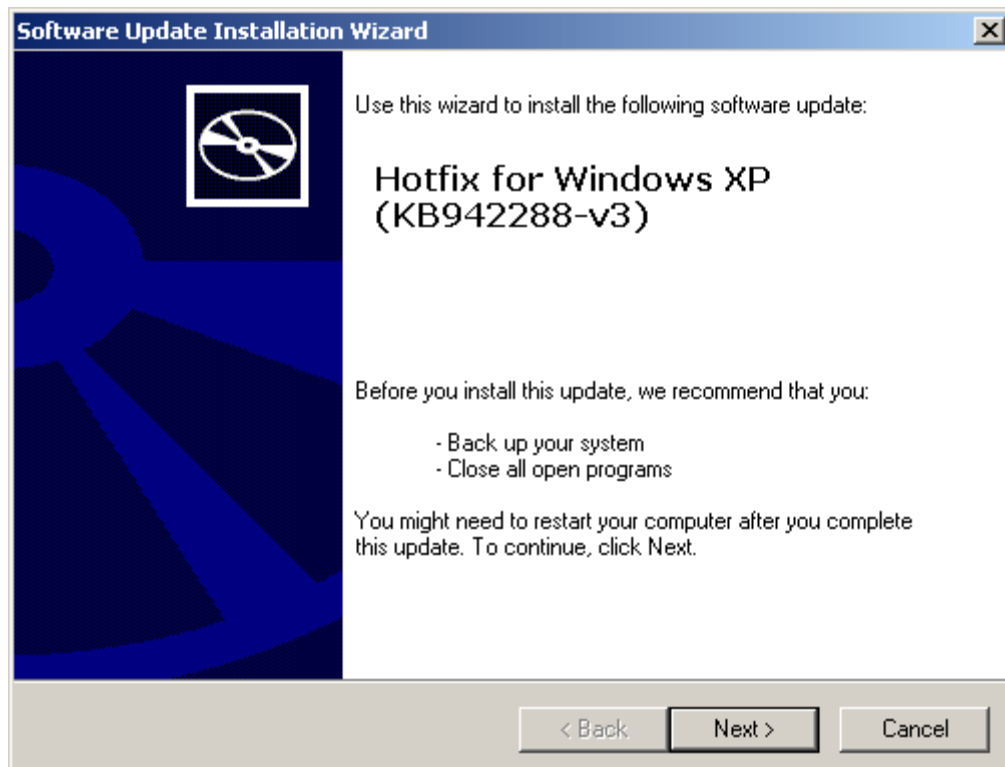
² يتطلب SQL Server 2008 إصدار Microsoft .NET Framwork 2.0 SP2 فما فوق وإصدار Windows Installer 4.5 فما فوق، أما SQL Server Management Studio 2008 فيتطلب Microsoft .NET Framework 3.5 SP1 أو نسخة أحدث.



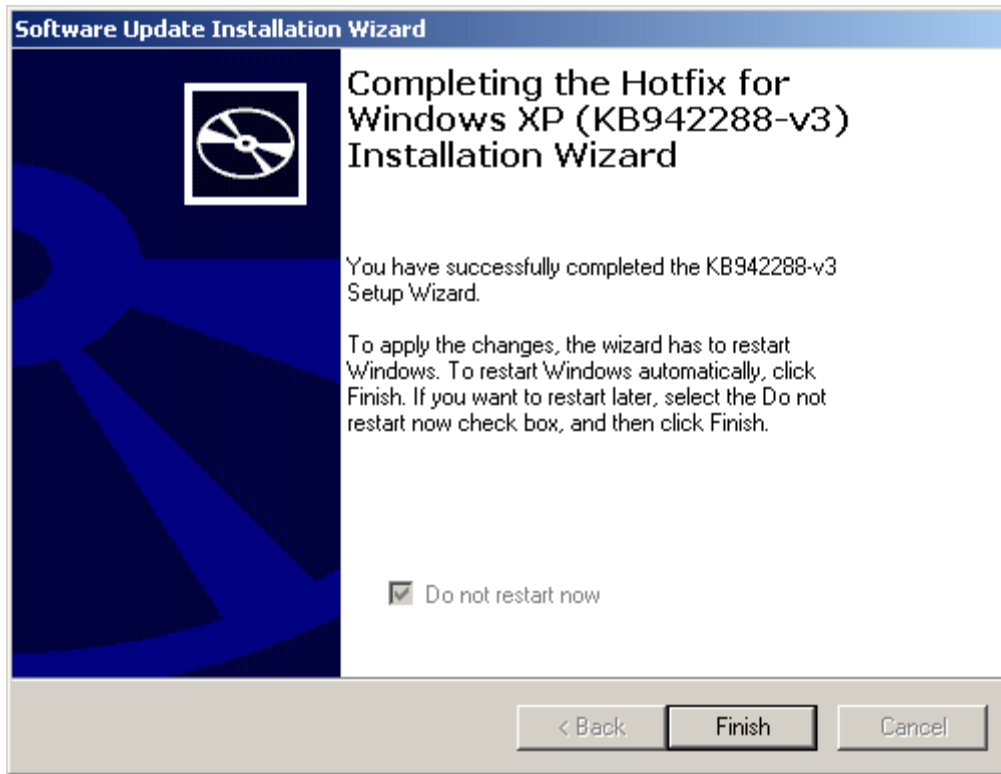
بعد الموافقة، انقر على Install ليبدأ تحميل ملفات التنصيب ثم تركيب .NET Framwork على جهازك، في النهاية تتلقى رسالة:



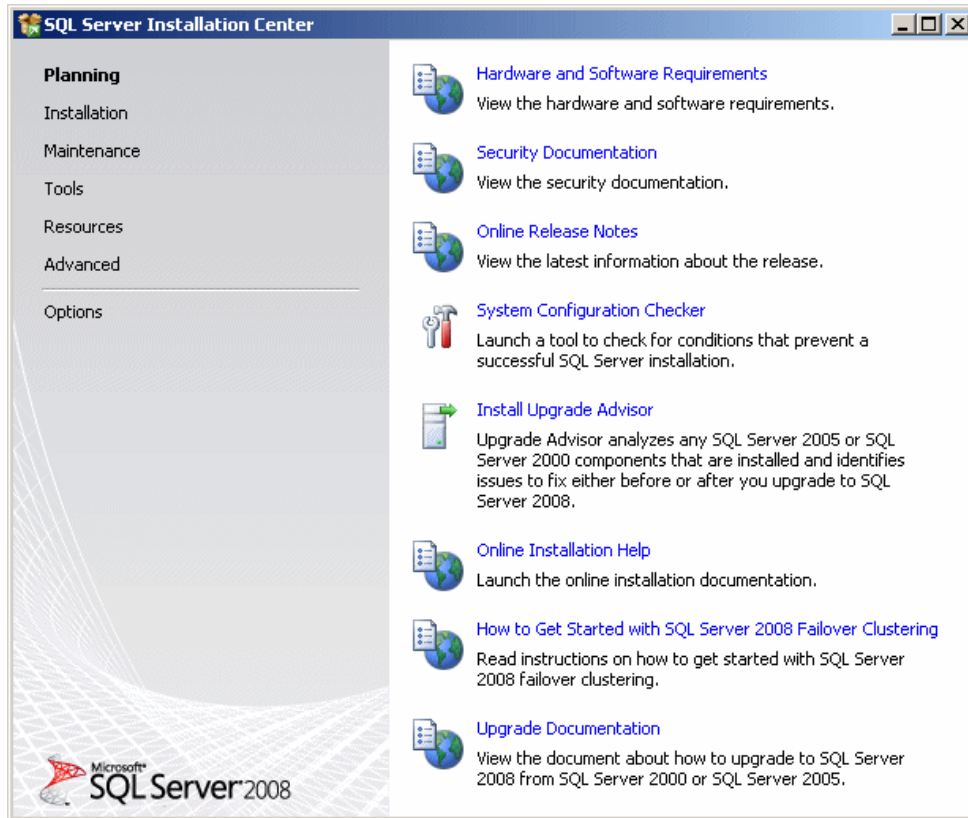
من هنا، يمكنك النقر على إنهاء وقد يطلب منك تثبيت Hotfix:



اقبل ذلك للثبيت، ثم اتبع المعالج حتى نهاية التركيب، هنا قد يطلب منك إعادة تشغيل النظام:



لبدء التثبيت، شغل قرص Microsoft SQL Server 2008 DVD، لتظهر نافذة:



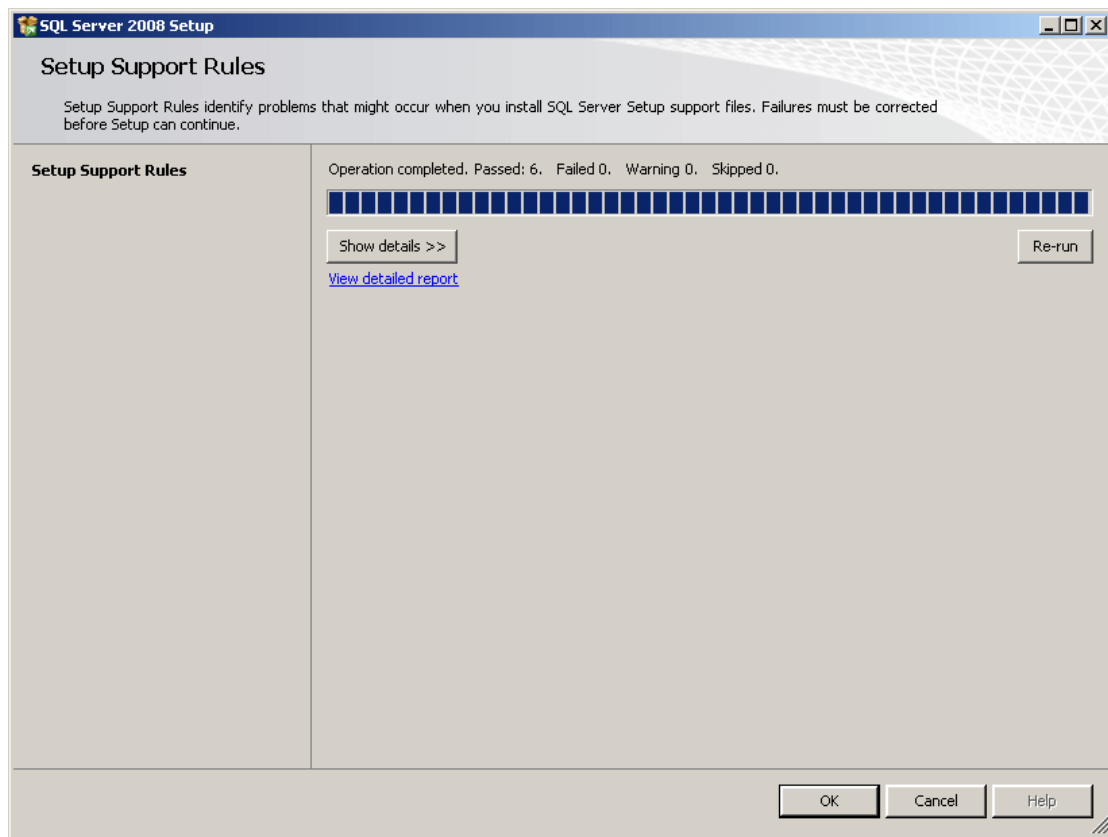
في الجزء الأيسر انقر على Installation



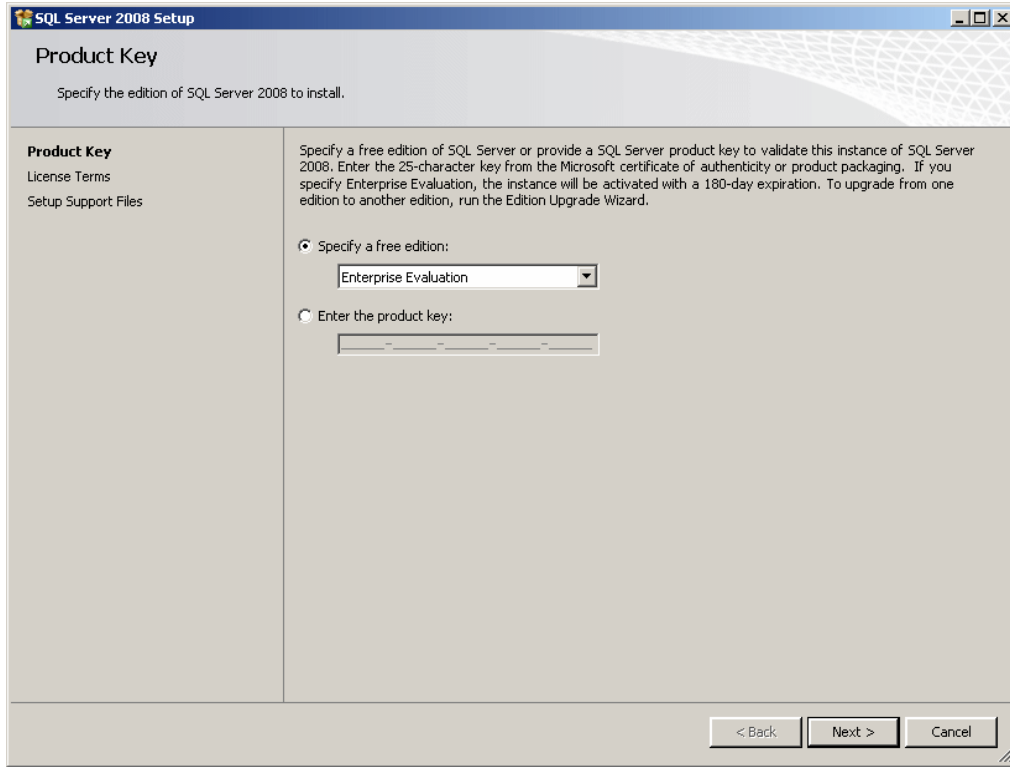
إذا كنت مهتما بدراسة تصميم قواعد البيانات، انقر من الجانب الأيمن على New SQL Server Stand-Alone ... وسيبدأ تركيب:



بعد مربع الرسالة، انقر على موافق



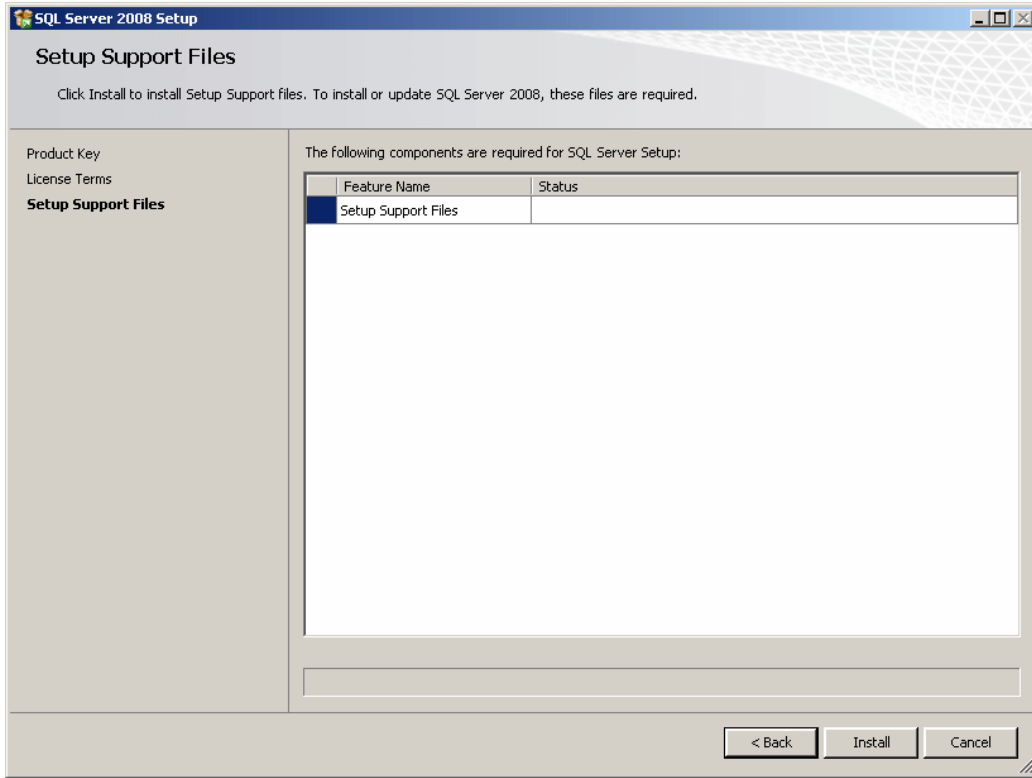
ستظهر رسالة أخرى، عندما تنتهي من الفحص ستظهر الصفحة التالية من المعالج، إذا كنت تستخدم النسخة التجريبية، اقرأ الرسالة وانقر على Next.



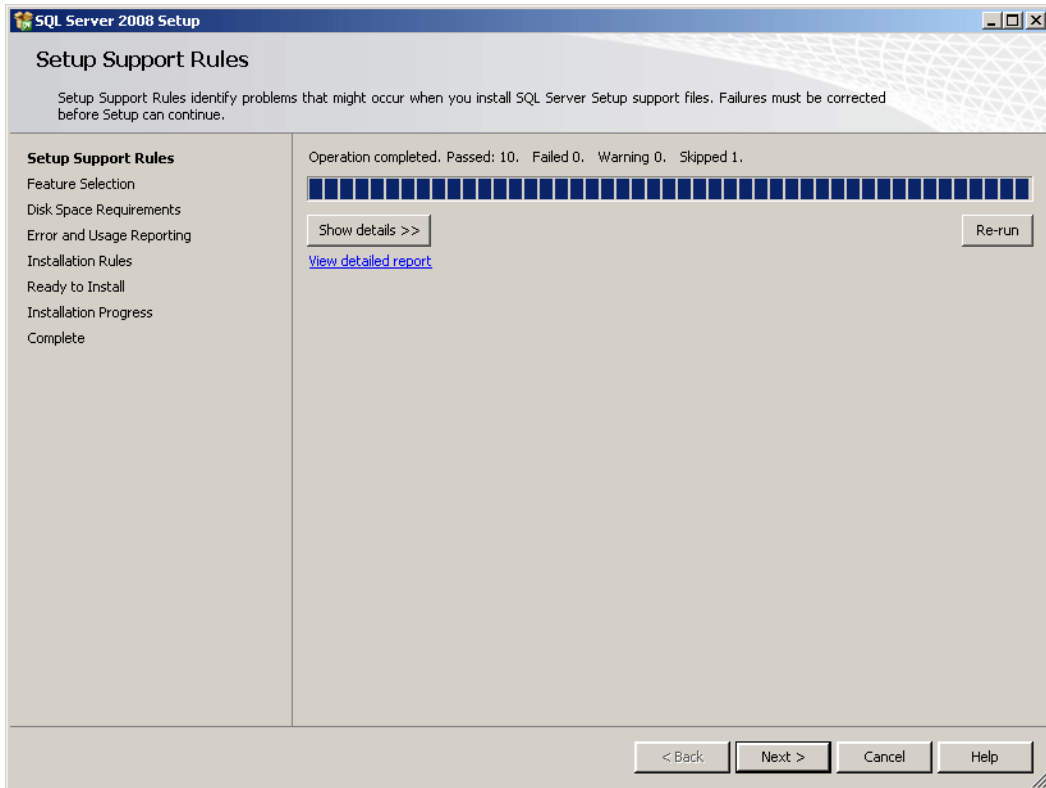
ستظهر صفحة شروط الترخيص، اقرأها إذا كنت لا توافق على الترخيص انقر على إلغاء، غير ذلك انقر على مربع أوافق ثم على التالي:



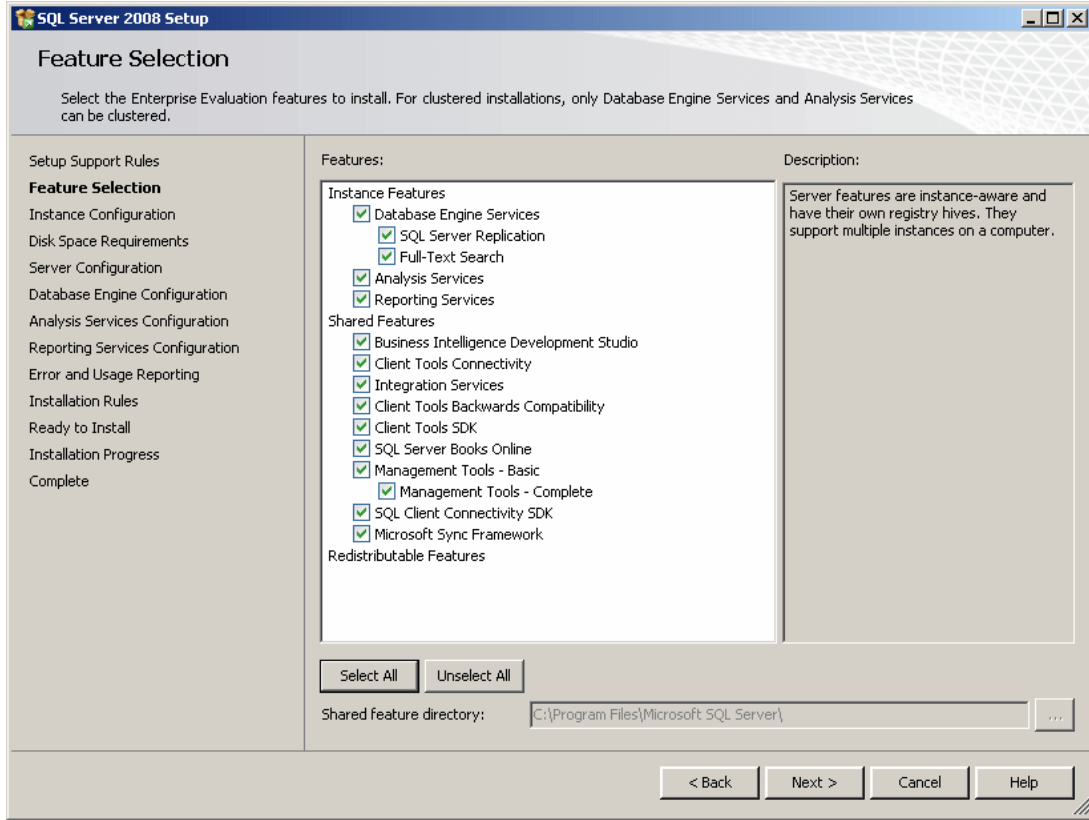
على الصفحة التالية، اقرأ الرسائل ثم انقر Install :



على الصفحة التالية، اقرأ الرسائل ثم انقر Install :



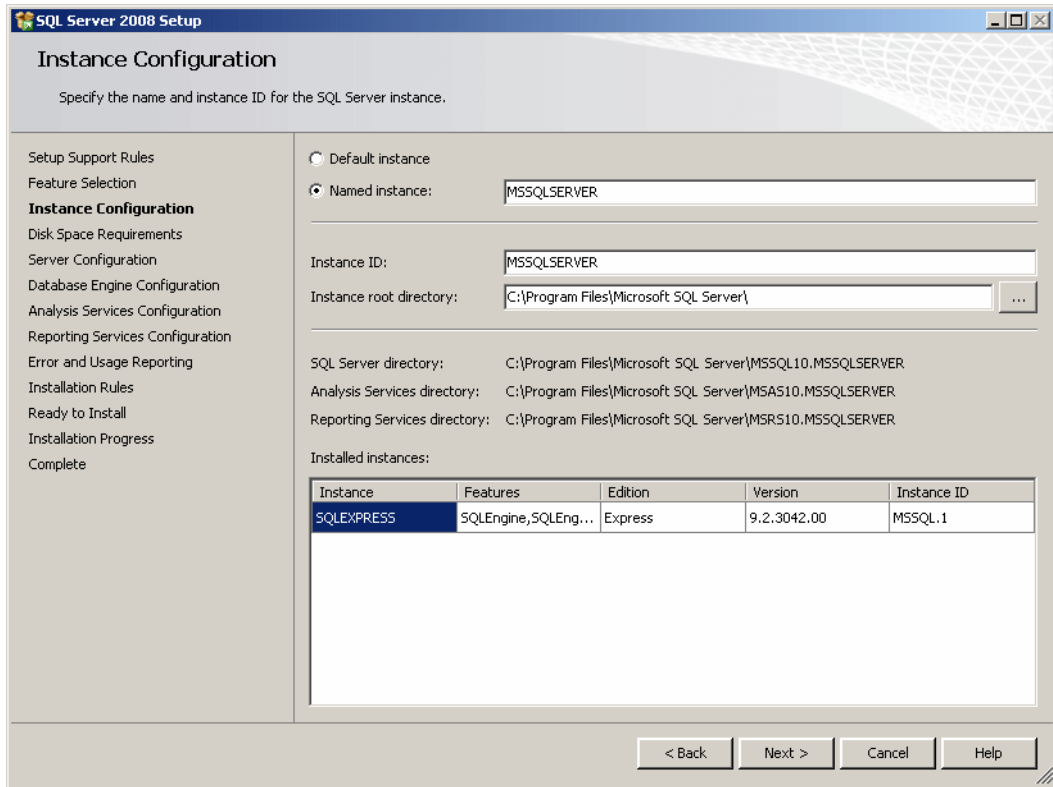
تسمح لك الصفحة التالية أن تحدد ما تريد، وحذف ما يمكن استبعاده، بعد اختياراتك انقر على التالي¹:



الصفحة التالية تسمح لك بتسمية النموذج (Instance²) وتحدد موضع تركيب الخادم، بعد اختياراتك انقر على التالي:

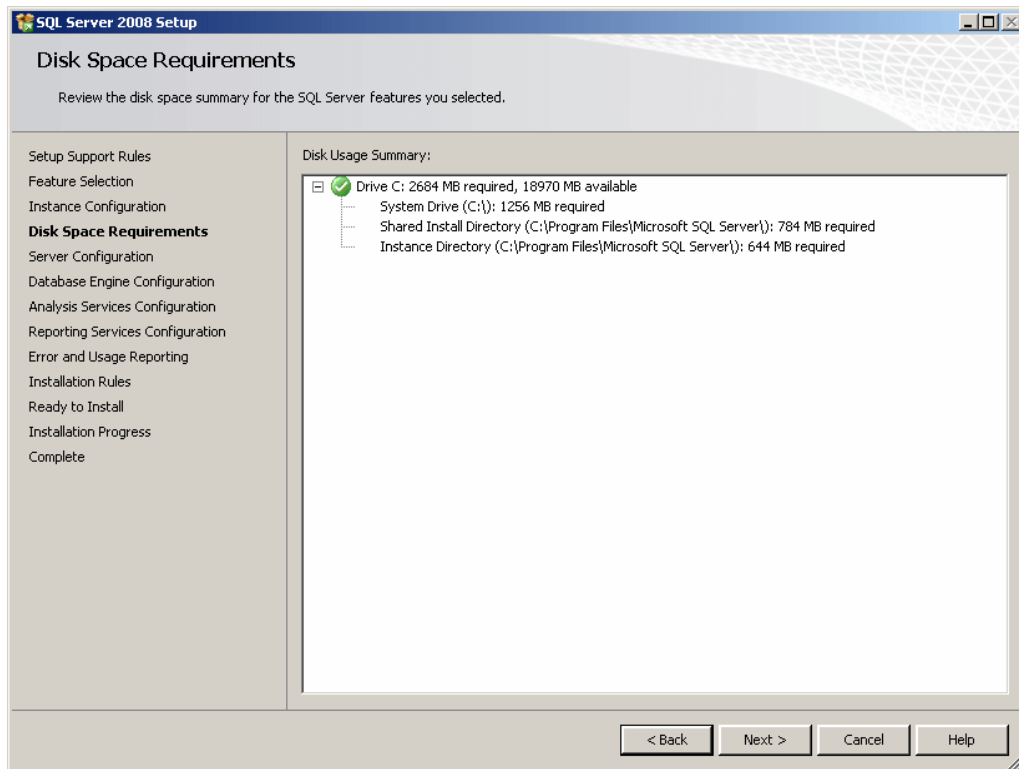
¹ تختلف الخيارات المتوفرة حسب نسخة MS SQL Server.

² يمكن تثبيت عدة نسخ (Instances) مستقلة من SQL Server على نفس النظام مع اختيار اسم مغاير لكل نسخة.

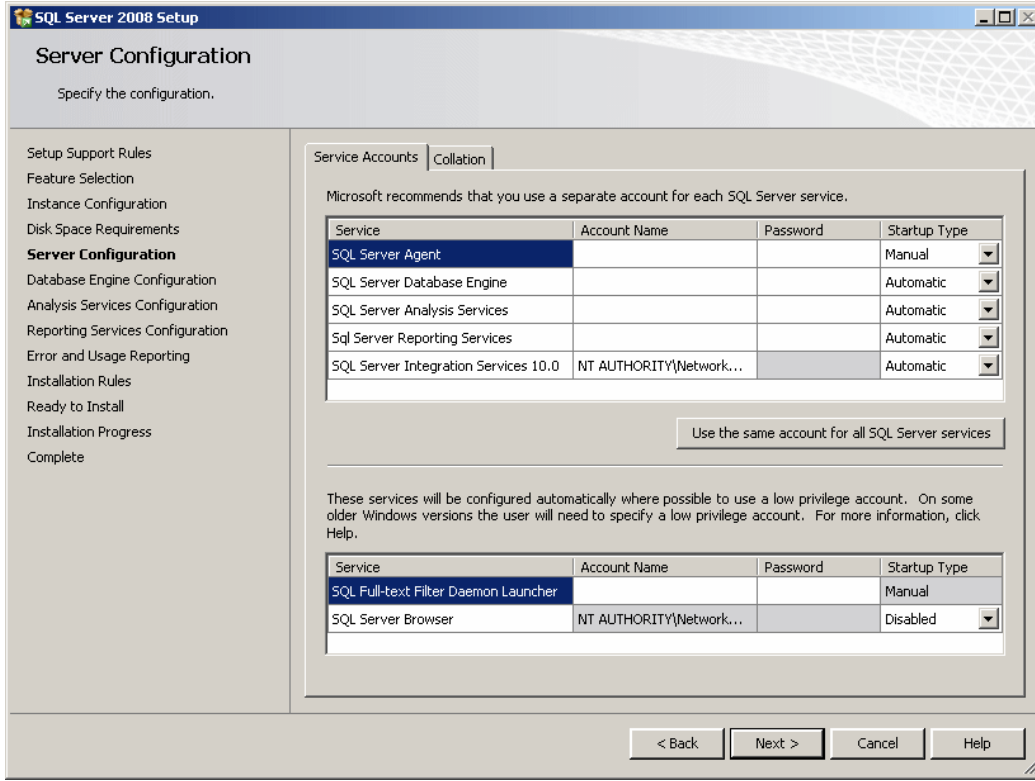


الصفحة التالية تتيح لك فحص والتأكد من مساحة القرص، بعد التحقق من ذلك انقر على

التالي



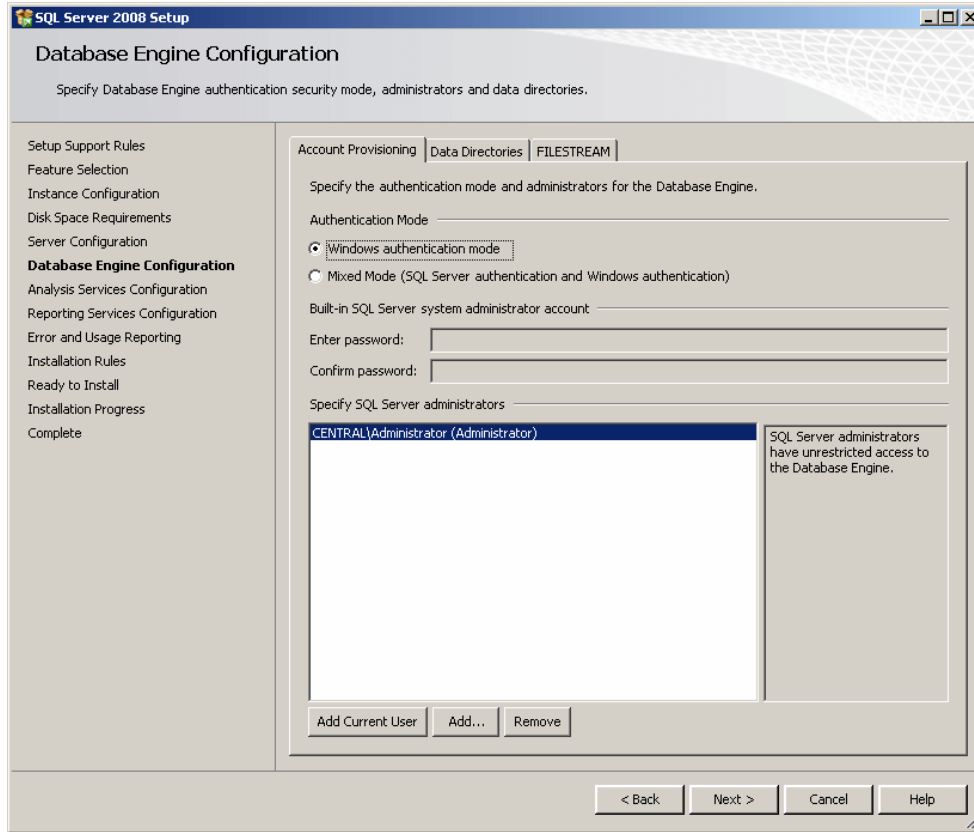
تسمح لك الصفحة التالية بإعداد الخادم، بعد إدخال خيار أنك انقر على التالي¹



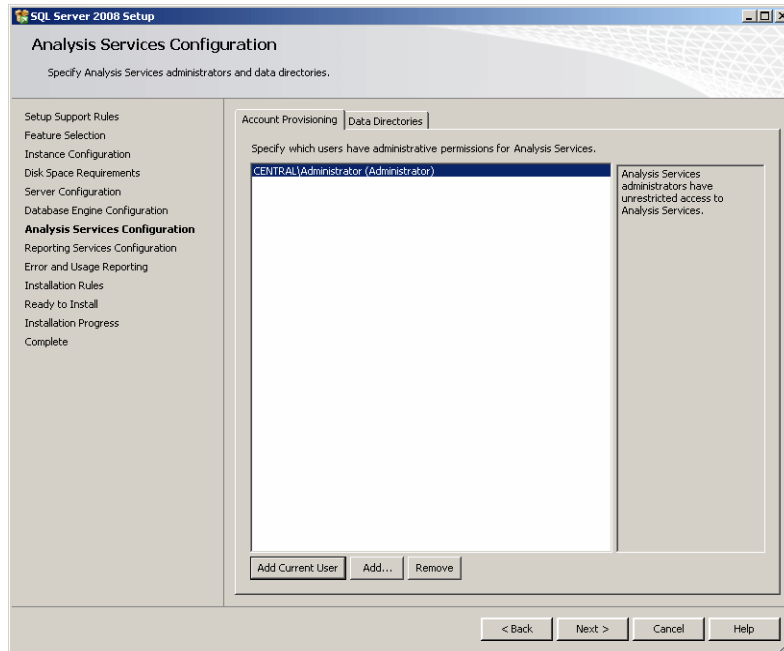
في الصفحة التالية، يتم تحديد طريقة التوثيق، ويجب أيضا أن تحدد استخدام حساب المدير، للقيام بذلك انقر على Add Current User. ثم انقر على التالي²

¹ هنا قم باختيار مستخدم Windows المخول له باستخدام SQL Server، أو انقر على Use the same account for all SQL Server services لاختيار نفس المستخدم لجميع حسابات تسجيل الدخول.

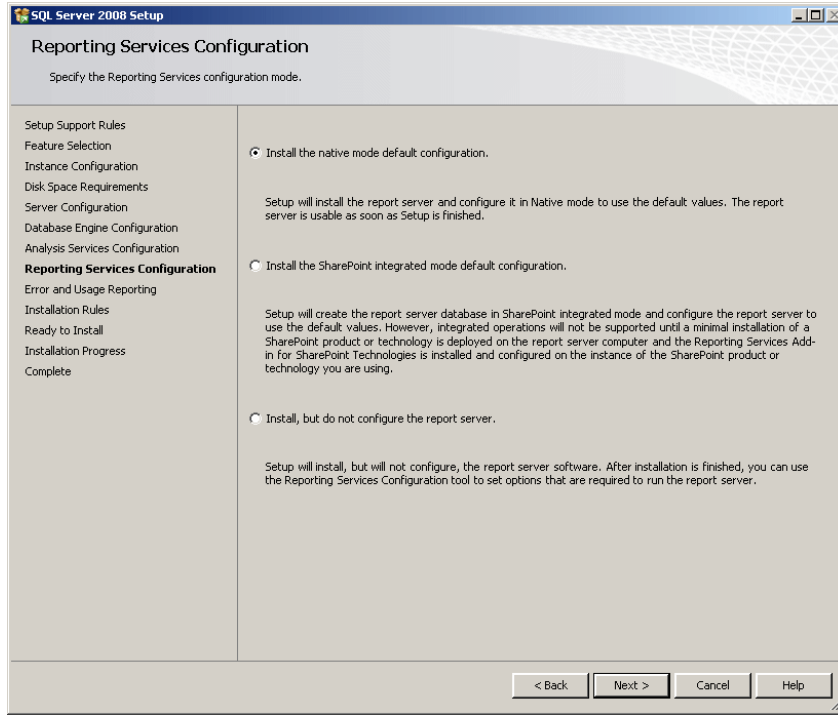
² هذه أهم مرحلة أثناء تنصيب الخادم، ولك أن تختار إما صلاحيات الدخول باستخدام حسابات Windows فقط من Windows authentication، أو باختيار Mixed Mode باستخدام حسابات التسجيل Windows وحسابات (إضافية) لتسجيل الدخول إلى SQL Server، الخيار الثاني يوفر حماية أكبر لكنه يكلف عبء برمجي إضافي، وباختياره تصبح غير قادر على الاتصال بقاعدة بيانات موجودة على الخادم إلا باسم تسجيل وكلمة مرور، بينما الخيار الأول يكفي للاتصال بقاعدة بيانات تسجيل دخولك إلى Windows ثم إرسال استعلام الاتصال فقط.



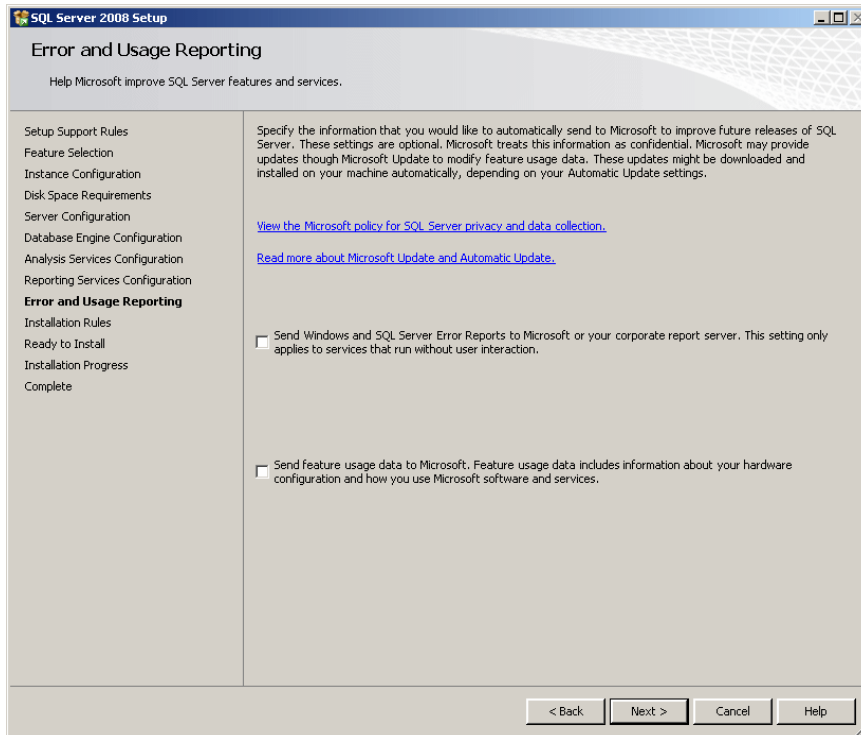
تسمح لك الخطوة التالية بتحديد حساب لتحليل الخدمات، يمكنك ببساطة النقر على أضيف المستخدم الحالي (Add Current User) وانقر على التالي



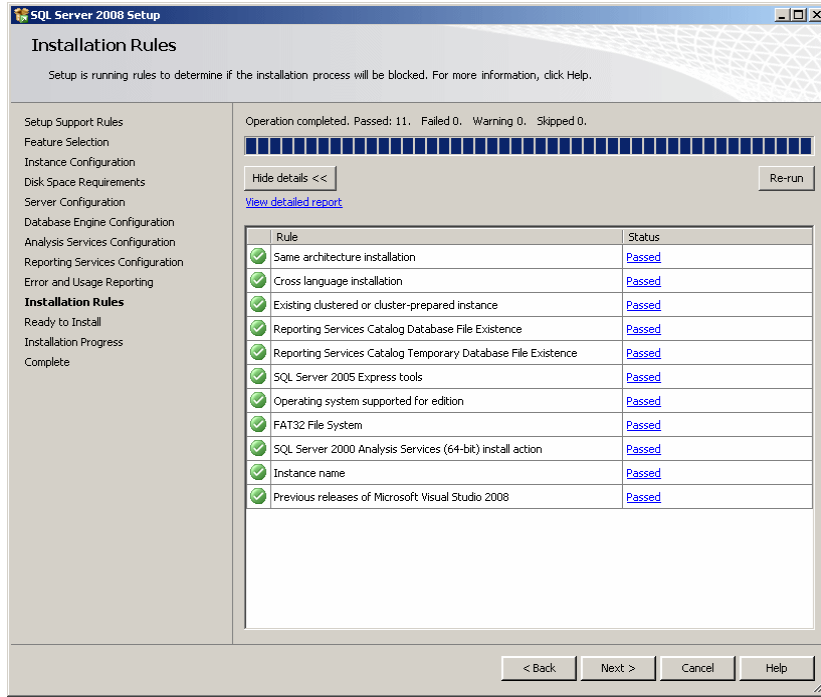
الصفحة التالية تسمح لك بتحديد خيار تركيب خدمة Reporting Services، اقبل الافتراضي أو غيره بما يناسب، ثم انقر على التالي



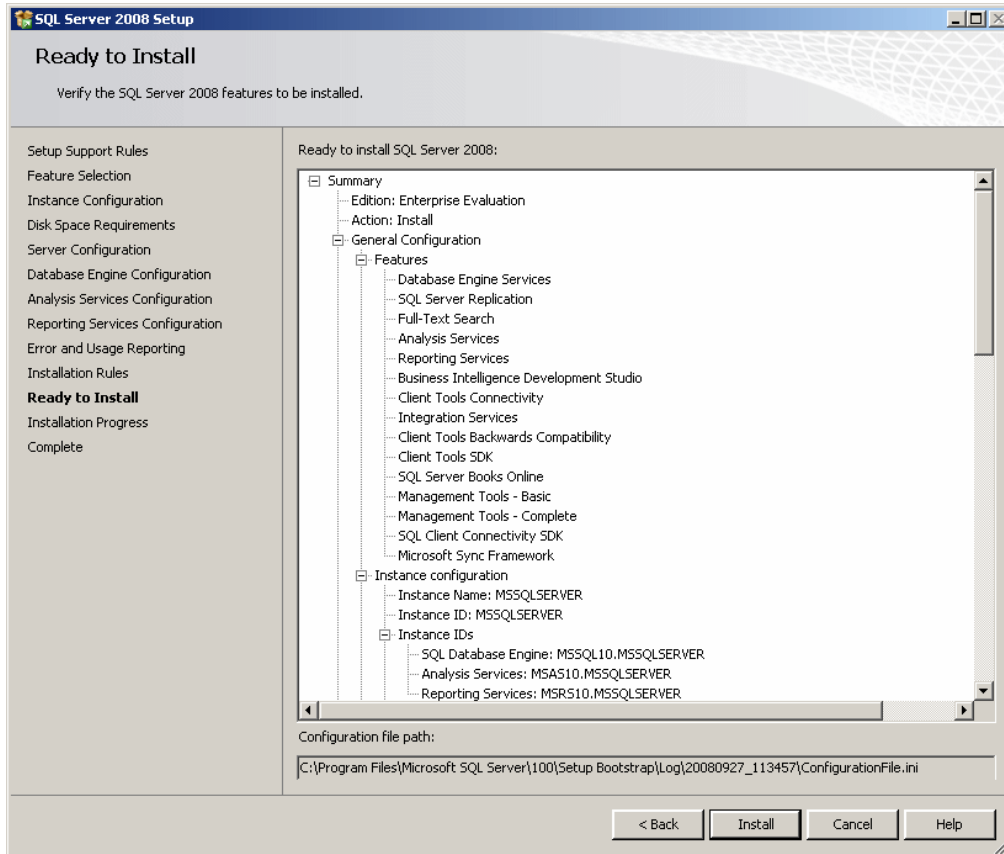
في الصفحة التالية يسمح لك أن تحدد فيما إذا كنت ترغب بإرسال التقارير التي يتعين جمعها وإرسالها إلى شركة Microsoft، بعد اختيارك انقر على التالي



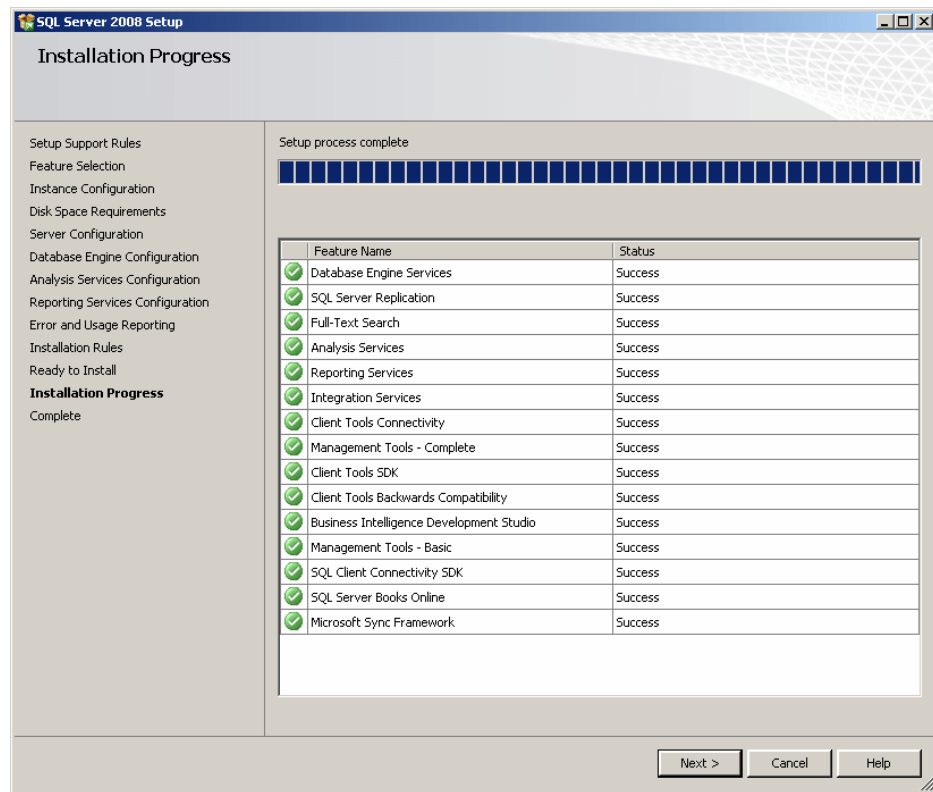
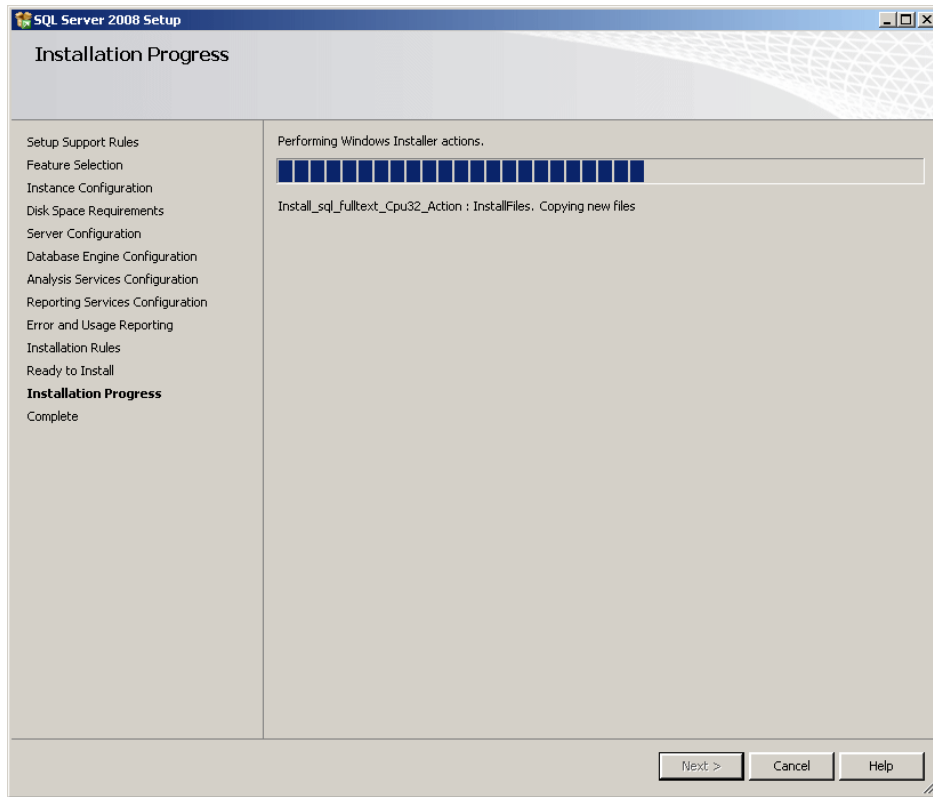
سيشرح المعالج في التركيب وسيعلمك عندما تنتهي من Installation Rules display:



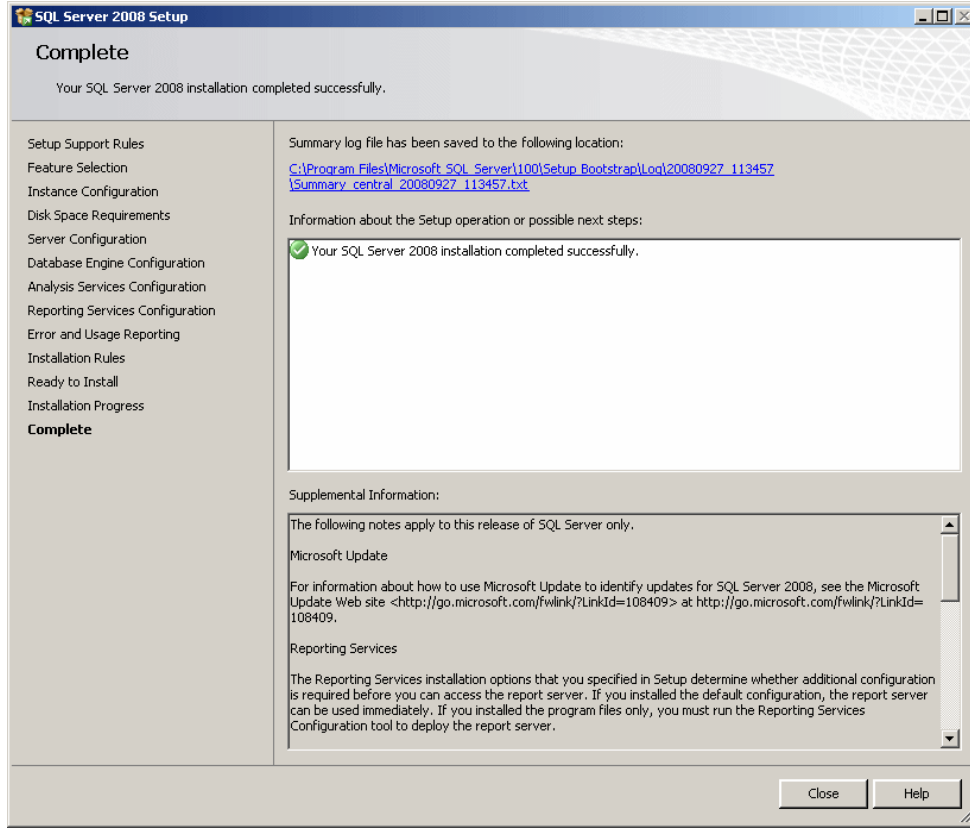
بعد القراءة... انقر على التالي، وهذا يشير إلى أن عملية التنصيب على وشك الانطلاق:



انقر على تثبيت لتبدأ عملية التركيب وتقدم شريط يشير إلى تطورها، وقد يستغرق هذا بعض الوقت...



عند انتهاء هذه المرحلة ستظهر رسالة لتعلمك بذلك وبمحل ملف تقرير تنصيب البرنامج، انقر على التالي، والصفحة الأخيرة من المعالج تخبرك بنهاية عملية التنبيت بالكامل.



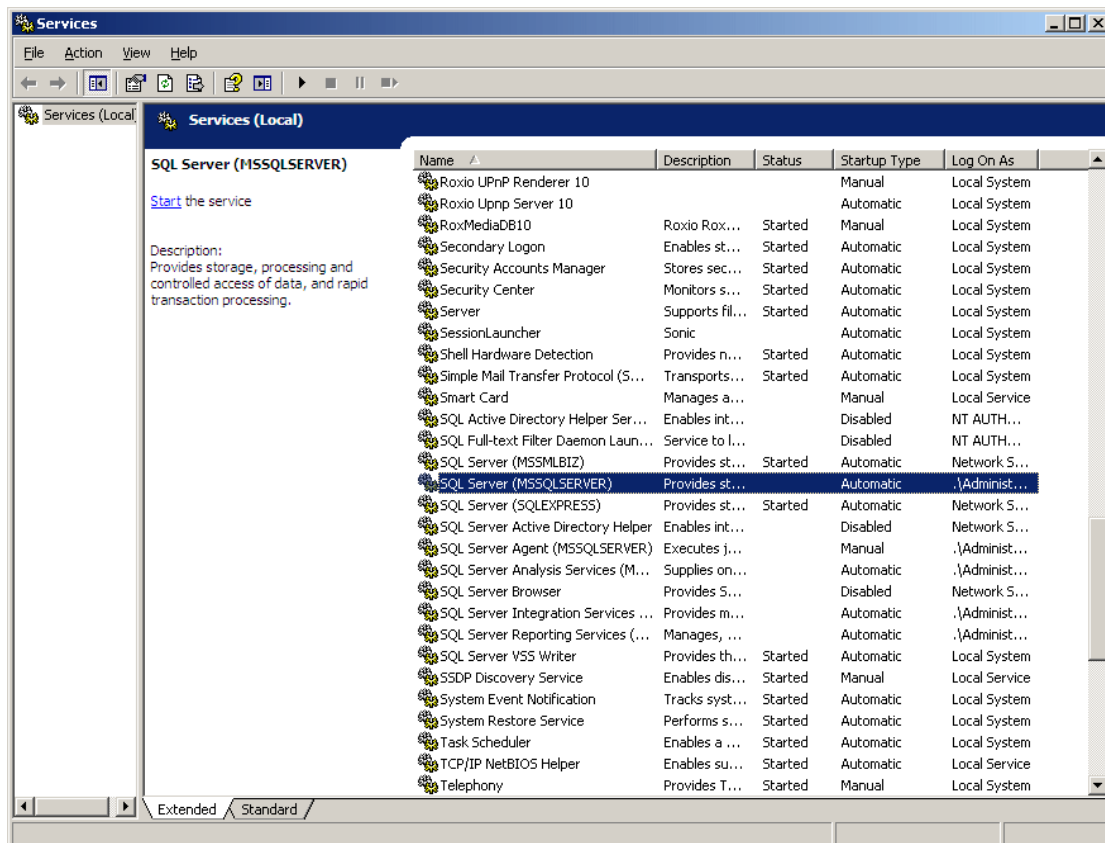
بعد قراءة الرسالة، يمكنك النقر على إغلاق.

البداية مع Microsoft SQL Server

خدمات MS SQL Server:

بعد تثبيت MS SQL Server، يمكنك الانطلاق في استخدامه .

لأن خادم مايكروسوفت SQL يعمل في شكل خدمات لنظام التشغيل، فإن لاستخدامها يجب عليك التأكد من تشغيل هذه الخدمات، للتأكد من تشغيلها (في Microsoft Windows XP Professional، Windows Vista، Windows Server 2003، Windows Server 3008) افتح Control Panel ثم Administrative Tools. وفي نافذة Administrative Tools افتح الخدمات، في نافذة الخدمات، تحقق من وضع خادم (SQL (MSSQLSERVER):

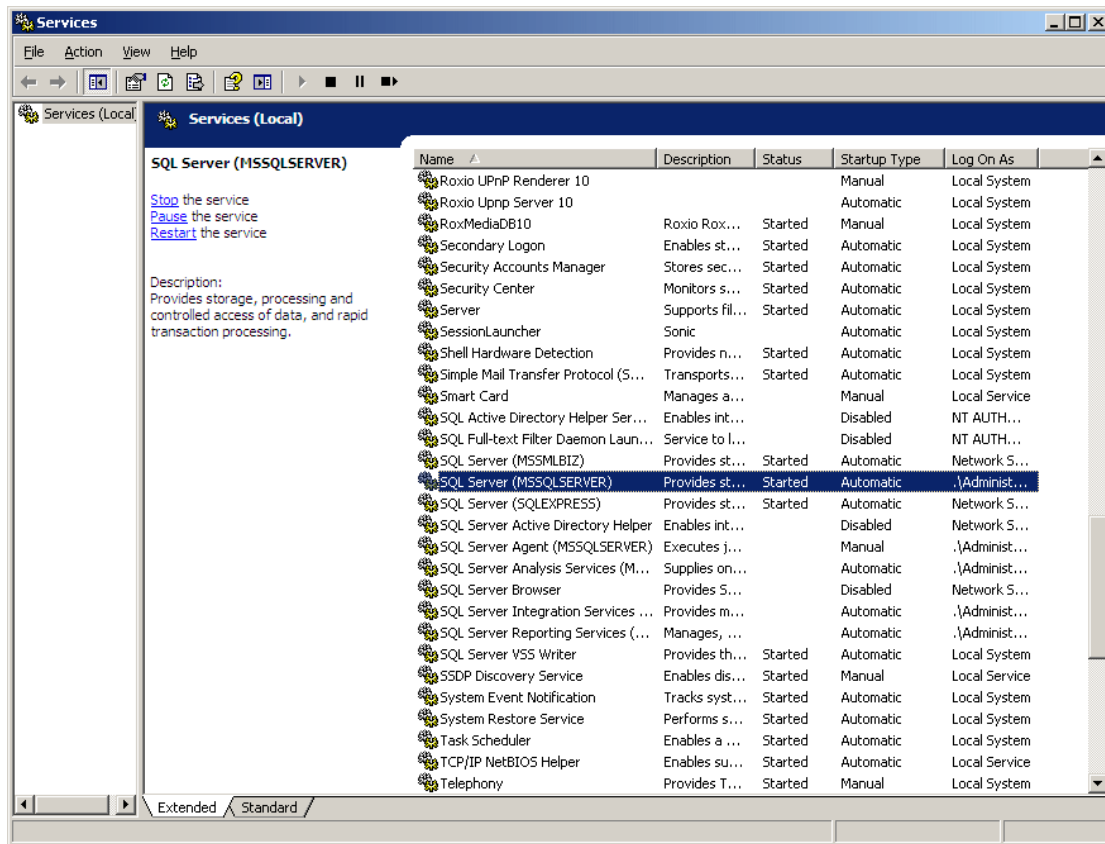


يجب تشغيل الخدمة MSSQLSERVER إذا كانت موقفة، للقيام بذلك يمكنك النقر بالزر الأيمن ثم انقر على Start، إذا لم ينجح ذلك تحقق من الحساب الذي سجلت به الدخول:

◀ إذا كنت تستخدم Microsoft Windows XP Professional، وسجلت الدخول بصفحتك مديرا ولم تحدد كلمة السر، فقم بفتح Control Panel ثم User Accounts، وافتح حساب المدير وأدخل كلمة السر.

◀ إذا كنت تستخدم Microsoft Windows Server 2003 أو Microsoft Windows Server 2008، تحقق من تسجيل دخولك بحساب يسمح لك بتشغيل الخدمات.

عند تشغيل الخدمة ستكون بهذا الشكل:

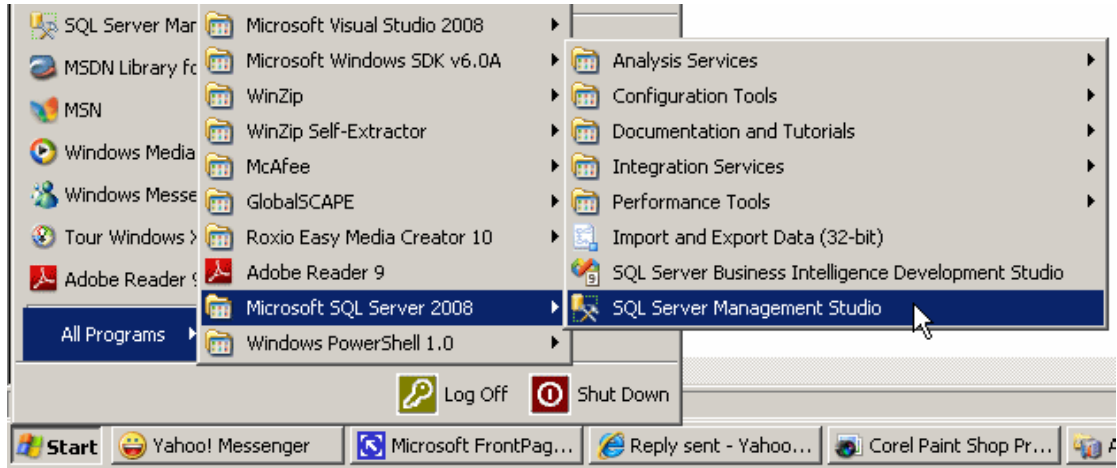


تشغيل¹ SQL Server Management Studio

للتشغيل Microsoft SQL Server، انقر على (All) Programs -> Start ->

SQL Server 2008 -> SQL Server Management Studio

¹ برنامج SQL Server Management Studio لإدارة محرك SQL Server؛ غير مضمن في نسخة Express Edition من Microsoft SQL Server، ويتوجب عليك تثبيت بشكل مستقل، متوفر للتحميل مجانا من الموقع الرسمي: [Microsoft® SQL Server®](http://Microsoft.com). هناك أيضا برنامج EMS SQL Manager for SQL Server وله نسخة مجانية من هنا: [EMS SQL Manager for SQL Server Freeware](http://EMS.com).



عند تشغيله سيواجهك مربع الحوار لتسجيل الدخول.

تطبيق عملي: تشغيل SQL Server Management Studio

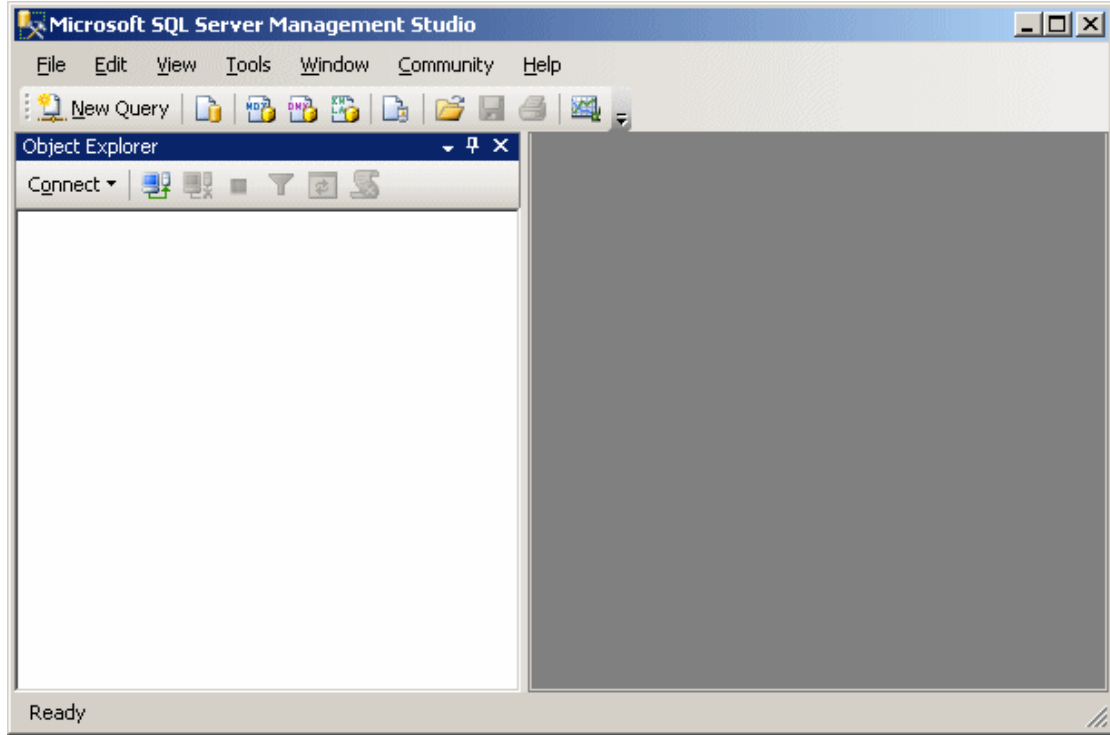
قم بتشغيل: Start -> (All) Programs -> Microsoft SQL Server 2008 -> SQL Server Management Studio



على مربع الحوار للاتصال بالخادم، انقر على إلغاء.

برنامج Microsoft SQL Server Management Studio

هناك العديد من الوسائل التي يمكن استخدامها في Microsoft SQL Server، من بينها Microsoft SQL Server Management Studio، وهو يستخدم كمدير للتخاطب مع الخادم، بعد غلق مربع حوار الاتصال ستواجه الواجهة التالية:



الجزء الأعلى من SQL Server Management Studio يعرض شريط العنوان الكلاسيكي للنافذة الرئيسية مع أيقونة على اليسار، ثم عنوان البرنامج وأزرار النظام على الجانب الأيمن.

القائمة الرئيسية (Main Menu)

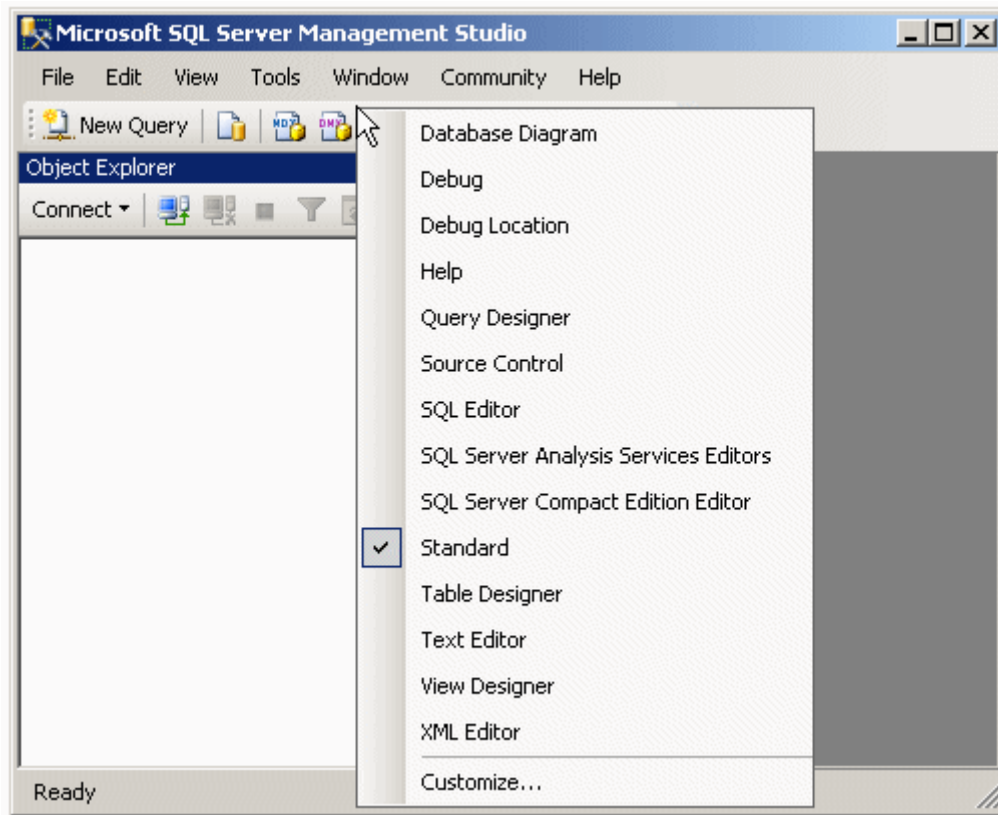
تحت شريط العنوان، تجد شريط القوائم وبه مجموعة من القوائم التي تستخدم لأداء مختلف العمليات الأساسية اللازمة.

شريط الأدوات الأساسي (Standard Toolbar)

يعرض شريط الأدوات الأساسي تحت القائمة الرئيسية:



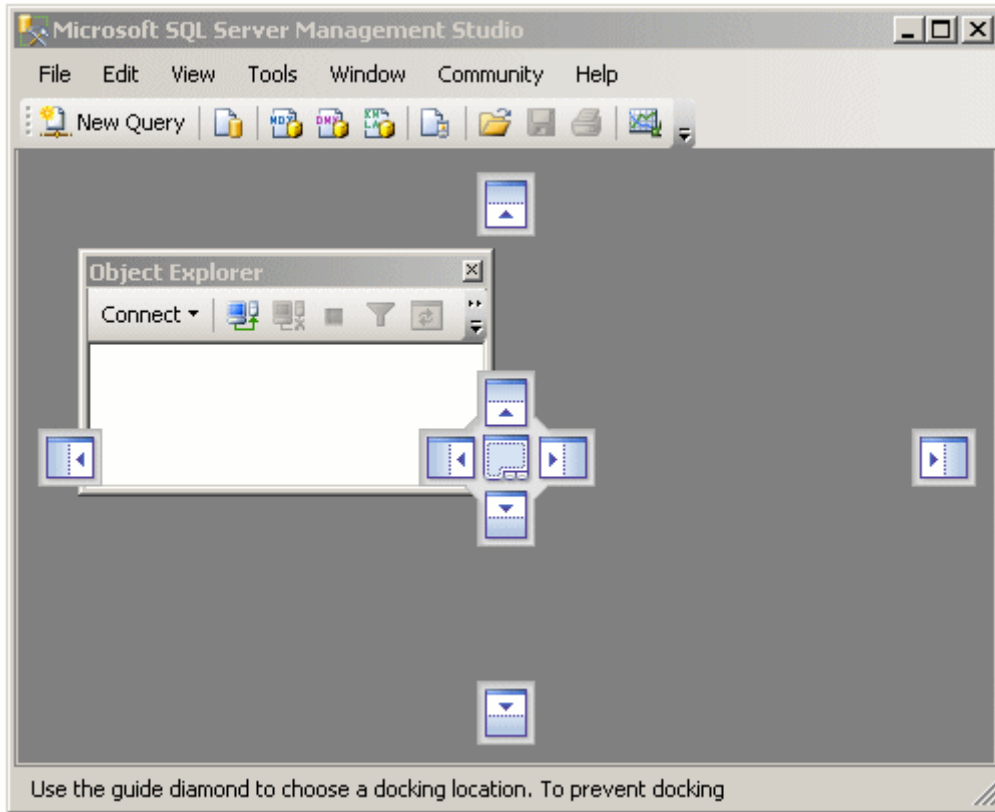
هناك العديد من الأدوات فإذا كنت تريد أن تظهر أي شريط الأدوات، يمكنك النقر على أي شريط أدوات بالزر الأيمن لعرض أشرطة الأدوات المتاحة، وهناك تظهر القائمة التالية:



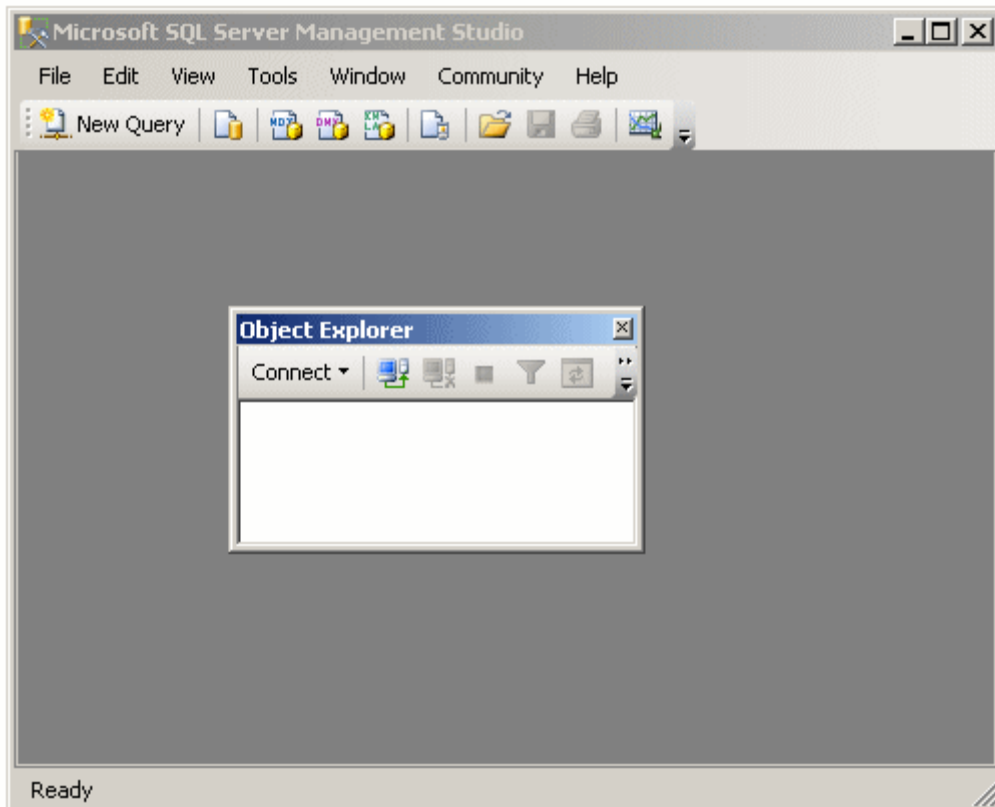
متصفح الكائنات (Object Explorer)¹:

في الجانب الأيسر من الواجهة الرئيسية نجد نافذة Object Explorer، وهذه النافذة تقبل الالتحام (Dockable)، أي أنه يمكن نقلها من جانب لآخر لتلتحم مع نافذة الواجهة الرئيسية، للقيام بذلك يمكنك الضغط بالزر الأيسر على عنوان الشريط إلى أي موقع تختاره، وعندها ستلاحظ أسهم في مربعات صغيرة تأتي بهذا الشكل:

¹ Object Explorer نافذة تعرض فيها كل الكائنات في عرض شجري (TreeView) بما في ذلك: قواعد البيانات، الجداول، الحقول، القوالب، المستخدمين، القيود... الخ.



يمكنك سحب وإفلات النافذة في أي من تلك المواضع، كما يمكنك وضعها في أي مكان ما في منتصف الواجهة:

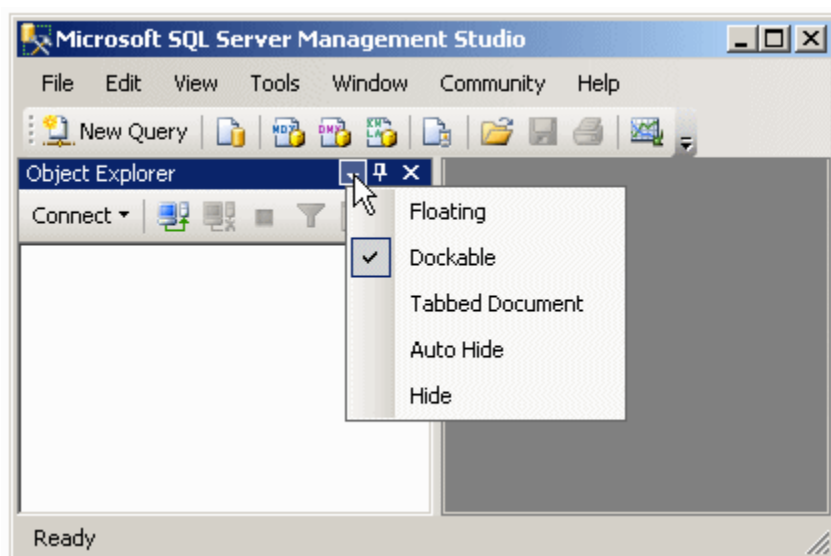


لوضع النافذة في وضعها السابق، قم بنقر مزدوج فوق شريط عنوانها، يمكن أيضا جعلها على شكل تبويب رأسيا أو أفقيا.

يمكنك في أي وقت غلق Object Explorer أو إخفاؤه بالنقر على زر الإغلاق.

على الجانب الأيمن من متصفح الكائنات تجد ثلاثة أزرار، إذا ضغطت على أولها تظهر

قائمة شكلها:




تتيح القائمة تحديد وضعية شكل النافذة Floating, Dockable, Tabbed.

في الجانب الأيمن من الواجهة نافذة فارغة، تستخدم هذه المساحة لعرض محتويات ما هو محدد في متصفح الكائنات، أو أن لعرض النتائج لبعض العمليات، يمكن في هذه المساحة عرض العديد من النوافذ على شكل تبويب.

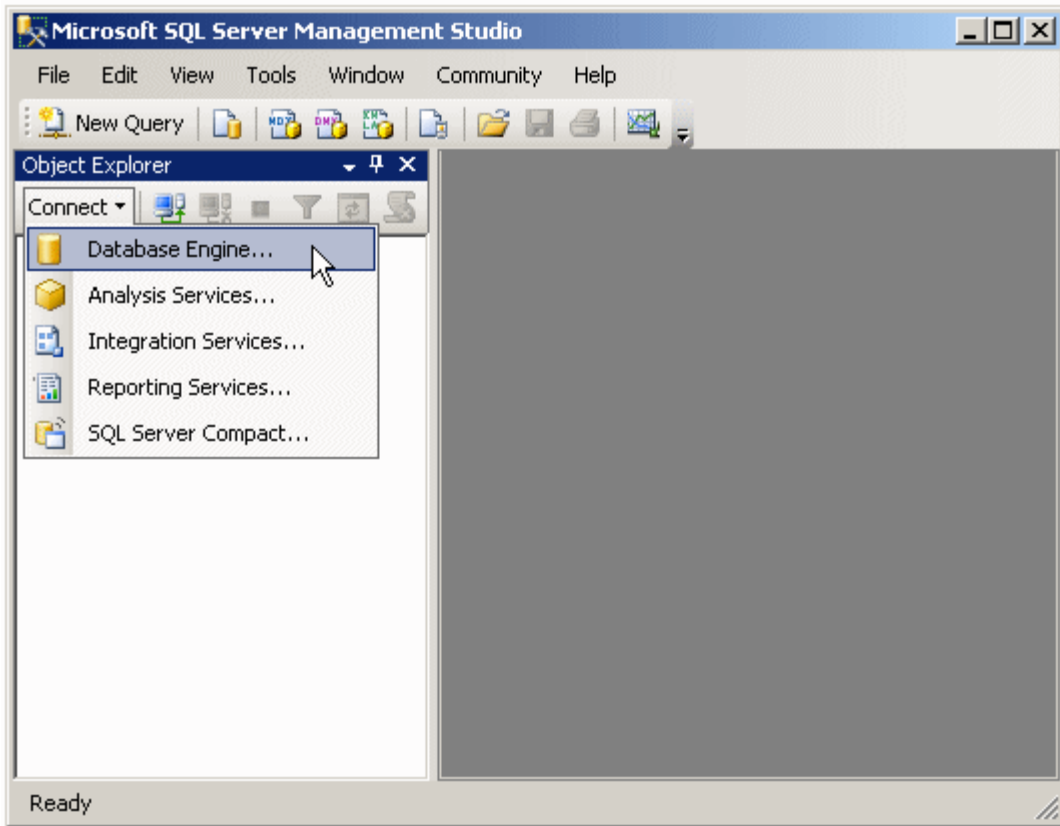
الاتصال بخادم قواعد البيانات Microsoft SQL Server:

للقيام بأي شيء في Microsoft SQL Server يجب التسجيل للدخول إلى الخادم، إذا قمت بتشغيل Microsoft SQL Server Management Studio من زر ابدأ سيواجهك نافذة الاتصال، للاتصال بالخادم (إذا كنت قد أغلقت نافذة الاتصال) قم:

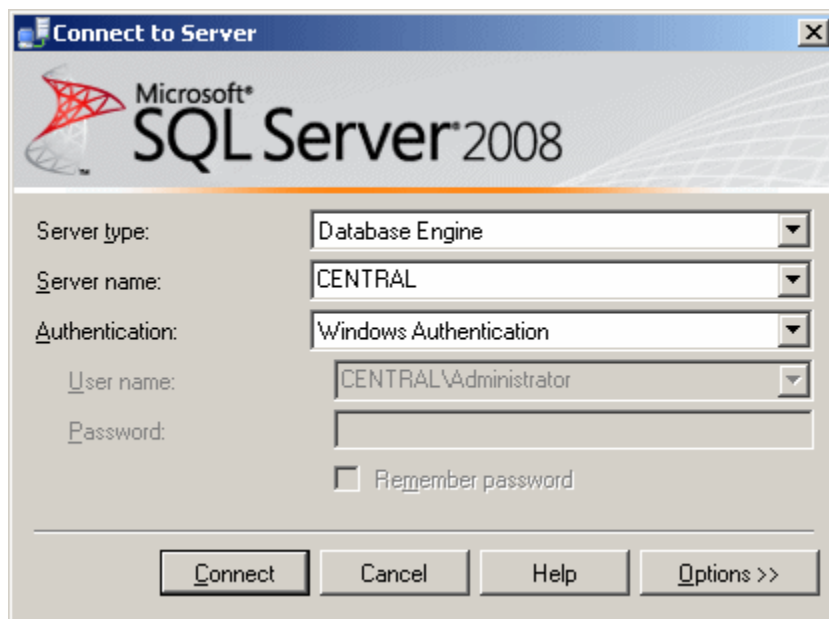
← من القائمة الرئيسية باختيار File ثم Connect Object Explorer.

← من شريط الأدوات الرئيسي بالنقر على زر "استعلام جديد" .

من متصفح الكائنات يمكنك النقر على سهم الزر Connect لتختار أحد الخيارات، مثل محرك قاعدة بيانات (Database Engine)



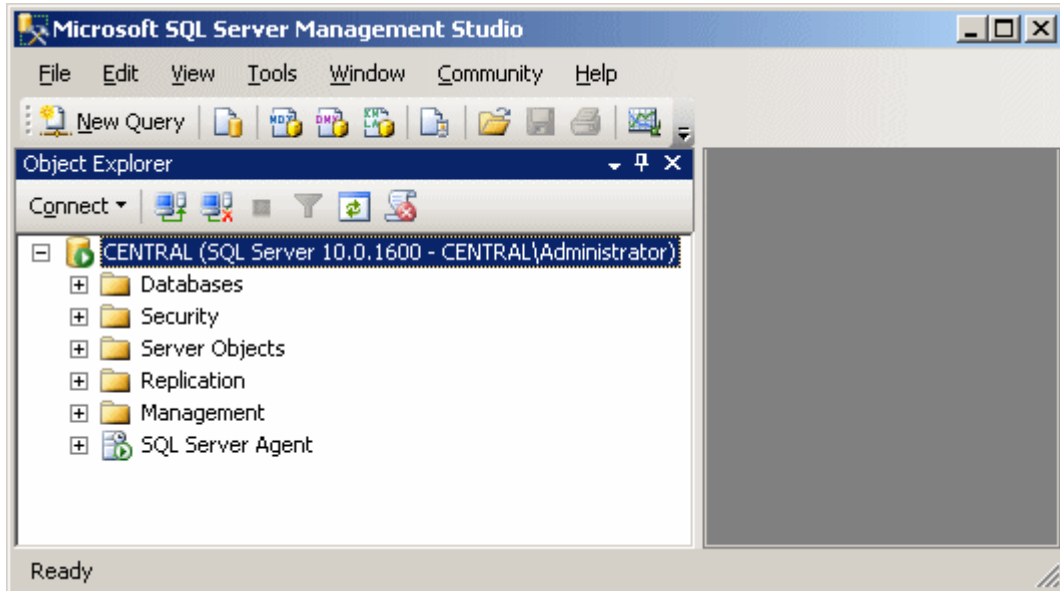
كل هذه الخيارات تعرض مربع الحوار للاتصال بالخادم:



من هنا يمكنك اختيار اسم الخادم ونوع التوثيق المستخدم، إذا اخترت توثيق Windows فقط انقر على Connect، أو في خانة الاختيارات Authentication يمكنك اختيار SQL Server Authentication، وفي هذه الحالة يجب إدخال اسم مستخدم وكلمة سر:



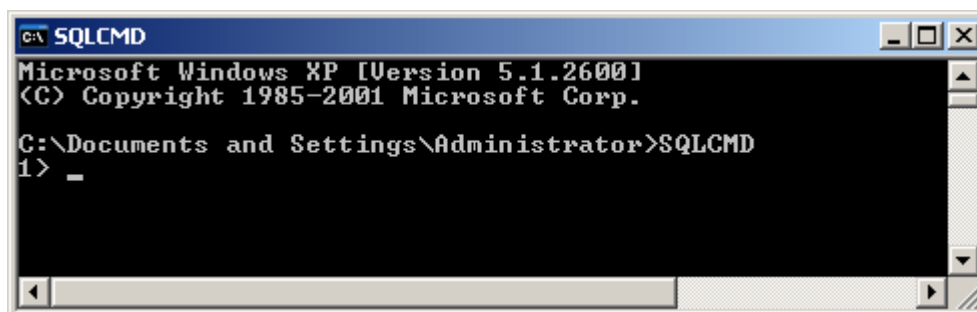
بعد ذلك انقر على Connect للاتصال، إذا قمت بتسجيل الدخول بنجاح، سيبدو لك متصفح الكائنات مع بعض العناصر الأخرى:



بعد استخدام Microsoft SQL Server Management Studio، يمكنك إغلاقه كأى برنامج.

باستخدام سطر الأوامر:

إلى جانب SQL Server Management Studio، يمكنك أيضا العمل على Microsoft SQL Server من سطر الأوامر، ويتم ذلك باستخدام برنامج أو تعليمات SQLCMD.EXE، لاستخدامه افتح موجه الأوامر اكتب SQLCMD واضغط على Enter.



```
c:\ SQLCMD
Microsoft Windows XP [Version 5.1.2600]
Copyright (C) 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator>SQLCMD
1> _
```

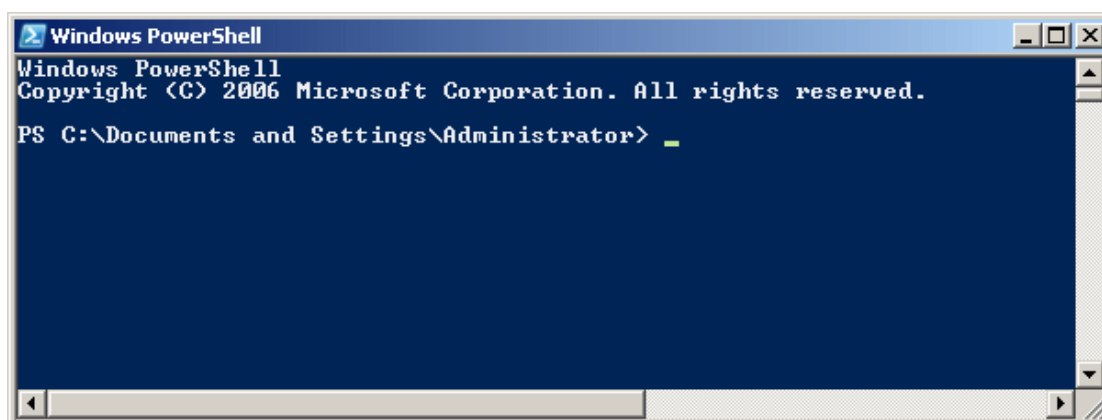
للخروج أدخل Quit واضغط Enter.

باستخدام Windows PowerShell:

عندما يتم تركيب Microsoft SQL Server 2008، يتم معه تركيب PowerShell 1.0، وهو برنامج تعليمات من شركة Microsoft، إلى جانب SQL Server Management Studio و سطر الأوامر، يمكنك استخدام PowerShell لإنشاء وإدارة قواعد البيانات، لتشغيله:

Start -> (All) Programs -> Windows PowerShell 1.0 -> Windows PowerShell

ستعرض لك نافذة DOS عنوانها Windows PowerShell:



```
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.
PS C:\Documents and Settings\Administrator> _
```

للوصل إلى خادم Microsoft SQL Server من PowerShell، اكتب SQLCMD واضغط

:Enter

```
SQLCMD
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Documents and Settings\Administrator> SQLCMD
1> _
```

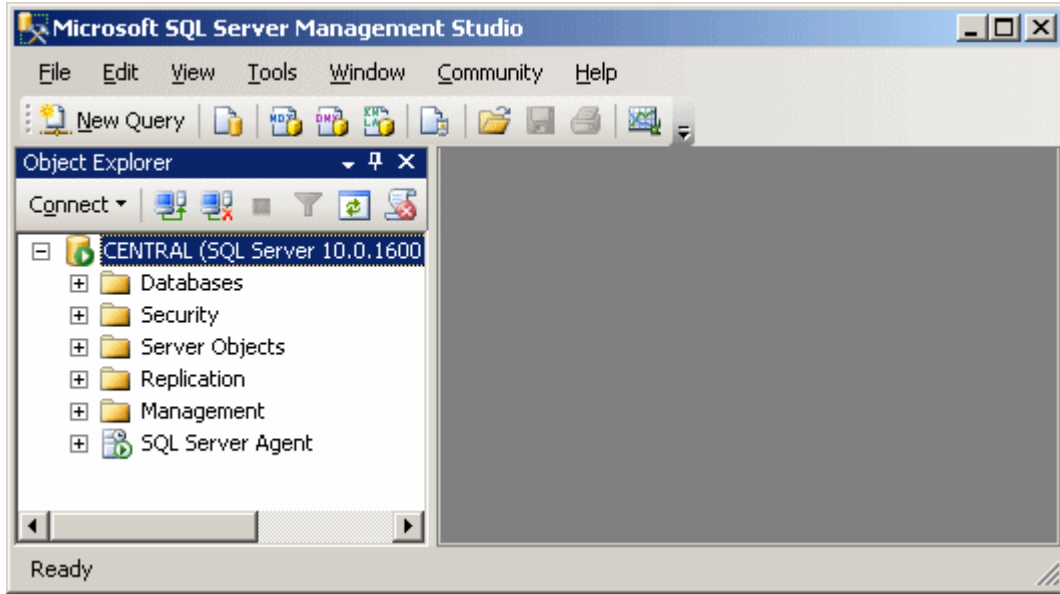
لاحظ ما كتب في شريط العنوان: SQLCMD، مما يشير إلى أن البرنامج على استعداد لتلقي الأوامر والاتصال مع Microsoft SQL Server، للخروج من PowerShell، اكتب Quit واضغط Enter:

```
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

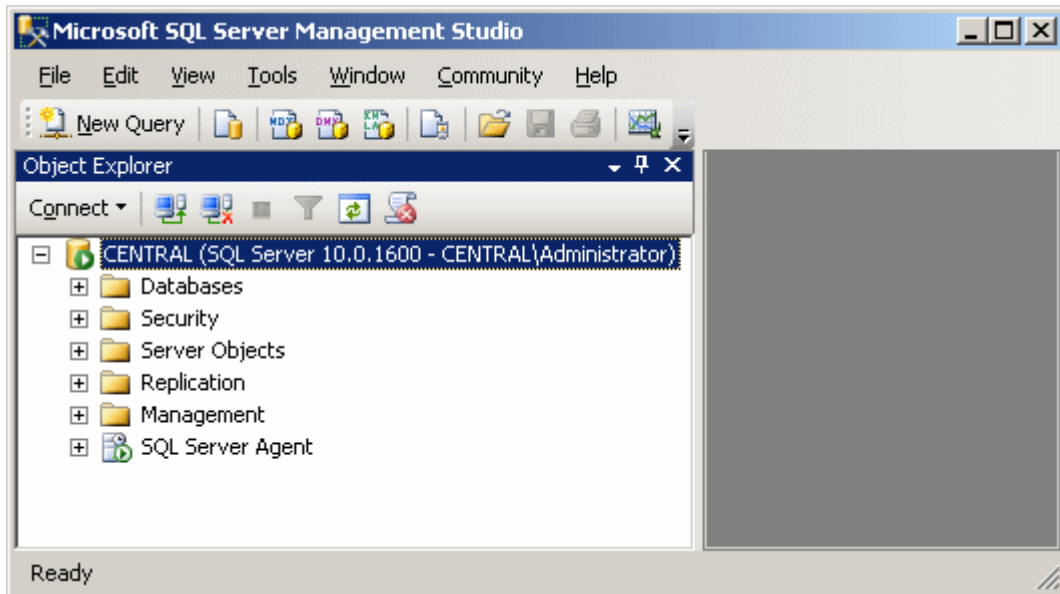
PS C:\Documents and Settings\Administrator> SQLCMD
1> Quit
PS C:\Documents and Settings\Administrator> _
```

استخدام Microsoft SQL Server Management Studio:

يعرض متصفح الكائنات قائمة من البنود في شكل شجرة، أحد أكثر البنود التي يتعامل معها هي اسم الخادم الذي تستخدمه، إذا كنت تود الاطلاع فقط على تطوير قواعد البيانات أو أنك مبتدئ في تطوير قواعد البيانات، يمكنك استخدام أو التعامل مع خادم واحد فقط، قد تحتاج في بعض الحالات إلى التعامل مع أكثر خادم، أيا كان؛ يجب أن تكون دائماً على علم بالخادم الذي تتصل به حالياً، وهذا سهل للتفحص؛ فقط لاحظ أول بند في متصفح الكائنات، مثلاً في ما يلي اسم الخادم Central:

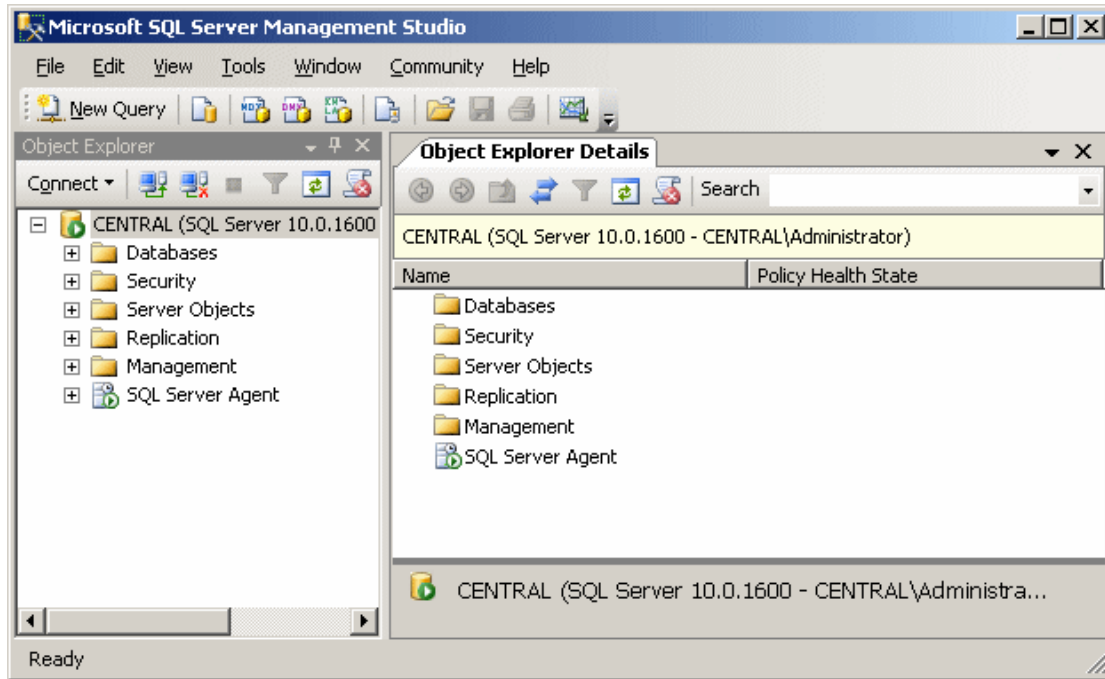


رأينا فيما سبق أنه من أجل إنشاء اتصال بالخادم يجب أن تملك حساب تسجيل الدخول، أحيانا يمكنك استخدام نفس الحساب مرارا وتكرارا، في حالات أخرى قد تكون لديك حسابات مختلفة تستخدمها لسيناريوهات مختلفة، مثل حساب لتطوير قواعد البيانات، وحساب آخر لإدارة قواعد البيانات، وثالث لاختبار لقاعدة بيانات. بعض الحسابات لا يمكنها أن تقوم ببعض العمليات، وعند أداء بعض العمليات يجب عليك دائما معرفة الحساب المستخدم، ويمكنك التحقق من ذلك في القوسين بعد اسم الخادم، في ما يلي حساب يدعى Administrator مسجل للدخول إلى خادم اسمه Central:

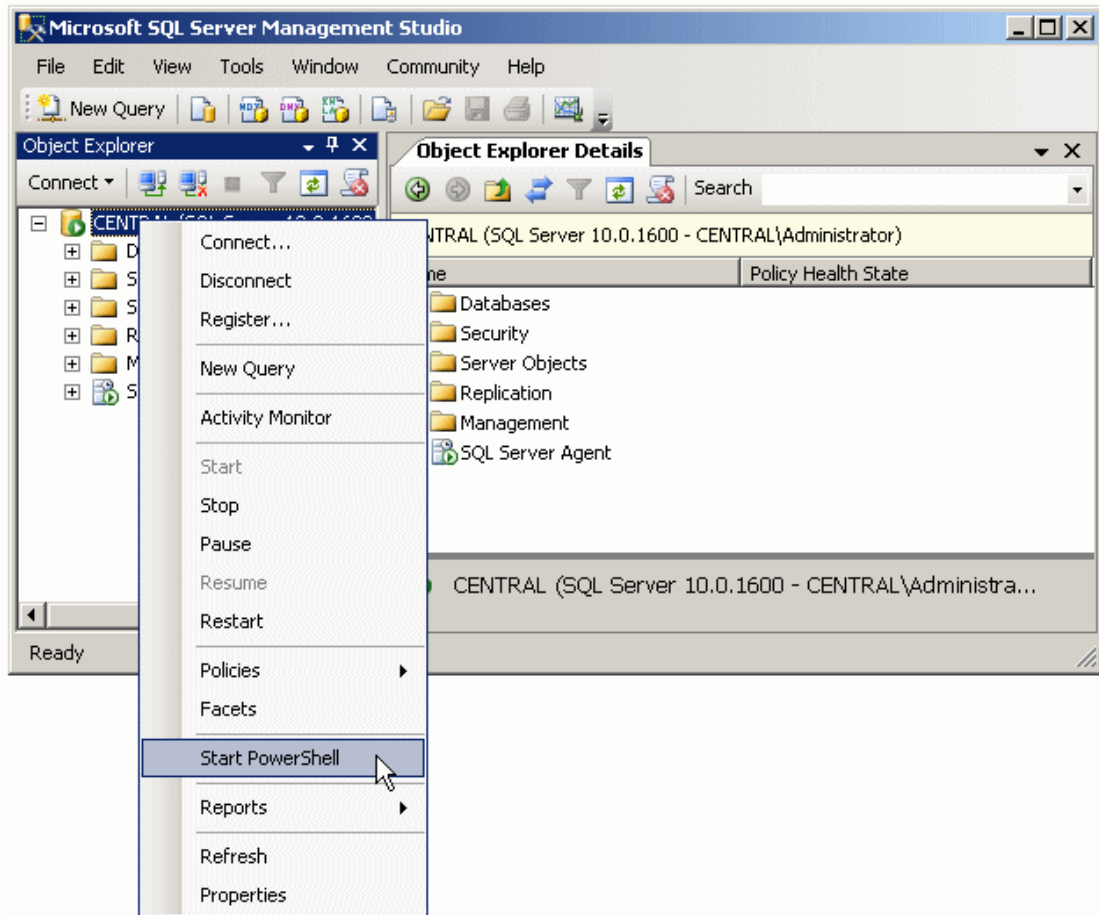


تفاصيل متصفح الكائنات (Object Explorer Details)

عندما تحدد عنصرا في متصفح الكائنات تظهر في الجهة اليمنى من الواجهة معلومات أكثر تفصيلا عن هذا البند، للقيام بذلك من القائمة الرئيسية انقر على Object -> View Explorer Details، وستشاهد على الجانب الأيمن المعلومات المتعلقة بالبند المحدد:



أكثر العناصر أهمية هي Databases، يحمل هذا البند أسماء قواعد البيانات على الخادم المتصل به، ومن خلاله يمكنك القيام -تقريبا- بكل ما يلزم لإنشاء وإدارة قاعدة بيانات، للاطلاع على الإمكانيات المتاحة يمكنك فتح البند Databases وبعض عناصره الفرعية، أو يمكنك النقر بالزر الأيمن على أي من قواعد البيانات أو أحد عناصرها الفرعية لاستعراض قائمة بمختلف العمليات، مثلا لتشغيل PowerShell يمكنك النقر بالزر الأيمن على البند قواعد البيانات (Databases) أو على اسم الخادم ثم اختيار Start PowerShell:





عندما يشتغل PowerShell يعرض ما تم النقر عليه.

مدخل إلى محرر الاستعلام:

يمكنك محرر الاستعلام من إجراء العديد من العمليات على قاعدة البيانات، حيث أن بعض العمليات تتطلب منك كتابة الاستعلام، للقيام بهذا يوفر Microsoft SQL Server Management Studio محرر الاستعلامات وقوالب جاهزة مختلفة (Code templates).

فتح محرر الاستعلام:

- ◀ من القائمة الرئيسية يمكنك النقر على Query -> New -> File بالاتصال الحالي.
- ◀ على شريط الأدوات الأساسي انقر على زر استعلام جديد 
- ◀ في متصفح الكائنات انقر بالزر الأيمن على اسم الخادم وانقر على استعلام جديد.

وهذا من شأنه إنشاء استعلام جديد على الجانب الأيمن من الواجهة، يمكنك حفظ ملف SQL بمحتوى محرر الاستعلام بالضغط على الزر  أو مفتاح Ctrl + S أو من القائمة الرئيسية.

لغة الاستعلامات البنوية The Structured Query Language

بعد إنشاء الاتصال بالخادم يمكنك إجراء العمليات على قاعدة البيانات.

لغة الاستفسارات البنوية المعروفة باسم SQL، هي لغة عالمية تستخدم في مختلف النظم الحاسوبية لإنشاء وإدارة قواعد البيانات.

كغيرها من اللغات مثل C/C++، Pascal، أو Java، يمكن تطبيق SQL على مختلف نظم قواعد البيانات، لتكييف SQL مع Microsoft SQL Server وضعت الشركة Transact-SQL كاعتماد للغة SQL. Transact-SQL هي اللغة المستخدمة بشكل داخلي في Microsoft SQL Server وMSDE¹، أيضا SQL Server يدعم بشكل كبير تعليمات SQL القياسية، فقط بعض التفاصيل قد لا تنطبق مع غيره من أنظمة قواعد البيانات مثل MySQL، Oracle، و MS Access، على الرغم من أنها تتفق مع المعايير.


ما سنتعامل معه هنا هو Transact-SQL، أي أننا نفترض أنك تستخدم Microsoft SQL Server كمدير لقواعد البيانات، وهذا يعني أنه ما لم ينص على خلافه فإنه في هذه الدروس ترجع استعلامات SQL إلى Transact-SQL الخاص بـ Microsoft SQL Server.

مترجم SQL:

باعتبار SQL لغة حاسوب فهي تستخدم لإرسال تعليمات داخلية إلى برنامج يدعى مترجم (Interpreter)، وكما سيأتي إن شاء الله يجب التأكد من دقة تعليمات SQL، لغة SQL ليست حساسة لحالة الحروف. وهذا يعني أن CREATE و create و Create تعني الشيء نفسه. و جرت العادة في كتابة عبارات SQL الخاصة بالأحرف الكبيرة، وهذا يساعد على التمييز بين تعليمات SQL وأسماء كائنات من قاعدة البيانات.

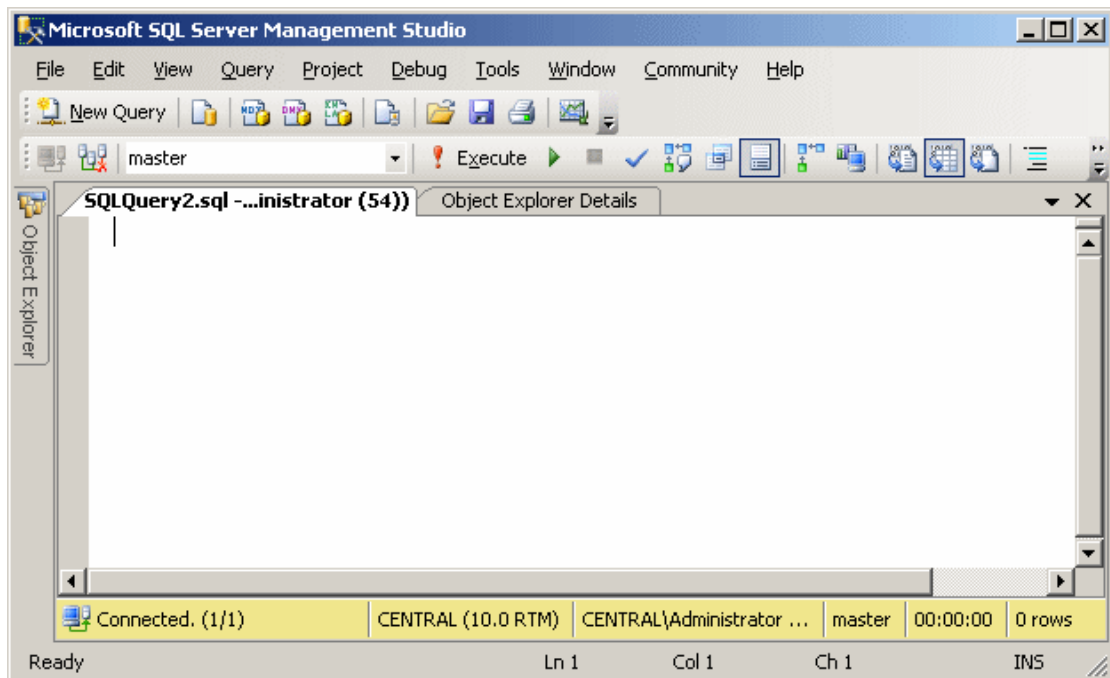
¹ MSDE أو Microsoft SQL Server Desktop Engine هي نسخة محدودة ومجانية من Microsoft SQL Server.

سنتعامل كثيرا مع SQL في هذه السلسلة من خلال كتابة الاستعلامات من خلال نافذة يوفرها Microsoft SQL Server لكتابة الاستعلامات. من الجانب الأيسر للواجهة الرئيسية انقر بالزر الأيمن على اسم الخادم وانقر على New Query، بنفس الطريقة يمكنك فتح عدة نوافذ استعلامات دفعة واحدة.

عند فتح محرر الاستعلامات يمكنك كتابة الكود ثم حفظه في ملف بامتداد sql، ويمكنك فتح ملف استعلام (أو أكثر) في محرر الاستعلامات في شكل تبويب، لغلق أحد هذه النوافذ حددها ثم انقر على الزر الإغلاق ، وسيطلب منك حينئذ حفظ الاستعلام، وسنتعلم كيفية كتابة الاستعلام وتنفيذه.

تنفيذ الاستعلام:

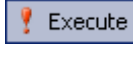
في الدروس القادمة إن شاء الله، سنتعلم تقنيات مختلفة لكتابة استعلامات SQL لإنشاء قواعد بيانات، عند فتح نافذة الاستعلام ستظهر مساحة فارغة بهذا الشكل:



بعد كتابة الاستعلام يمكنك تنفيذه أو مجرد اختياره¹، لتنفيذ الاستعلام:

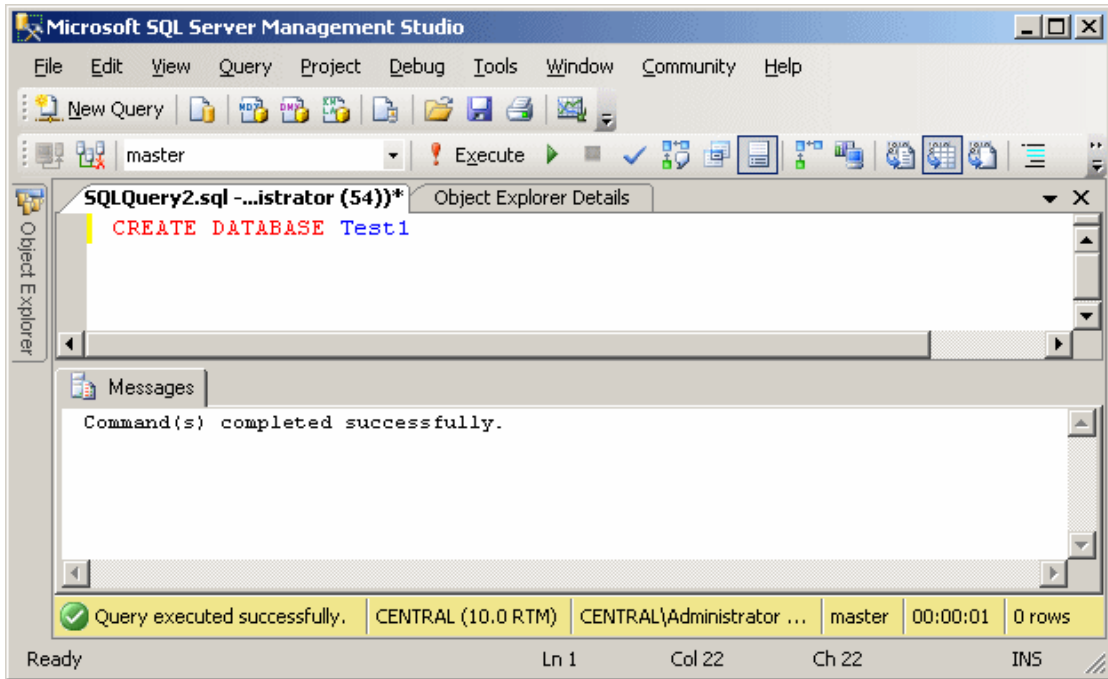
◀ يمكنك الضغط على المفتاح F5

◀ من القائمة الرئيسية، يمكنك النقر على Query -> Execute

◀ في محرر SQL على شريط الأدوات، يمكنك النقر على زر التنفيذ 

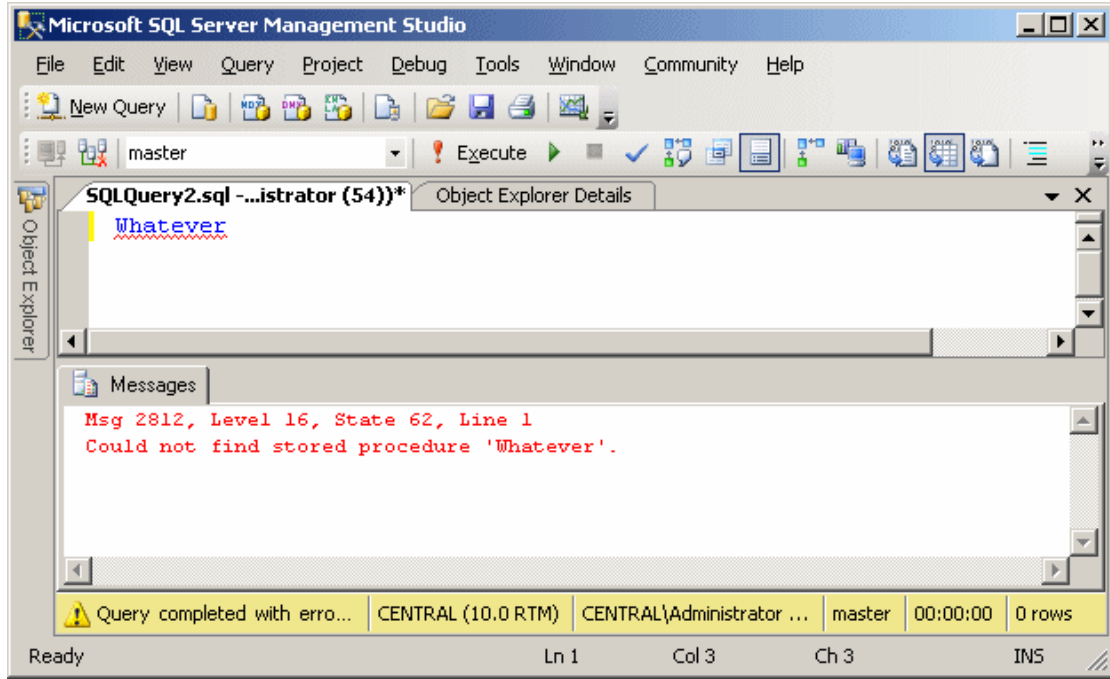
◀ يمكنك النقر بالزر الأيمن على المحرر في أي مكان وانقر على Execute

عند تنفيذ الاستعلام سينقسم المحرر إلى قسمين بشكل أفقي:



قبل تنفيذ الاستعلام يقوم المترجم بتحليله، فإذا وجد خطأ فسيعرضه في الجانب الأسفل من المحرر، هنا مثال:

¹ يوفر SQL Server Management Studio خيارات متقدمة لتحليل الاستعلام، منها تنقيح (Debugging) الاستعلام وتنفيذه خطوة بخطوة، وتحليله في شكل مخطط لقياس الزمن المستغرق للاستعلام وغير ذلك.



إذا لم يكن هناك خطأ في الاستعلام، ستظهر نتيجته أو إشعار بنجاح تنفيذه حسب نوع الاستعلام.

ملحقات لكتابة الاستعلام SQL:

التعليقات:

التعليق هو النص الذي لا يعتبره مترجم الاستعلامات، لذلك يمكن كتابة التعليق بأي شكل تريد وفي أي مكان، يدعم Transact-SQL نوعين من التعليقات، أسلوب التعليق الذي يبدأ بالرمز /* وينتهي بالرمز */، ويمكن استخدامها على عدة أسطر، مثال:

```
/* First find out if the MotorVehicleDivision database we
   want to create exists already.
   If that database exists, we don't want it anymore. So,
   delete it from the system. */
```

يدعم Transact-SQL أيضا رمز شرطة مزدوجة (--) لكتابة التعليق، هذا التعليق يطبق على سطر واحد فقط من النص، لاستخدامه إبدأ السطر بالرمز (--)، مثال:

```
-- =====
-- Database: MotorVehicleDivision
-- =====
```

نهاية عبارة SQL:

في SQL وبعد كتابة عبارة الاستعلام، يمكنك إنهاؤها برمز قاطعة (؛)، وإذا أردت كتابة العديد من العبارات دفعة واحدة، يجب في نهاية كل منها إضافة رمز قاطعة، عندما تستخدم عبارات كثيرة يجب أن تراعي الترتيب حيث أن بعض الاستعلامات يجب أن تأتي بعد أخرى.

استعمال عبارة GO:

تشير الكلمة المحجوزة GO في SQL Server إلى نهاية عبارة استعلام التي سيتم تنفيذها¹، في الواقع واستنادا إلى معايير SQL فإن المطلوب هو رمز قاطعة (؛)، ولكن مترجم Microsoft SQL Server يقبل نهاية الاستعلام بالكلمة GO.

حوصلة:

هدف الدرس:

- التعريف بـ Microsoft SQL Server
- الطرق المختلفة التي يمكنك الحصول بها على Microsoft SQL Server
- النسخة الحالية من Microsoft SQL Server
- الاتصال بخادم قواعد البيانات Microsoft SQL Server.

تمارين: استخدام موجه الأوامر

- من خدمة المساعدة لـ Microsoft SQL Server (Books Online)، ابحث عن الأوامر التي تستخدم للاتصال بالخادم
- حاول الاتصال بالخادم من Command Prompt أو من Windows PowerShell.

¹ بمعنى أن مترجم SQL في SQL Server عندما يبدأ بقراءة الاستعلام، فإنه لن يقوم بالتنفيذ حتى يصل إلى نهاية الاستعلام أو يجد الكلمة المحجوزة GO.

مدخل إلى قواعد البيانات

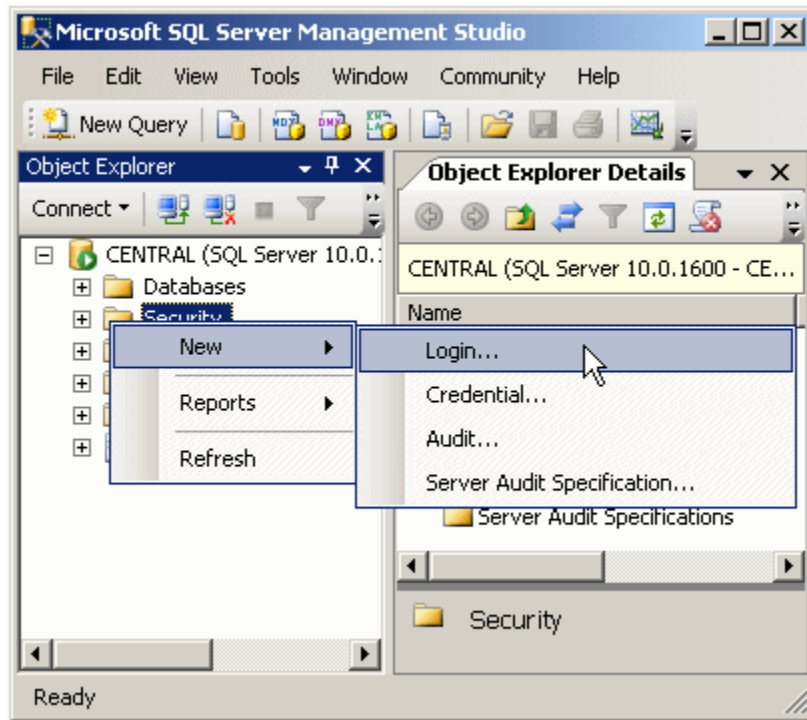
مقدمة لإنشاء قاعدة بيانات:

مبدئياً، قاعدة بيانات هي مجموعة من الملفات التي لها اسم وموقع معين، حيث أن هناك طرقاً مختلفة للاتصال بالخادم للتحكم في هذه الملفات، ولذلك هناك طرق مختلفة لإنشاء قاعدة بيانات.

لإنشاء قاعدة بيانات جديدة في Microsoft SQL Server Management Studio، انقر من متصفح الكائنات بالزر الأيمن على البند Databases ثم اختر New Database... وسيفتح مربع الحوار لقاعدة البيانات الجديدة.

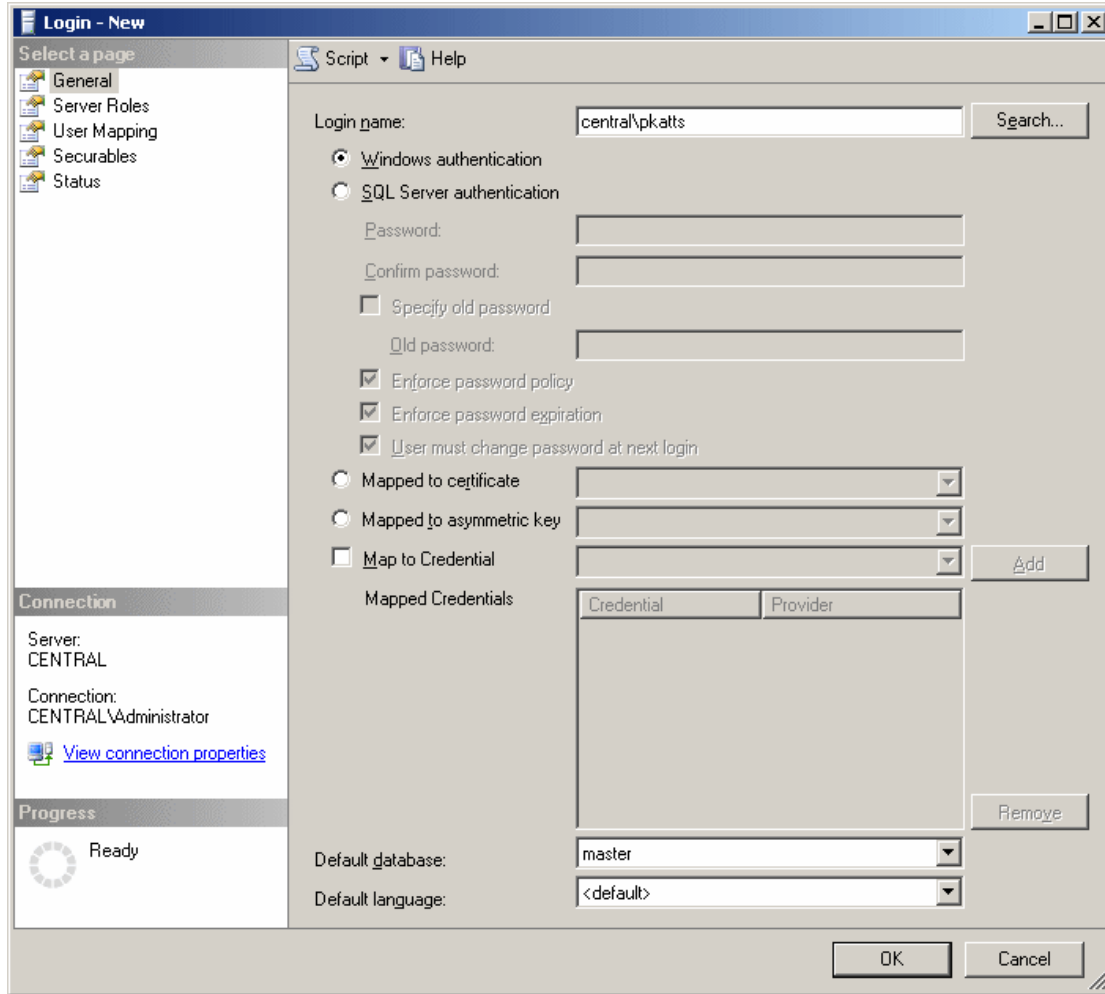
تطبيق عملي: إنشاء حساب تسجيل

- 1- تأكد أنك قد أنشأت حساب المستخدم باسم pkatts كما رأينا في الدرس الأول، ثم شغل SQL Server Management Studio.
- 2- في مربع حوار الاتصال Connect to Server، أدخل معلومات التسجيل المناسبة، ثم انقر على Connect.
- 3- في متصفح الكائنات، افتح اسم الخادم إذا لزم الأمر، ثم انقر بالزر الأيمن على البند Security، اختر New ثم انقر على Login...



4- أدخل اسم المالك (اسم الكمبيوتر يليه الرمز \ ثم اسم المستخدم المسجل كمدیر)
ComputerName\pkatts (في Windows XP أو Vista) أو DomainName\pkatts
(في ويندوز Server)، يمكنك أيضا النقر على Search... للعثور على اسم المستخدم.

5- انقر على OK



اسم قاعدة بيانات:

لعل أهم عنصر عند إنشاء قاعدة بيانات هو الاسم. تعتبر SQL مرنة جدا عندما يتعلق الأمر بالأسماء، فهي أقل تقييدا من غيرها من لغات الكمبيوتر، ومع ذلك هناك قواعد يتوجب عليك إتباعها عند تسمية أي شيء موجود في قواعد البيانات:

- يمكنك أن تبدأ بأي من الحروف اللاتينية (من a إلى z)¹، أو الأرقام (من 0 إلى 9)، شرطة (_) أو أي رمز غير مقروء، أمثلة: **act**, **%783**, **Second**, **_n**.
- بعد أول حرف (حرف، رقم، شرطة، رمز)، يمكن أن تستخدم مجموعة من الحروف أ/و أرقام أو/و رموز، أمثلة: **_n24** أو **act_52_t**.

¹ ينصح في SQL Server بعدم استخدام حروف Unicode لكتابة أسماء الأغراض، بما في ذلك حروف باللغة العربية.

- ويمكن استعمال رمز الفراغ¹ مثال: #g10b, (records), govmt@st, countries

.\$Survey

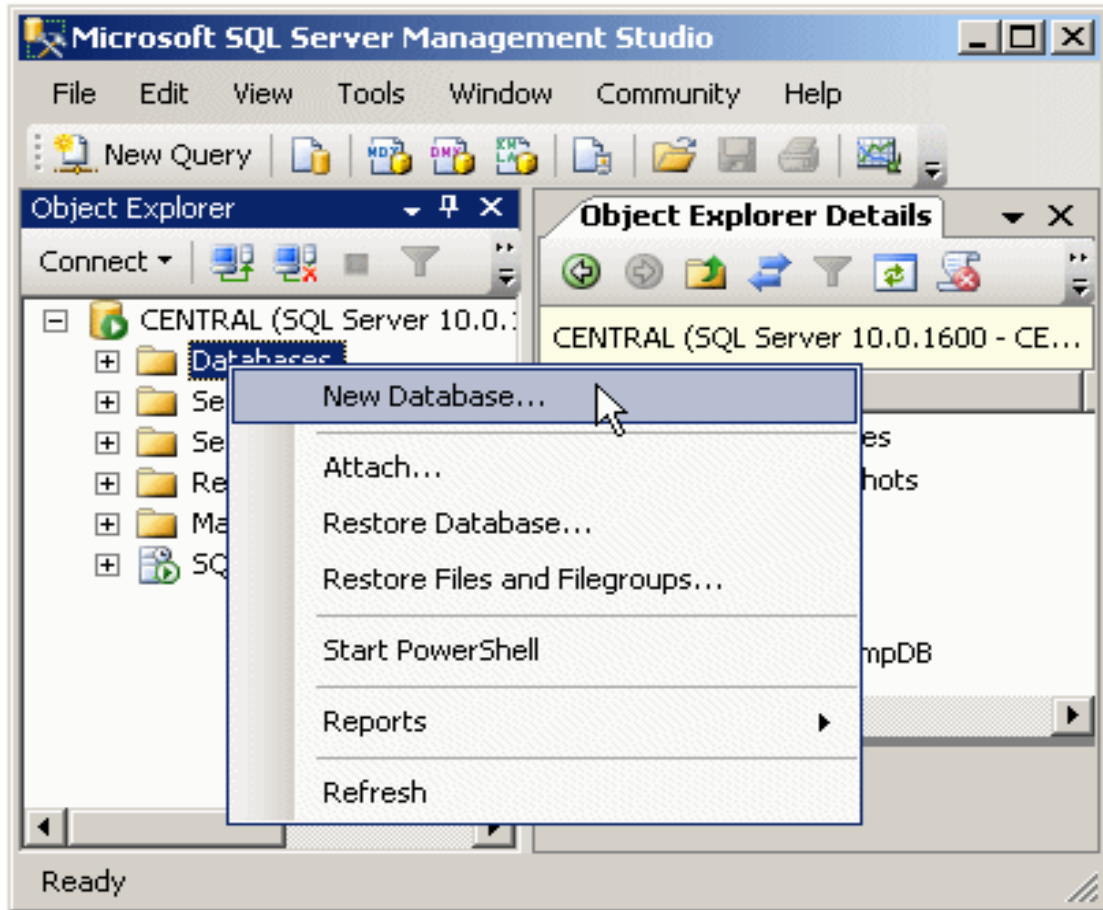
لأجل مرونة SQL فإنه يمكن أن يكون صعبا الحفاظ على الأسماء في قاعدة بيانات، لذلك هناك اصطلاحات سنستخدمها لتسمية أي غرض، وسنعمد القواعد المستخدمة في أكثر لغات البرمجة ما لم ينص على خلاف ذلك. سنجعل أول حرف هو أحد الأحرف اللاتينية a..z أو الرمز "_"، وبعده مزيج من الأحرف والأرقام والرمز "_"، وإذا كان الاسم مكون من كلمتين سنجعل الحرف الأول لكل كلمة حرف كبير (Uppercase)، أمثلة: Countries Statistics, Global Survey, _RealSport, FullName, DriversLicenseNumber.

إذا كان اسم الغرض يحتوي على فراغات فيجب كتابته بين معقوفين "[]"، مثل [Global Survey], [Countries Statistics]، ويمكن استخدامها حتى لو كان الاسم لا يشمل فراغ، مثل: [FullName], [DriversLicenseNumber].

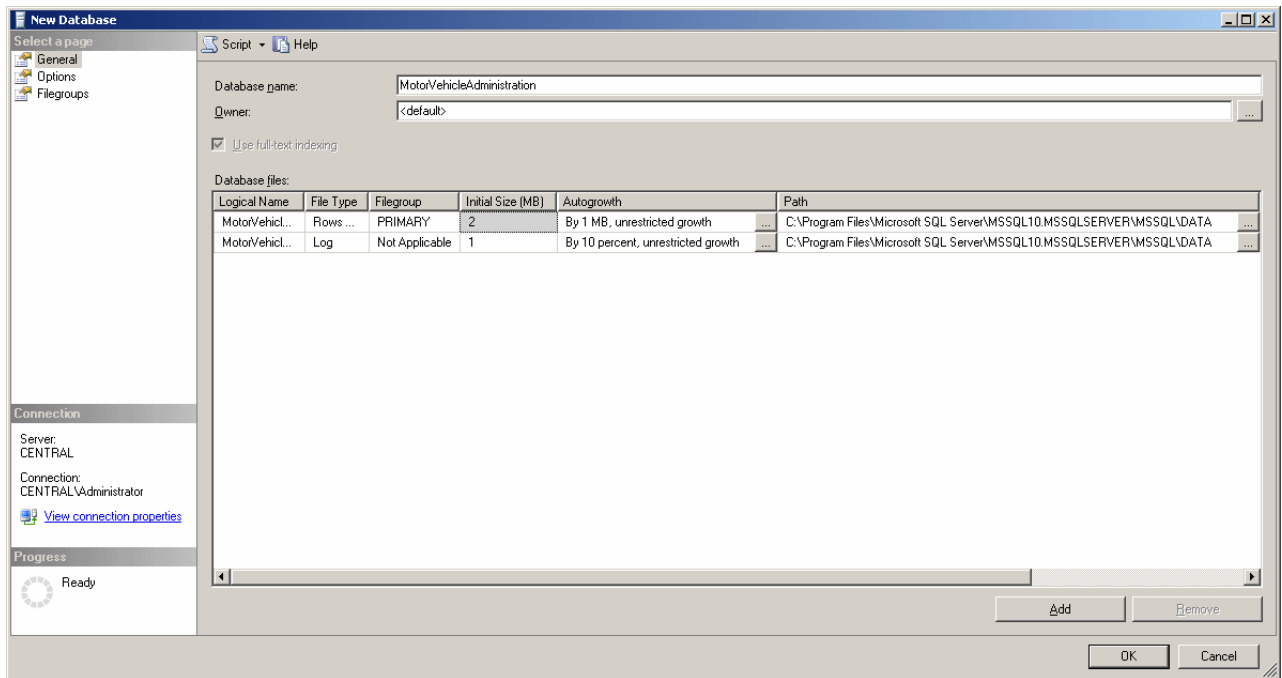
تطبيق عملي: إنشاء قاعدة بيانات باستخدام SQL Server Management

1- في متصفح الكائنات انقر بالزر الأيمن على Databases ثم اختر New Database...

¹ استخدام مساحة فراغ في كتابة الأسماء يجبرك فيما بعد باستدعاء الأغراض مع وضع أسمائها بين معقوفتين [] أو شالتين " .



2- في خانة الاسم، اكتب MotorVehicleAdministration



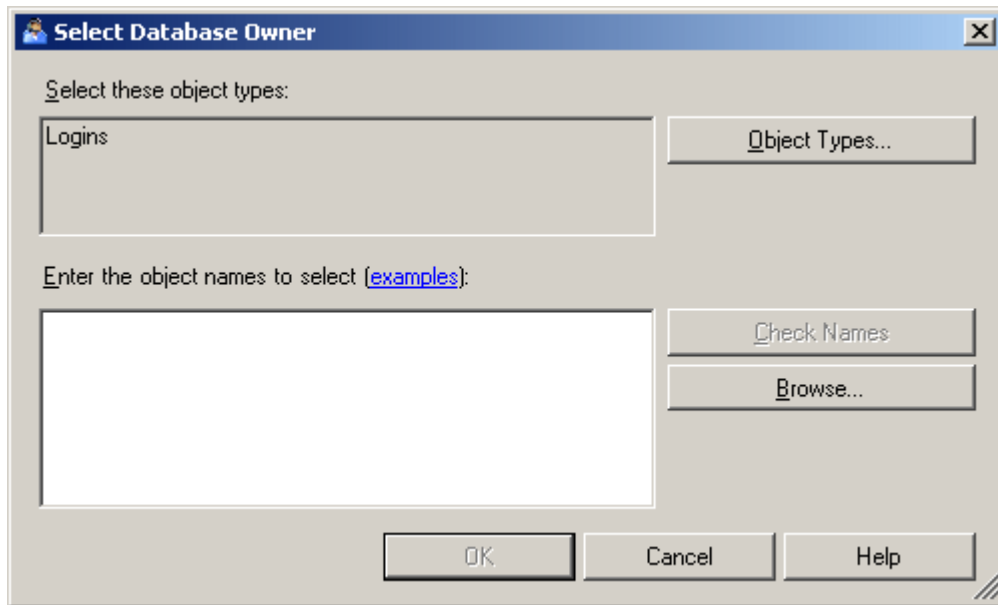
مالك قاعدة البيانات:

يريد SQL Server عند إنشاء قاعدة بيانات جديدة تتبع الذي قام بإنشائها، وهو ما يعرف باسم مالك قاعدة البيانات (Database Owner)¹.

بشكل افتراضي يقوم Microsoft SQL Server بإنشاء حساب تسجيل لقاعدة البيانات خاص يدعى **dbo** (لمالك قاعدة البيانات)، عندما تقوم بإنشاء قاعدة بيانات دون تحديد مالك معين يستخدم هذا الحساب.

يملك الحساب **dbo** صلاحيات القيام بجميع أنواع العمليات التي يمكن القيام بها على قاعدة بيانات، وهذا شيء مريح في معظم الحالات، ومع ذلك يمكنك تحديد مستخدم آخر (قد تم إنشاؤه) كمالك قاعدة البيانات.

نرجع للمثال... لتحديد مالك قاعدة البيانات التي قمنا بإنشائها، انقر على <default> في مربع النص **Owner**، ثم اكتب اسم الحاسب متبوعاً بالرمز "\" ثم اسم المستخدم الذي يملك قاعدة البيانات، أو يمكنك النقر على زر الاستعراض (على الجانب الأيمن من مربع النص)، وهذا من شأنه أن يفتح مربع الحوار **Select Database Owner**:



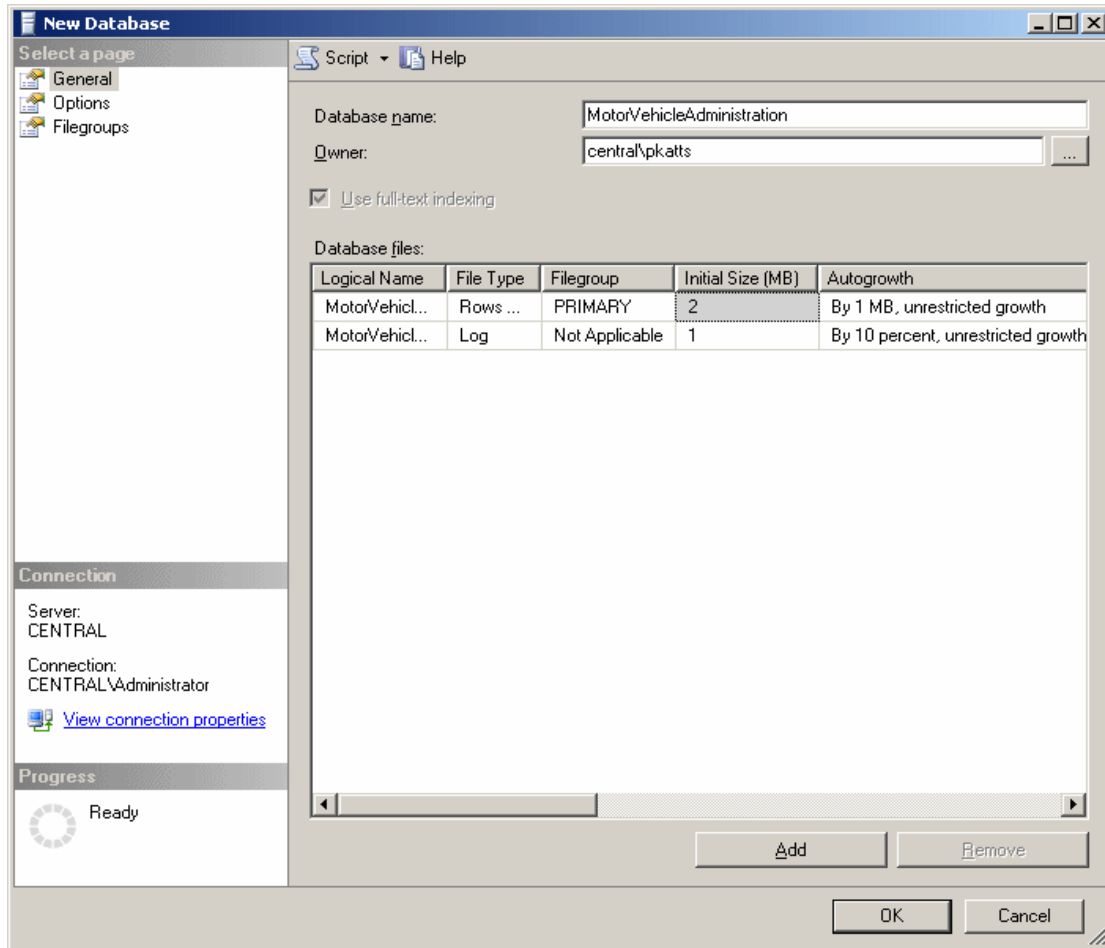
في خانة أسماء الكائنات (Object names) أدخل اسم الكمبيوتر الذي سيستخدم قاعدة البيانات، بعد ذلك انقر على **Check Names**، إذا كان الاسم صحيح سيتم قبوله وإلا ستتلقى

¹ مالك قاعدة البيانات عند إنشائها هو الحساب الذي تم به تسجيل الدخول إن لم يحدد غيره.

رسالة خطأ تقترح عليك تصحيحه، يمكنك النقر على الزر Browse...، هذا من شأنه أن يفتح تصفح نافذة Objects dialog لتحديد أحد الحسابات المسجلة كمالك لقاعدة البيانات، إذا رأيت المستخدم الذي ترغب في استخدامه (وهناك سنجد central\pkatts) انقر على مربع الاختيار ثم على OK¹.

تطبيق عملي: تحديد مالك قاعدة بيانات

1- في الخانة Owner، انقر على <default> واكتب DomainName\pkatts مع استبدال DomainName باسم جهازك*، أو تخطي هذه الخطوة وانقر على زر Browse... من الجانب الأيمن لتحديد اسم المستخدم.



¹ وهناك أيضا ستجد الحساب الافتراضي sa الذي ينشئه SQL Server.
* للتعرف على اسم جهازك (DomainName) على My Computer انقر بالزر الأيمن، ثم اختر Properties ثم اختر التبويب Computer Name لتشاهد اسم جهازك على الجهة اليمنى.

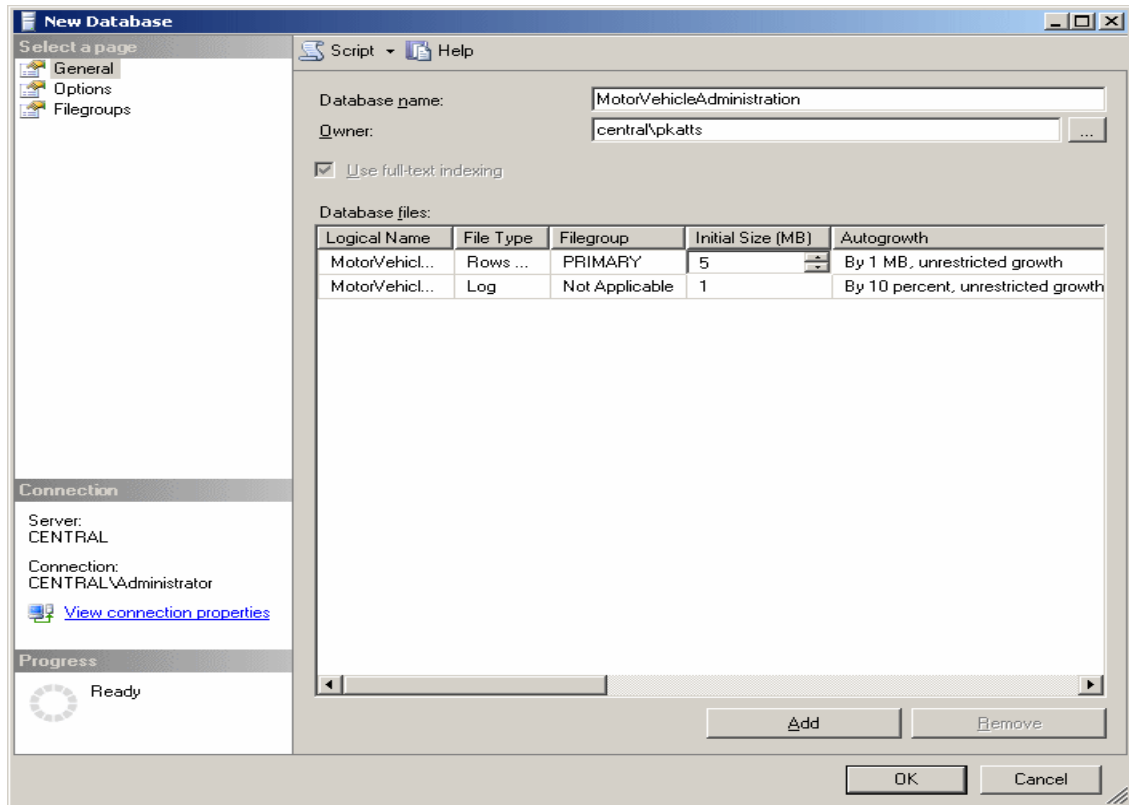
الحجم الابتدائي لقاعدة البيانات:

عند إنشاءك لقاعدة بيانات لا يمكنك معرفة عدد السجلات والأغراض التي سيشملها المشروع، لذلك فإن قاعدة البيانات تحجز مساحة معينة على الأقل في البداية، حجم المساحة التي تستخدمها قاعدة البيانات تشير إلى حجم ملفاتها، إذا كنت تستخدم مربع الحوار New Database فبعد تحديد اسم قاعدة البيانات والموافقة على إنشائها، سيقوم المترجم تلقائياً بتحديد حجمها الابتدائي 2 Mb، هذا يكفي للبداية وبالطبع يمكنك تغيير هذا الحجم في وقت لاحق أو زيادته إذا لزم الأمر.

إذا كنت ترغب في تحديد حجم مختلف عن الافتراضي، وإذا كنت في نافذة New Database فمن عمود Initial Size يمكنك تغيير الحجم بما يناسب.

تطبيق عملي: تحديد حجم ملف قاعدة البيانات

- في منطقة Database Files، انقر على العمود Initial Size، ثم على زر السهم للأعلى لزيادة القيمة إلى 5.



موضع قاعدة بيانات:

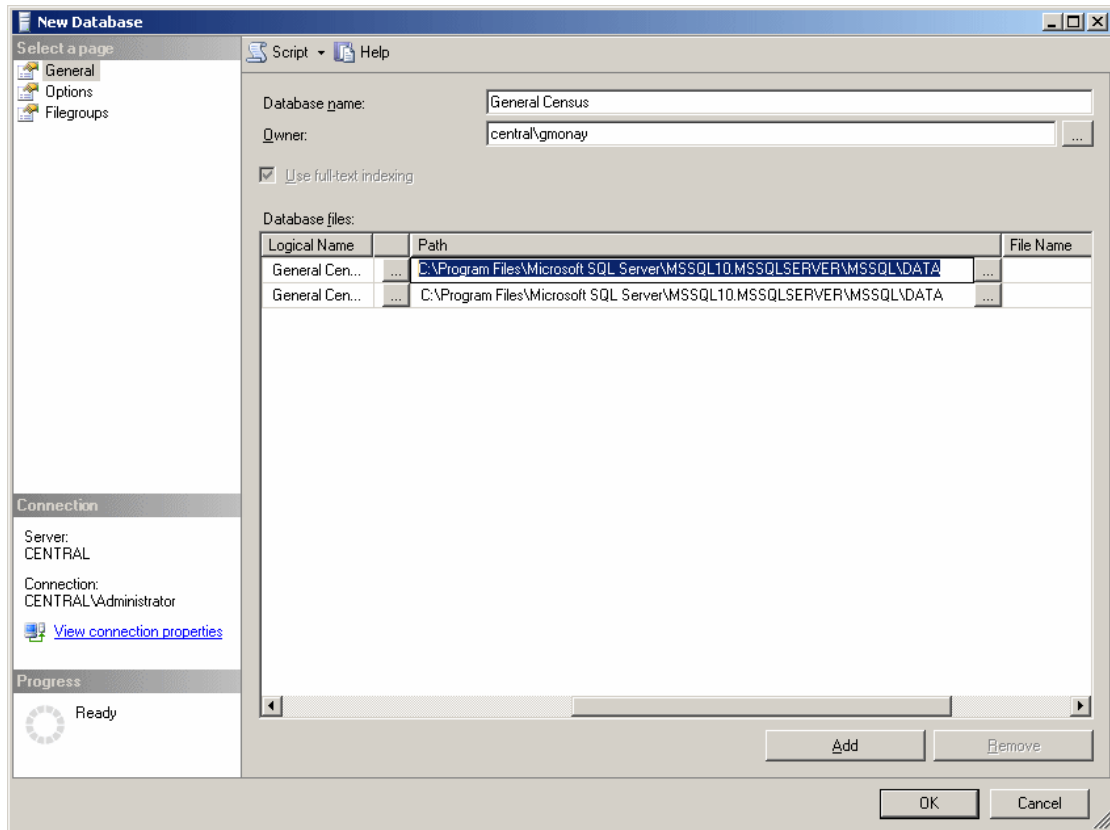
من خلال تجربتك في استخدام الحاسب فإنك تدرك أن لكل ملف مسار، مما يتيح لنظام التشغيل معرفة مكان الملف ليستدعي من قبل تطبيق آخر.

بشكل افتراضي، عند إنشاء قاعدة بيانات جديدة سيجعلها Microsoft SQL Server في المسار التالي:

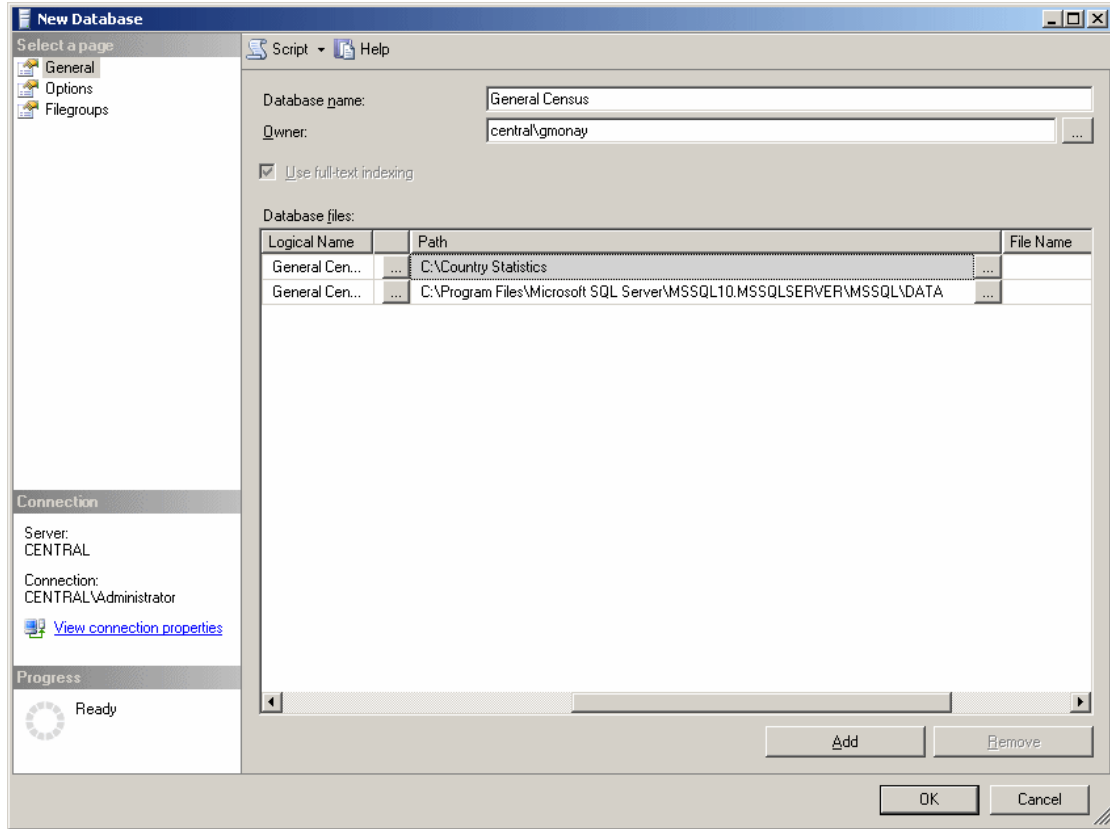
Drive:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA

في مربع الحوار New Database، إذا قمت بتحديد اسم قاعدة البيانات ثم وافقت على ذلك، فسيقوم المترجم تلقائياً بإنشاء ملف جديد بامتداد MDF. وهذا هو الملف الرئيسي لقاعدة البيانات.

إذا كنت لا ترغب باستخدام المسار الافتراضي يمكنك تغييره، في مربع الحوار New Database، لتغيير المسار أدخل المسار الجديد في العمود Path:

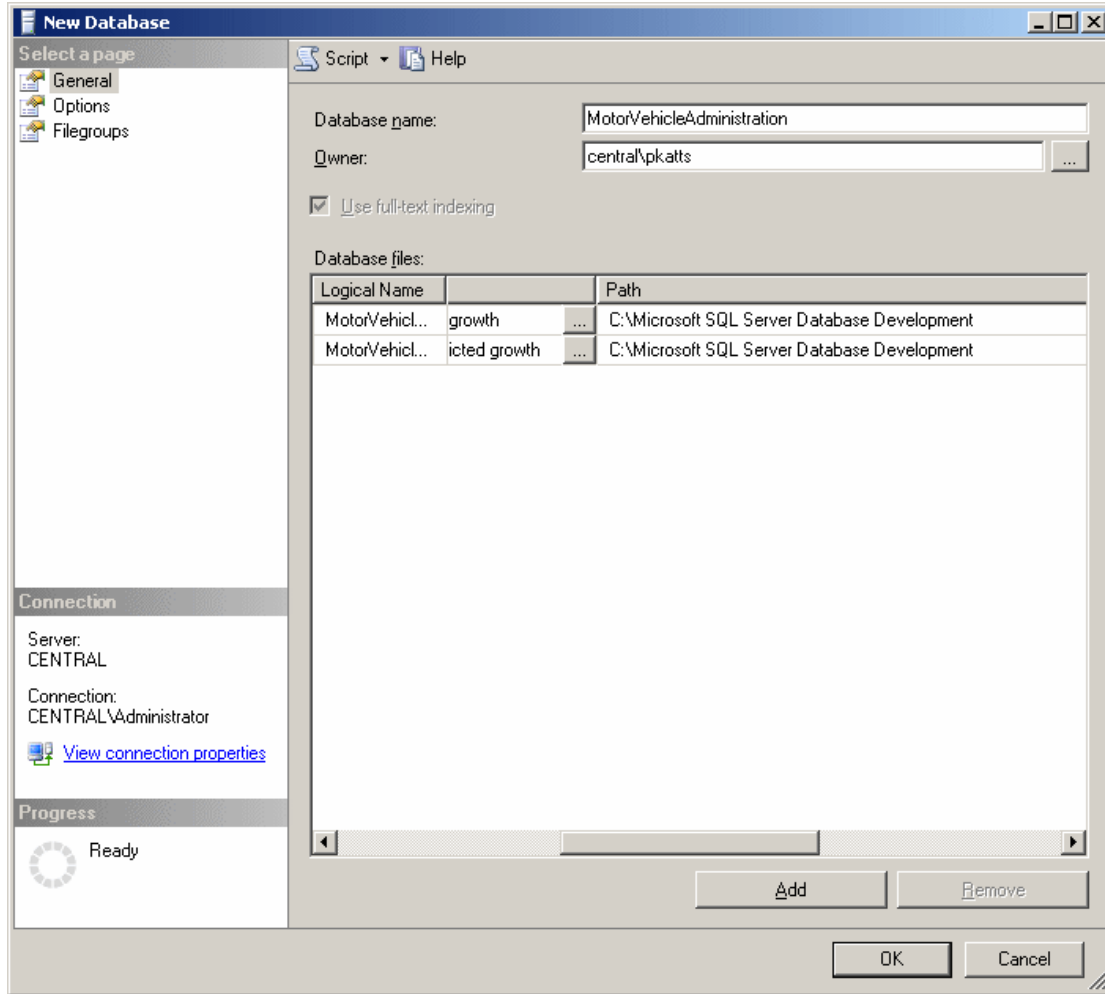


يمكنك تغييره بما يوافق اختيارك.



تطبيق عملي: فحص مسار ملف بيانات

- 1- من الجانب الأيمن ، وتحت اسم العمود Path انقر على مسار الملف.
- 2- من متصفح ويندوز (Windows Explorer) أنشئ مجلد باسم Microsoft SQL Server Database Development في القرص C:.
- 3- ارجع إلى مربع الحوار New Database وانقر على الزر ...
- 4- حدد المجلد Microsoft SQL Server Database Development الذي قمت بإنشائه.
- 5- افعل الشيء نفسه لمسار الملف الآخر.
- 6- انقر على OK.



قاعدة البيانات الافتراضية:

يقوم Microsoft SQL Server عند تثبيته بتنشيط 4 قواعد بيانات: master، model، msdb، tempdb قواعد البيانات هذه ستكون للاستخدام الداخلي، يعني هذا أن عليك أن تتجنب استخدامها مباشرة، إلا إذا كنت تعرف بالضبط ما تقوم به.

قاعدة بيانات النظام:

واحدة من أهم قواعد البيانات التي تم تركيبها مع Microsoft SQL Server تدعى master، تحمل هذه القاعدة كل المعلومات عن الخادم الذي تم عليه تثبيت Microsoft SQL Server، مثلا: رأينا فيما سبق أن لأداء أي عمل على الخادم يجب عليك تسجيل الدخول، قاعدة البيانات الرئيسية تحدد متى وكيف لأي شخص (مستخدم) دخل إلى قاعدة البيانات.

إضافة إلى تحديد المستخدمين، تقوم قاعدة البيانات الرئيسية master بتتبع كل ما يحدث على مستوى الخادم، بما في ذلك إنشاء وإدارة قواعد البيانات، يجب ألا تعبت بقاعدة البيانات الرئيسية وإلا فقد تهدد النظام بخلل ما، كعدم اشتغال النظام.

إنشاء قاعدة بيانات بالاستعلام:

تستخدم لإنشاء قاعدة بيانات بـ SQL العبارة التالية:

```
CREATE DATABASE DatabaseName
```

رأينا أنه يمكن استخدام نافذة الاستعلام لإدخال شيفرة SQL.

عبارة CREATE DATABASE (تذكر أن SQL ليست حساسة لحالة الحروف) ضرورية، حيث يمثل DatabaseName اسم قاعدة البيانات الجديدة التي ستقوم بإنشائها، على الرغم من أن SQL ليست حساسة لوضع الأحرف فيجب أن تكون على علم من حالات الأحرف لاسم الأغراض الخاصة بك، كل عبارة استعلام في SQL يمكن إنهاؤها برمز قاطعة ";"، وإن كان هذا الشرط مطلوباً في العديد من عبارات SQL، إلا أن في Microsoft SQL Server يمكنك حذفه، أو كتابته على الشكل التالي:

```
CREATE DATABASE DatabaseName ;
```

استخدم هذه الصيغة إذا لم تكن ترغب بإضافة أي خيار آخر، رأينا سابقاً أن قاعدة بيانات تخزن في ملف أو أكثر وشاهدنا المسار الافتراضي للملفات، ورأينا أيضاً أنه يمكنك تحديد موقع الملفات على القرص، لتحديد مسار الملف الرئيسي في قاعدة البيانات يمكنك استخدام الصيغة التالية:

```
CREATE DATABASE DatabaseName  
ON PRIMARY  
( NAME = LogicalName, FILENAME = Path )
```

ثلاثة عوامل يمكن تغيير قيمها من هذه الصيغة هي اسم قاعدة البيانات (DatabaseName) التي شرحناها، الاسم المنطقي (LogicalName)، والمسار (Path). الاسم المنطقي يمكن أن يكون أي اسم في كلمة واحدة ولكن ينبغي أن يختلف عن اسم قاعدة البيانات، المسار هو دليل موقع الملف، وينتهي هذا المسار باسم الملف مع بامتداد .mdf، كما يجب أن يكون كاملاً وبين شاليتين "، مثال:

```

CREATE DATABASE NationalCensus
ON PRIMARY
( NAME = DataRepository, FILENAME =
'C:\Exercises\NationalCensus.mdf' )
GO

```

إضافة إلى الملف الرئيسي، قد ترغب في إنشاء وتخزين ملف السجل (Log file)، لتحديد ملف السجل لقاعدة البيانات، يمكنك استخدام الصيغة التالية:

```

CREATE DATABASE DatabaseName
ON PRIMARY
( NAME = LogicalName, FILENAME = Path.mdf )
LOG ON
( NAME = LogicalName, FILENAME = Path.ldf )

```

العبارة الجديدة LOG ON تعني إنشاء ملف السجل، مثل الملف الرئيسي ملف السجل يجب أن يكون له اسم (مع الاسم المنطقي)، وينتهي المسار باسم الملف وامتداد .ldf. مثال:

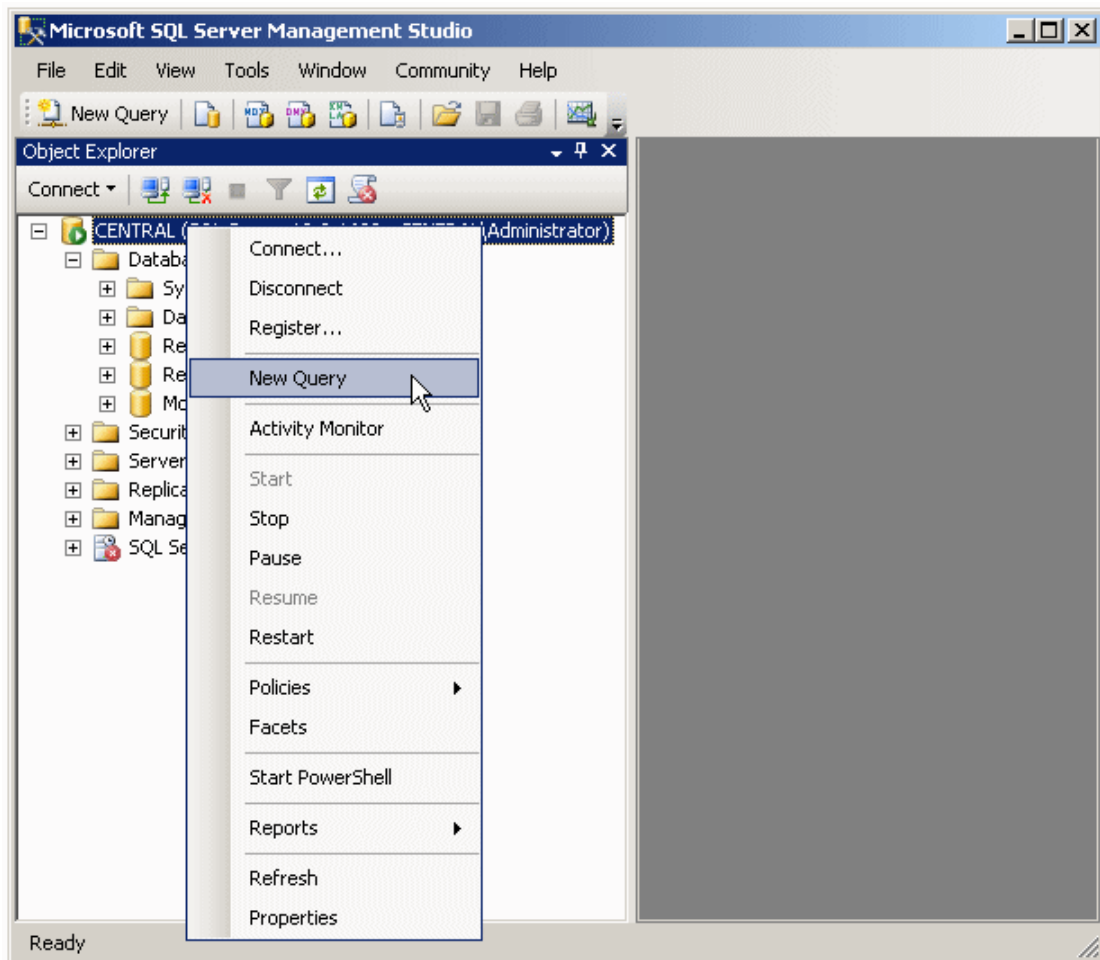
```

CREATE DATABASE NationalCensus
ON PRIMARY
( NAME = DataRepository, FILENAME =
'C:\Exercises\NationalCensus.mdf' )
LOG ON
( NAME = DataLog, FILENAME = 'C:\Exercises\NationalCensus.ldf' )
GO

```

تطبيق عملي: إنشاء قاعدة بيانات باستخدام SQL

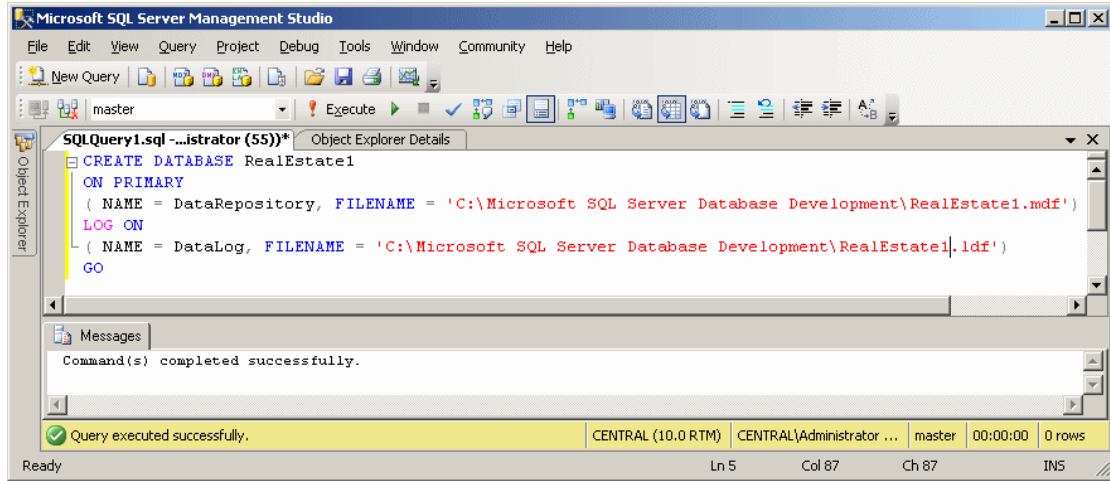
1- بالزر الأيمن انقر على متصفح الكائنات ثم اختر New Query لإنشاء استعلام جديد.



2- في المساحة الفارغة أكتب الشيفرة:

```
CREATE DATABASE RealEstate1
ON PRIMARY
( NAME = DataRepository, FILENAME = 'C:\Microsoft SQL Server Database
Development\RealEstate1.mdf' )
LOG ON
( NAME = DataLog, FILENAME = 'C:\Microsoft SQL Server Database
Development\RealEstate1.ldf' )
GO
```

3- لتنفيذ الاستعلام اضغط F5.



استخدام قوالب SQL

لتوفير المزيد من السهولة في كتابة الاستعلامات تم تجهيز Microsoft SQL Server Management Studio بمجموعة من شفرات SQL جاهزة للاستعمال (تدعى بالقوالب Templates)، بحيث يمكنك استخدامها لأداء مختلف المهام، كإنشاء قاعدة بيانات والتعديل عليها وحذفها وغير ذلك، يمكن استخدام هذه القوالب من خلال قائمة Template Explorer.

ل للوصول إلى متصفح القوالب (Template Explorer)، من القائمة الرئيسية انقر على View ثم Template Explorer، لاستعمال أحد القوالب يجب أولاً فتح نافذة استعمال.

- لإنشاء قاعدة بيانات جديدة باستخدام القوالب، قم بتوسيع بند Databases من متصفح القوالب، ثم قم بسحب وإفلات البند Create Database إلى نافذة الاستعمال، قاعدة البيانات الجديدة التي سيتم إنشاؤها بهذا الاستعلام ستكون في الخادم المتصل به حالياً.
- إذا كان لديك أكثر من خادم، وأردت إنشاء قاعدة بيانات على خادم آخر أو استعمال عدة اتصالات، فقم بتوسيع البند Databases من متصفح القوالب، ثم انقر بالزر الأيمن على Create Database ثم Open، ثم من النافذة Connect to Database Engine اختر الخيارات المناسبة وانقر OK للاتصال بالخادم.

في الحالتين سيقوم MS SQL Server Management Studio بتوليد الاستعلام التالي:

```

-- =====
-- Create database template
-- =====
USE master
GO

-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'<Database_Name, sysname,
Database_Name>'
)
DROP DATABASE <Database_Name, sysname, Database_Name>
GO

CREATE DATABASE <Database_Name, sysname, Database_Name>
GO

```

الآن أنت بحاجة إلى تعديل الشيفرة لتنفيذها وإنشاء قاعدة البيانات. في الدروس السابقة استعرضنا بعض العبارات مثل علامة التعليق: --، و بعض الاستعلامات مثل: CREATE DATABASE، GO، وستأتي البقية فيما بعد إن شاء الله.

للتعديل في هذا الاستعلام يجب تعديل قيم المتغيرات، يكفي لأجل ذلك الاختيار من قائمة Query خاصة Specify Values For Template Parameters، ثم أدخل اسم كل متغير.

صيانة قاعدة البيانات:

حذف قاعدة بيانات:

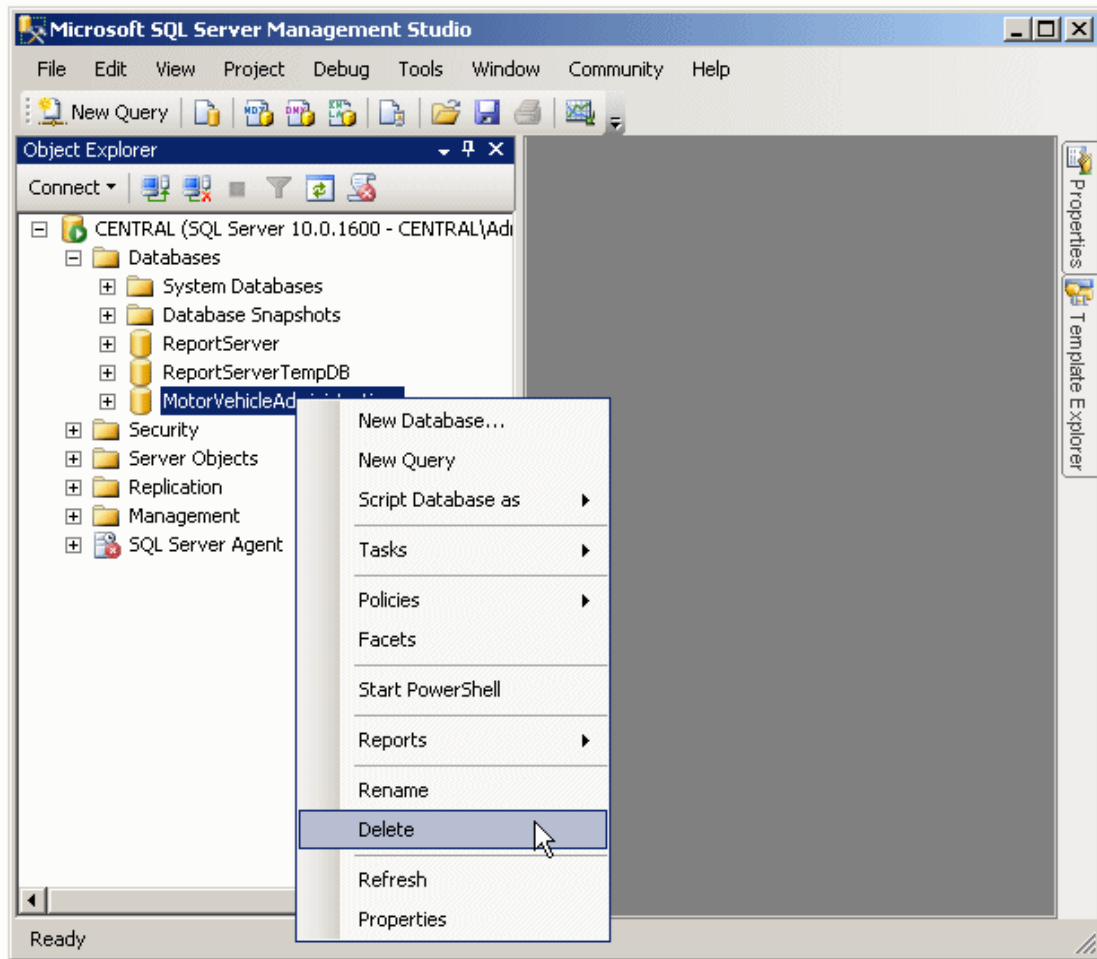
إذ تعد بحاجة إلى قاعدة بيانات يمكنك حذفها، وبصرف النظر عن كيفية إنشاء قاعدة بيانات يمكنك حذفها بأي طريقة، كل أداة توفر وسائلها الخاصة.

باستخدام Microsoft SQL Server Management Studio:

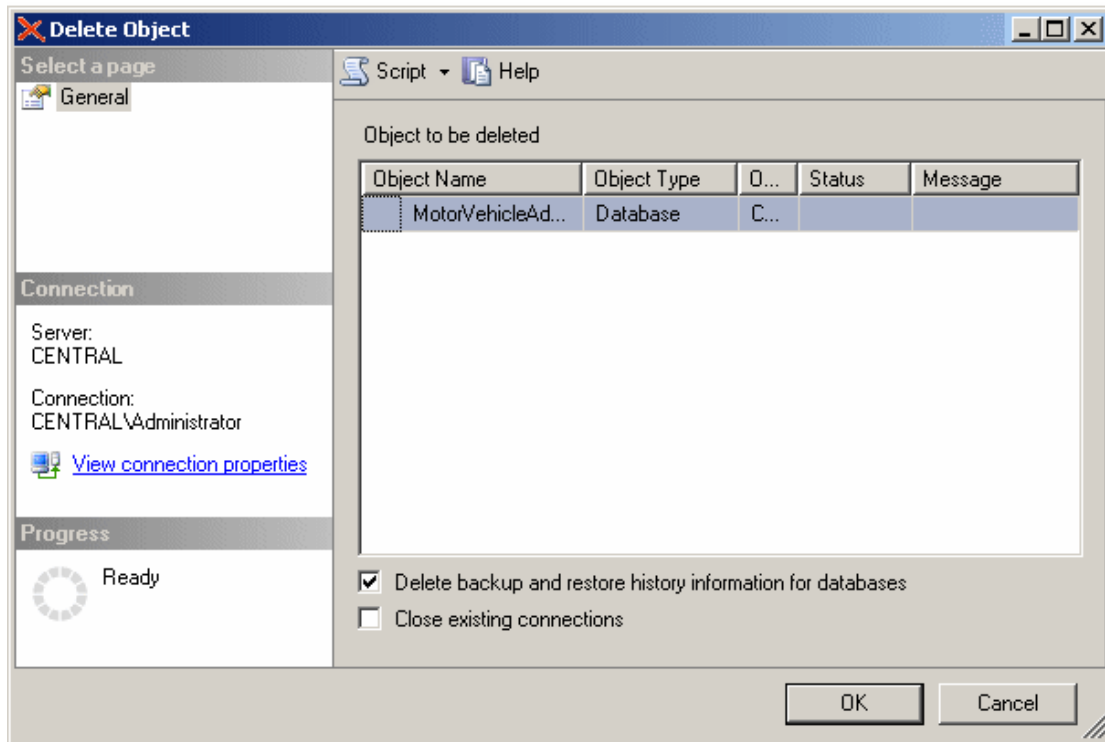
لحذف قاعدة بيانات انقر بالزر الأيمن عليها من خلال متصفح الكائنات، ثم انقر على Delete، ليظهر مربع حوار يطالبك بتأكيد خيارك، انقر على OK للحذف، الخيار Close existing connections مهم ومفاده غلق الاتصال بقاعدة البيانات المراد حذفها قبل محاولة الحذف.

تطبيق عملي: حذف قاعدة البيانات

1- في متصفح الكائنات، انقر بالزر الأيمن على MotorVehicleAdministration وانقر على Delete.



في مربع الحوار ، انقر على OK.



حذف قاعدة البيانات باستخدام SQL:

لحذف قاعدة بيانات في SQL باستخدام محرر الاستعلام استخدم عبارة DROP DATABASE متبوعة باسم قاعدة البيانات، الصيغة العامة:

`DROP DATABASE DatabaseName ;`

قبل حذف قاعدة بيانات في SQL يجب التأكد أولاً من أنها لا تستخدم في الحصول على أي عرض، ولا تستخدم من قبل أي كان.

تطبيق عملي: حذف قاعدة بيانات باستخدام SQL



1- من شريط الأدوات القياسي انقر على زر استعلام جديد

2- اكتب الاستعلام التالي:

```
DROP DATABASE RealEstate1;
GO
```

اضغط على F5 لتنفيذ الاستعلام.

أوامر قاعدة بيانات:

قاعدة البيانات الحالية:

عند كتابة الشيفرة في محرر الاستعلام، عليك أن تعرف قاعدة البيانات التي تعمل عليها، وإلا قد يحصل تنفيذ استعلام بالخطأ على قاعدة البيانات، لتحديد قاعدة البيانات الحالية برمجياً (باستخدام SQL)، اكتب الكلمة المفتاحية USE متبوعة باسم قاعدة البيانات، الصيغة هي:

`USE DatabaseName;`

حيث DatabaseName يمثل اسم قاعدة البيانات التي سيتم التعامل معها.

في SQL Server Management Studio يمكنك تحديد قاعدة البيانات الحالية من قائمة الاختيارات في شريط الأدوات "تحرير SQL"



تحديث قائمة قواعد البيانات:

بعض النوافذ التي تظهر قواعد البيانات، مثل SQL Server Management Studio، لا تقوم بتحديث القائمة على الفور إذا حدثت عملية خارجها، على سبيل المثال: إذا قمت بإنشاء قاعدة بيانات في محرر الاستعلام فلن يظهر اسمها في متصفح الكائنات فور تطبيق الاستعلام، وبسبب هذه التغيرات الخارجية ينبغي عليك تحديث القائمة التي تحمل أسماء قواعد البيانات.

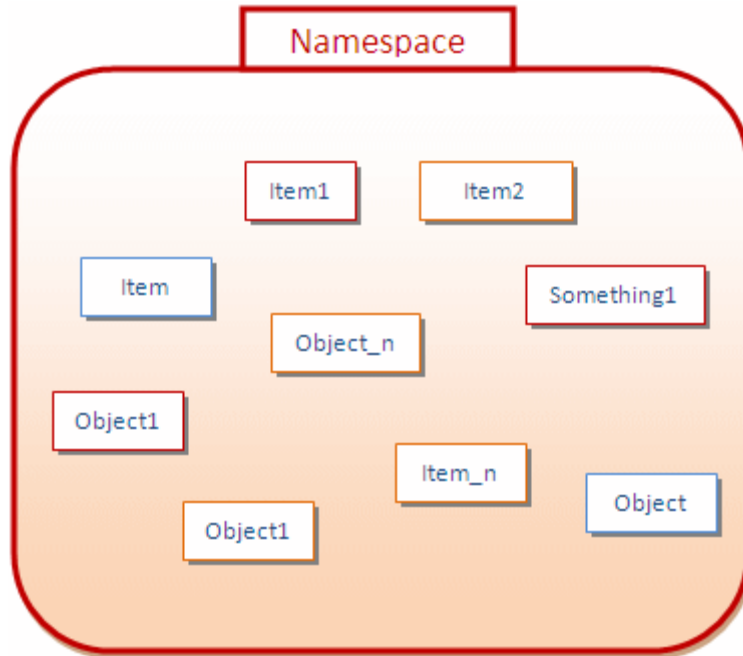
في SQL Server Management Studio، لتحديث قائمة قواعد البيانات يمكنك النقر بالزر الأيمن على البند في متصفح الكائنات ثم اختر Refresh، سيقوم البرنامج بتحديث هذه الفئة فقط، مثلاً لتحديث قائمة قواعد البيانات في متصفح الكائنات يمكنك النقر بالزر الأيمن على بند Databases وانقر على Refresh.

المخططات:

تمهيد إلى مجال الأسماء Namespaces:

Namespaces هي تقنية تسمح بإنشاء سلسلة من العناصر يحمل كل منها اسم فريد، على سبيل المثال إذا قمت بإنشاء العديد من قواعد البيانات حيث هناك احتمال بأن يتكرر الاسم، يمكنك استخدام namespace لعزل قواعد البيانات في مجالات أسماء (namespaces) مختلفة.

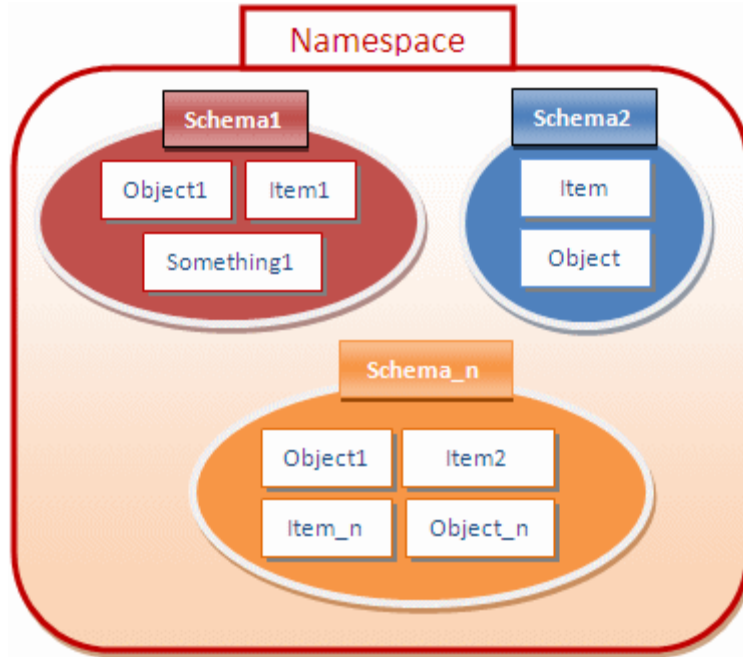
لإدارة العديد من الأغراض في خادم قاعدة البيانات الخاصة بك، يمكنك استخدام namespaces لوضع أي غرض عليها على غرار قواعد البيانات. يمكن توضيح ذلك على النحو التالي:



نلاحظ أن هناك أنواعا مختلفة من الأغراض لكل منها اسم يميزه في namespace.

مقدمة إلى المخططات (Schemas):

يمكنك إنشاء الأغراض حسب رغبتك ضمن namespace، للمزيد من السيطرة وإدارة أحسن للأغراض داخل أي namespace، يمكنك وضعها في مجموعات فرعية تسمى **مخططات (Schemas)**. المخططات هي مجموعة من الأغراض ضمن namespace، يمكنك الحصول على أكبر عدد تريده من المخططات:



لاحظ أن namespace يمكنه احتواء مخططات (Schemas)، والمخطط بدوره يمكن أن يحتوي على أغراض أخرى (سنتعامل مع الأغراض في جميع دروسنا).

لإدارة المخططات في نفس namespace، تحتاج إلى وسيلة لتمييز كل مخطط، وعلى هذا فإن لكل مخطط اسم، في الشكل الأول مخطط اسمه Schema1، والثاني Schema2 والآخر Schema_n.

هناك نوعين من المخططات التي يمكن استخدامها المنشأة مسبقاً والتي قمت بإنشائها، عند تثبيت Microsoft SQL Server يتم تثبيت بعض المخططات، أحد هذه المخططات يسمى sys.

مخطط sys يتضمن قائمة لبعض الأغراض التي توجد في النظام الخاص بك، واحدة من هذه الأغراض تسمى databases (وهي في الواقع view)، عند إنشاء قاعدة بيانات سيضاف اسمها إلى قائمة الغرض databases باستخدام نفس الاسم الذي أعطيتها.

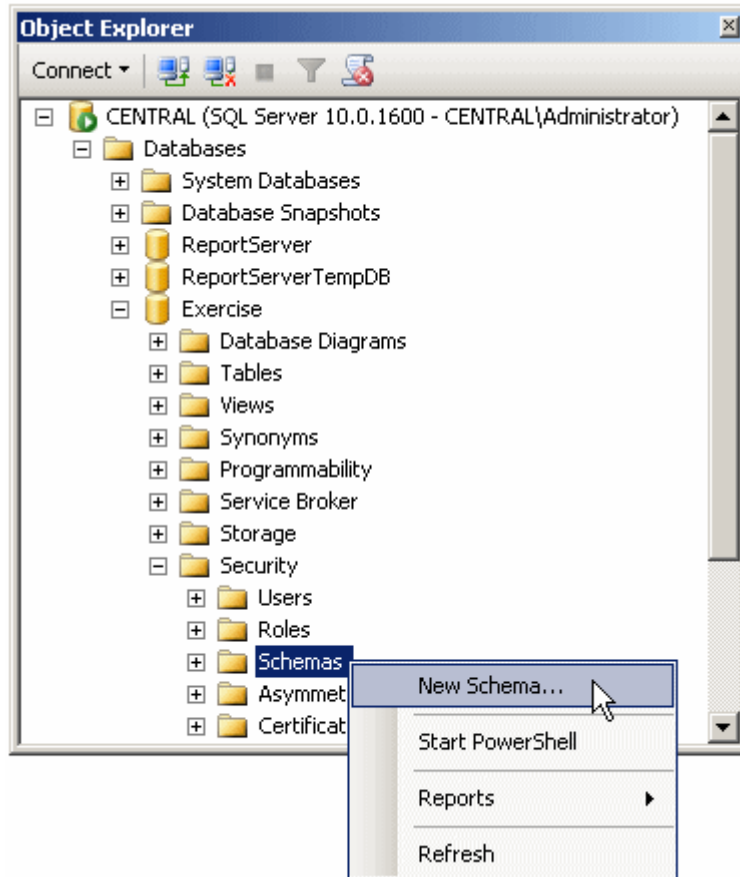
مثال: لاستعراض قواعد البيانات مع بعض خصائصها قم بتنفيذ الاستعلام الآتي:

```
select * from sys.databases
```

للوصول إلى مخطط قاعدة البيانات، في متصفح الكائنات (Object Explorer)، قم بتوسيع البند Databases، ثم بتوسيع قاعدة البيانات التي تتضمن أو تملك المخطط، ثم ابسط البند Security لتشاهد فرع المخططات (Schemas).

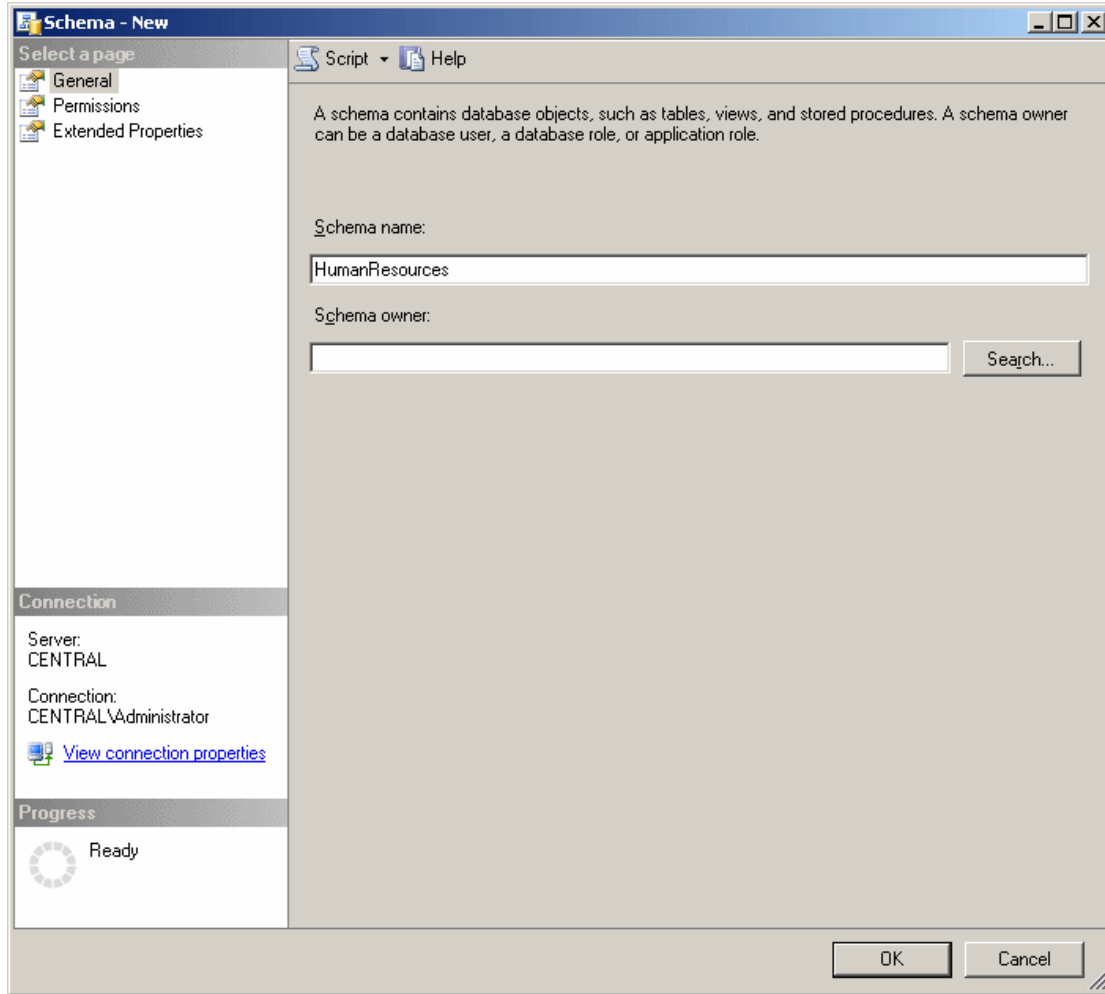
إنشاء مخطط¹:

لإنشاء مخطط انقر بالزر الأيمن على Schemas واختر New Schema...



سيظهر مربع الحوار Schema – New، في خانة النص Schema Name، أدخل اسم المخطط في لفظة واحدة، مثال:

¹ من أهم وظائف المخططات في SQL Server توسيع إمكانية إدارة الصلاحيات، حيث يمكن تحديد صلاحيات لكل Schema على حدة.



يكفي إدخال الاسم ثم النقر على OK.

سنعرض تطبيقاً عملياً إن شاء الله لإنشاء مخطط في درس لاحق.

الولوج إلى غرض من خلال مخطط:

لا يمكن اتخاذ نفس الاسم لغرضين مختلفين في المخطط، ولكن يمكن للغرض من أي مخطط أن يأخذ نفس الاسم لغرض آخر من مخطط آخر. وعلى هذا إذا كنت تريد في الوصول إلى غرض ما ضمن أي مخطط استخدم اسمه، لأن هذا الاسم سيكون فريداً، من جهة أخرى؛ ونظراً لإمكانية التعامل مع أسماء متشابهة في نفس الخادم، عند الولوج إلى غرض معين خارج المخطط يجب تمييزه، لذلك اكتب اسم المخطط الذي يحتوي على الكائن الذي ترغب في استخدامه متبوعاً برمز النقطة (.) ثم اسم الكائن (الغرض) الذي تريد استخدامه، في هذا المثال للولوج إلى الغرض Something1 الذي ينتمي إلى Schema1، نكتب:

حوصلة الدرس:

تمرين: إنشاء قاعدة بيانات لشركة مرافق

- باستخدام Microsoft SQL Server Management Studio، قم بإنشاء قاعدة بيانات اسمها UtilityCompany1 (مع قبول الإعدادات الافتراضية).
- من خلال سطر الأوامر قم بإنشاء نفس قاعدة البيانات بعد الاتصال بالخادم.

العمليات في SQL¹

العوامل والعمليات في SQL

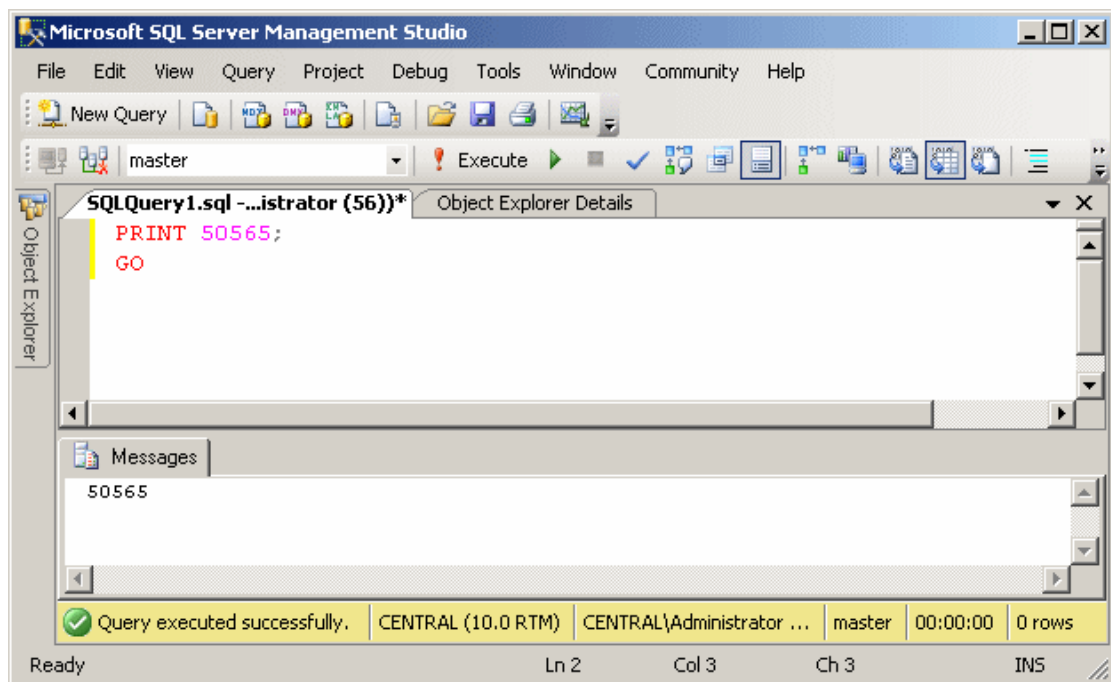
العملية هي حدث يقوم بين قيمة أو أكثر، إما لتعديل قيمة متغير أو كلا المتغيرين أو لإنتاج قيمة جديدة من خلال القيمتين السابقتين، تتم العملية على الأقل باستخدام رمز واحد وقيمة واحدة، يدعى الرمز المستخدم في العملية **بالعامل**، والقيمة المندرجة في العملية تسمى **المعامل**.

تعلية الطباعة:

مثل باقي اللغات جهزت SQL بمجموعة من العبارات التي تستخدم في تنفيذ مختلف العمليات، إحدى هذه العبارات هي أمر الطباعة: **PRINT**، لعرض أي شيء في نص عادي كنتيجة لتعليمة معينة. أكتب PRINT يليها ما تريد عرضه، وفق العبارة التالية:

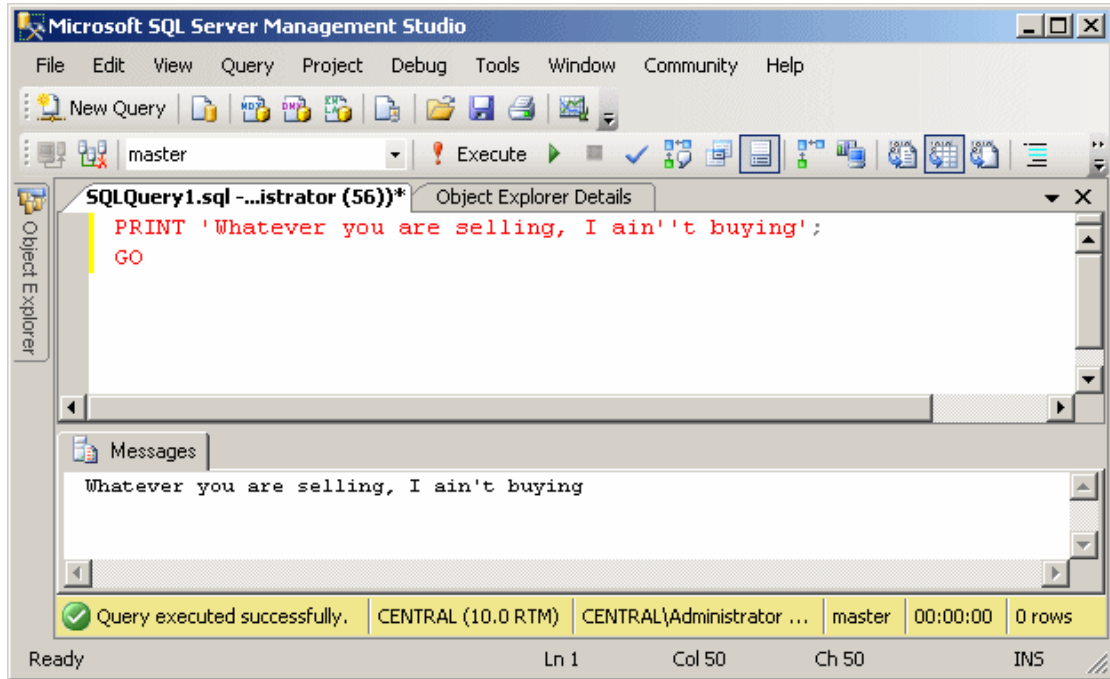
`PRINT WhatToPrint`

الغرض المطبوع يمكن أن يكون أي شيء مسموح بطباعته، ويكتب على يمين الكلمة **PRINT**، لطباعة عدد ثابت أكتبه على يمين الكلمة **PRINT** مثال:



¹ ما يقال في هذه الدروس حول لغة SQL ينطبق مع قواعد SQL Server، وليس يصلح بالضرورة مع باقي نظم قواعد البيانات.

كما يمكن طباعة أية عملية أو نتيجة عملية، إذا كنت ترغب في طباعة حرف، كلمة، أو جملة، أدرجها بين شاليتين "، وإذا كنت ترغب في أن تدرج شالة (') أكتبها مرتين، مثال:



كما يمكنك أيضا عرض مزيج من العبارات والأعداد كما سيأتي إن شاء الله.

الاستعلام SELECT:

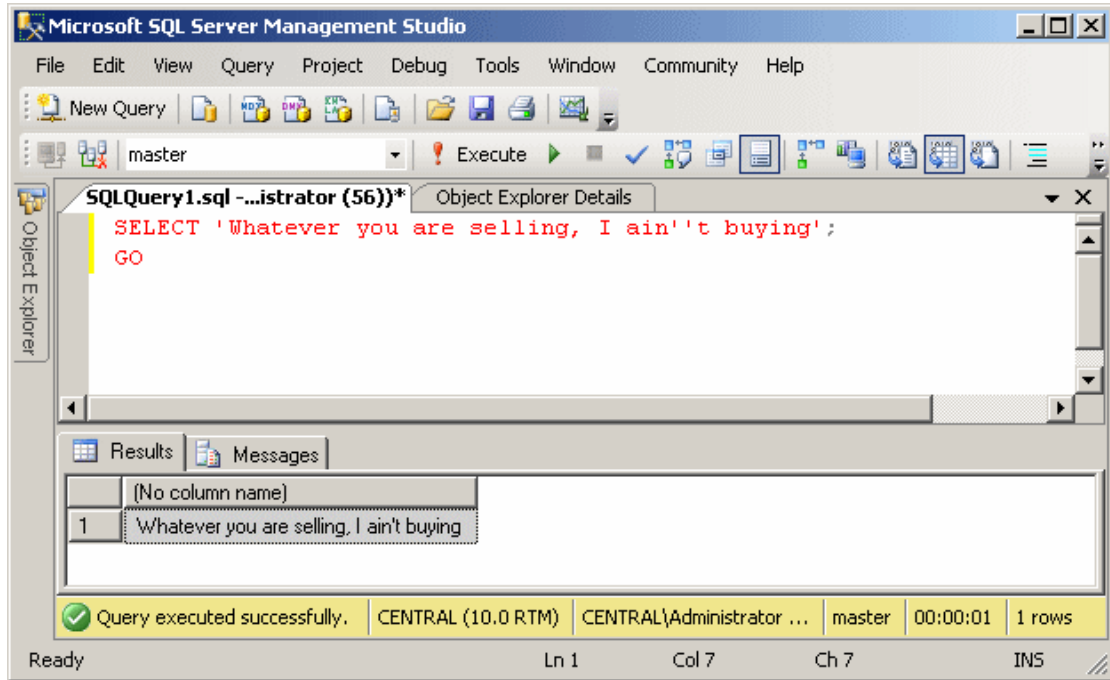
يستخدم الاستعلام SELECT مع أغراض أخرى لعرض القيم، ويستخدم النحو التالي:

SELECT What

وعلى هذا لاستخدامها اكتب الكلمة SELECT متبوعة بسلسلة حرفية، عدد، كلمة، عبارة. وينطبق هنا كل ما قلناه للأمر PRINT، والفرق بينهما:

- PRINT: يستخدم في الغالب لاختبار قيمة، سلسلة حرفية، أو عبارة، ولذلك فإنه يعرض النتائج في نافذة عادية بيضاء تحت تبويب Messages، وتستخدم PRINT مع قيمة وحيدة فقط.
- SELECT: أكثر استخدامها مع عوامل SQL، سنرى أنه يستخدم لاسترجاع السجلات من الجدول، لهذا فإنها تعرض نتائجها في نافذة منتظمة تحوي أعمدة وأسطر، تحت تبويب Results. يمكن استخدام تعليمة SELECT مع أكثر من قيمة.

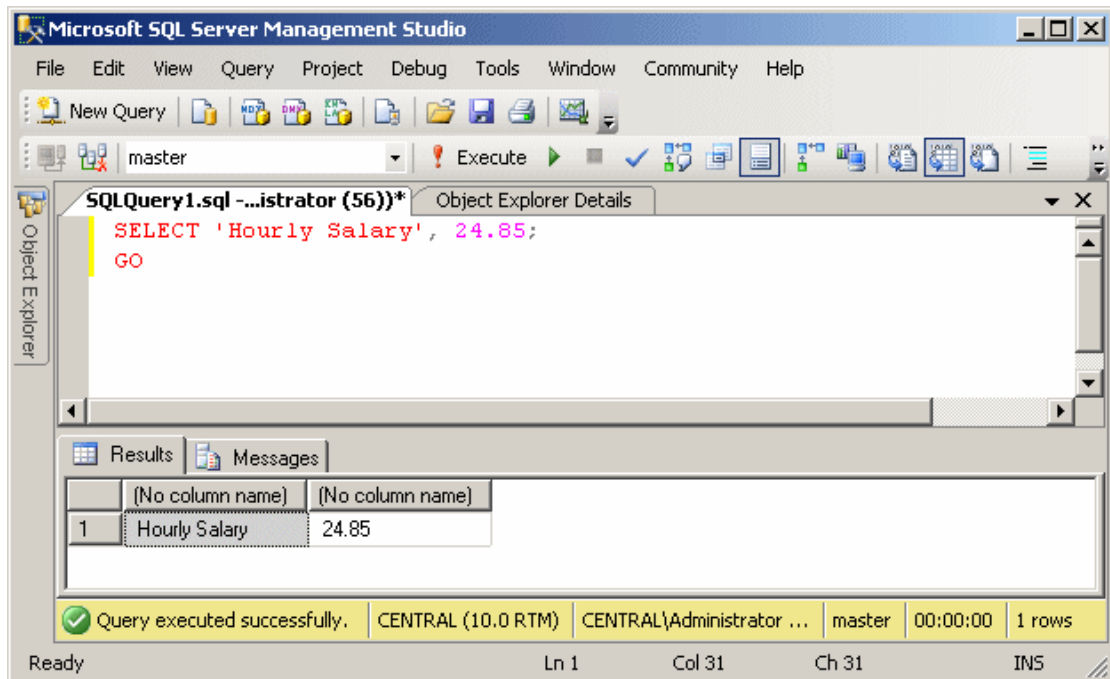
تماما مثل PRINT لعرض جملة نصية اكتبها بين شالتين على يمين العامل، مثال:



كما ذكرنا على عكس PRINT، يمكن أن تستخدم SELECT لعرض أكثر من قيمة لكن يجب الفصل بين القيم بفواصل، مثال:

```
SELECT 'Hourly Salary', 24.85;
```

ناتج الاستعلام:



التعامل مع الاستعلام SELECT:

يجب أن يكون ما على يمين تعليمة SELECT قيمة، مثال:

```
SELECT 226.75;
```

بناء على هذا، بدلا من كون الحاصل مجرد قيمة يمكن أن يكون شيئا قادر على إرجاع قيمة (تابع مثلا) سنرى ذلك لاحقا. يمكنك مثلا كتابة عبارة جبرية للاستعلام SELECT، لأننا ذكرنا أن ما على الجانب الأيمن يجب أن يعيد قيمة، كما يمكن أيضا استخدام استعلام SELECT آخر، للتمييز بين العبارتين ينبغي استعمال قوسين، مثال:

```
SELECT (SELECT 448.25);  
GO
```

عند كتابة عبارتين ينبغي أن تكون العبارات متداخلة (العبارة الثانية في العبارة الأولى).

كما يمكن أن تتداخل عدة عبارات، مثال:

```
SELECT (SELECT (SELECT 1350.75));  
GO
```

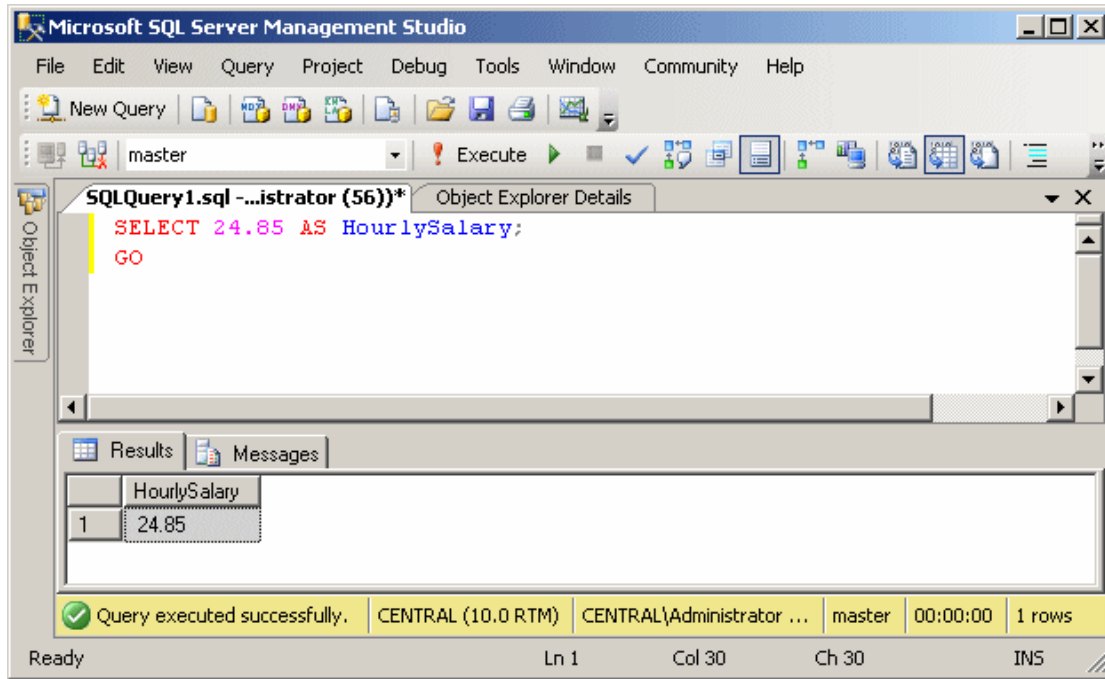
SELECT هذا AS ذلك:

رأينا طريقة عرض بيانات باستخدام تعليمة PRINT و SELECT في نافذة الاستعلام، ما يميز SELECT هو أن بإمكانها عرض النتائج في فروع مختلفة، تعرض SELECT كل قيمة في فرع يسمى العمود (Column)، يعرض كل عمود باسمه الذي يدعى العنوان (Caption)، يتم عرض العنوان بشكل افتراضي بهذا الشكل: "(No column name)".

إذا كنت ترغب باستخدام تسمية للتوضيح اكتب على يمين العبارة كلمة AS يليها العنوان المراد عرضه، العبارة المدرجة على يمين AS يجب أن تكون عبارة نصية، مثال:

```
SELECT 24.85 AS HourlySalary;
```

وهذا من شأنه أن ينتج:



يمكن إدراج العبارة التي بعد AS بين شاليتين "، مثال:

```
SELECT 24.85 AS 'HourlySalary';
```

إذا كانت العبارة على يمين AS مكونة من كلمات مختلفة بينها فراغات، يجب إدراجها

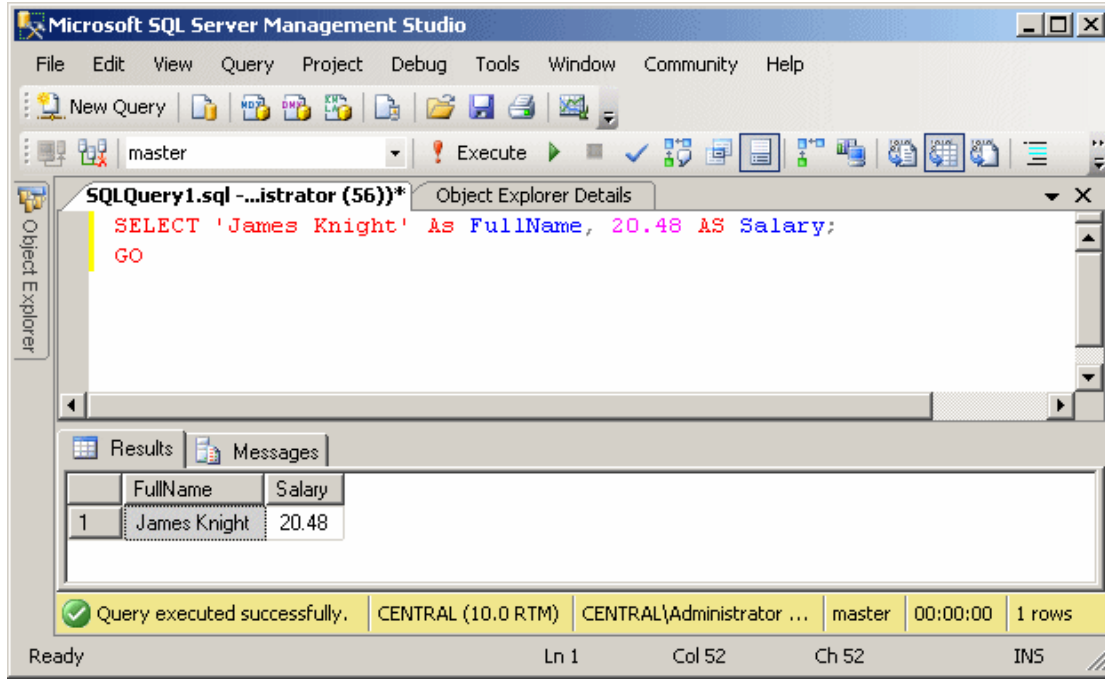
بين شاليتين أو عارضتين [], مثال:

```
SELECT 24.85 AS 'Hourly Salary';
SELECT 24.85 AS [Hourly Salary];
```

إذا كنت تريد عرض قيم مختلفة افصل بينهما بفاصلة، يمكنك إتباعها بالكلمة AS، مثال:

```
SELECT 'James Knight' As FullName, 20.48 AS Salary;
```

والنتيجة بهذا الشكل:



يمكن كتابة الاستعلام أعلاه على النحو التالي:

```
SELECT 'James Knight' As [Full Name], 20.48 AS [Hourly Salary];
```

العمليات الأحادية (Unary operator):

علامة الجمع (+) وعلامة الطرح (-):

العمليات الأحادية هي العمليات التي تطبق على عامل واحد فقط، من المعلوم في علم الجبر أن العدد الصحيح سالب أو موجب أو معدوم، للتعبير عن العدد الموجب يمكنك كتابة رمز علامة الجمع على يمين العدد (وليس على يساره) أو عدم كتابة أي علامة (ليصبح Unsigned)، وللتعبير عن العدد السالب أضف إليه رمز علامة الطرح على اليمين.

```
PRINT +1250
SELECT -1250
```

العمليات الثنائية (Binary Operators):

وهي العمليات التي تطبق على عاملين:



تدعم SQL الجمع والطرح والضرب بخصائصها (التجميع والتبديل...) بين أي نوع من الأعداد (طبيعية، عشرية...) بأي عدد من العوامل (عاملين أو أكثر) بالطريقة المعروفة كباقي لغات البرمجة، وتدعم الجمع بين العبارات النصية بشكل مباشر مثل:

```
PRINT 'Henry ' + 'Kono'
```

كبقية لغات البرمجة تستخدم SQL الأقواس لفصل العمليات، فمثلا العبارتين:

```
PRINT (154 - 12) + 8  
PRINT 154 - (12 + 8)
```

غير متساويتين، الأولى تعيد 150 والثانية 134.

حاصل القسمة باستخدام الرمز / مع أعداد صحيحة لا يعيد الجزء العشري، مثلا العبارة:

```
SELECT 7 / 4
```

تعيد القيمة 2 دون إجراء تدوير، أي التقريب بالنقصان.

حاصل القسمة باستخدام الرمز / مع أعداد عشرية يعيد الجزء العشري، مثال:

```
SELECT 7 / 4.0
```

يعيد النتيجة 1.750000 كاملة.

رمز المستخدم للعدد العشري هو النقطة (.) ورمز الفاصلة (,) مستعمل للفصل بين النتائج في أعمدة.

باقي القسمة أو يسمى Modulo يرمز له في لغة SQL برمز النسبة المئوية %، مثال:

```
PRINT 128 % 42
```

يعيد هذا المثال النتيجة 2، ويمكن كتابة عبارات أكثر تعقيدا:

```
SELECT  
  (SELECT 448.25 * 3) +  
  (SELECT 82.28 - 36.04);  
GO
```

العمليات المنطقية:

تحجز القيم المستعملة في قاعدة البيانات أو التطبيقات مساحة معينة من الذاكرة، يمكن تمثيل المساحة المحجوزة بمجموعة من الصناديق الصغيرة تدعى Bits، فمثلا يخزن الحرف B في 7 سبعة خانات (Bits) مع أنه حرف واحدة فقط.

تسمح عمليات المنطق بالتلاعب في كيفية تخزين القيم في Bits. لن تحتاج لهذه العملية دائما خاصة في المراحل الأولى من تصميم قاعدة البيانات، ومع ذلك فإن العمليات على Bits (وما يتعلق بها من عمليات) موجودة في جميع أو معظم بيئات البرمجة، لذلك يجب أن تكون على دراية بما تقوم به أو ما تقدمه.

العمليات المنطقية: عملية النفي NOT الرمز ~

من العمليات التي يمكنك القيام بها على الخانات عكس القيمة، إذا كانت الخانة تحمل القيمة 1 فتصبح 0 والعكس بالعكس، تتم هذه العملية باستخدام معامل النفي NOT ورمزه ~.

المعامل NOT أحادي التشغيل أي يوضع على الجانب الأيسر من العبارة:

~Value

مثال:

PRINT ~158

للقيام بهذه العملية، يقوم مترجم Transact-SQL باعتبار كل خانة Bit طرف من العملية ثم عكس قيمة كل Bit، من 1 إلى 0 أو 0 إلى 1 حسب قيمة الخانة.

مثال: باعتبار العدد مكون من أرقام byte مثل 248، يعبر عن القيمة العشرية 248 بالتعداد الثنائي 11111000 (وفي النظام الست-عشري بالقيمة XF80)، إذا طبقنا على هذه القيمة معامل النفي NOT لعكس قيم كل bit فسنحصل على النتيجة التالية:

Value	1	1	1	1	1	0	0	0
~Value	0	0	0	0	0	1	1	1

معامل المقارنة AND الرمز &

عملية المقارنة (رمزها &) هي ثنائية التشغيل وتستخدم النحو التالي:

Operand1 & Operand2

يطبق هذا المشغل بين قيمتين ويقارن بين كل bit بالمقابلة مع كل bit من القيمة الأخرى، إذا كان قيمة الخانتين 1 فالنتيجة 1، غير ذلك فالنتيجة 0، يمكن تخيص النتائج على النحو التالي:

Bit1	Bit2	Bit1 & Bit2
0	0	0
1	0	0
0	1	0
1	1	1

لدينا قيمتين 187 و 242، تكتبان في النظام الثنائي على الشكل 10111011 (في النظام الست-عشري XBB0). المترجمة بقيمة 242 عشري هو 11110010 (في النظام الست-عشري XF20)، سنقارن بين هاتين القيمتين باستخدام معام AND :

	Binary								Decimal
N1	1	0	1	1	1	0	1	1	187
N2	1	1	1	1	0	0	1	0	242
N1 & N2	1	0	1	1	0	0	1	0	178

قد ترغب أن يقوم المترجم بأداء هذه العملية لتستخدمها في برنامجك، وحينها يمكنك الحصول على النتيجة باستخدام هذه العملية، يمكن عرض العملية أعلاه كالتالي:

`PRINT 187 & 242`

وهذا من شأنه أن يعيد القيمة 178.

معامل المقارنة OR الرمز |

يمكنك القيام بنوع آخر من المقارنة على أي bit باستخدام الرمز |، وصيغته:

`Value1 | Value2`

أيضا، يقارن المترجم بين كل bit لكلا القيمتين، بحيث يكون الناتج 0 إذا كانت قيمة كل bit تساوي 0، غير ذلك يعيد القيمة 1، والجدول التالي يلخص النتائج:

Bit1	Bit2	Bit1 Bit2
0	0	0
1	0	1
0	1	1
1	1	1

لدينا القيمتين العشريتين 187 و 242، يعيد المعامل OR النتائج التالية:

	Binary								Decimal
N1	1	0	1	1	1	0	1	1	187
N2	1	1	1	1	0	0	1	0	242
N1 N2	1	1	1	1	1	0	1	1	251

كما يمكنك أن تكلف المترجم بالقيام بأداء العملية لتستعيد النتائج، مثال:

`PRINT 187 | 242`

نتيجة العملية 251.

معامل المقارنة XOR الرمز ^

مثل العمليتين السابقتين XOR للمقارنة بين قيمتين، وصيغتها:

`Value1 ^ Value2`

يقارن المترجم بين كل bit لكل قيمة بالنسبة للأخرى، فإذا كانت قيمة الخانتين متساويتين

(0 أو 1) فالنتيجة 0، ما عدا ذلك النتيجة 1، خلاصة العملية على النحو التالي:

Bit1	Bit2	Bit1 ^ Bit2
0	0	0
1	0	1
0	1	1
1	1	0

مرة أخرى نأخذ العددين العشريين 187 و 242، بتطبيق معامل XOR نحصل على:

	Binary								Decimal
N1	1	0	1	1	1	0	1	1	187
N2	1	1	1	1	0	0	1	0	242
N1 ^ N2	0	1	0	0	1	0	0	1	73

مثال تطبيق العملية:

`PRINT 187 ^ 242;`

والناتج هو القيمة 73.

حوصلة الدرس

تمرين: قاعدة بيانات شركة مرافق

- فتح نافذة الاستعلام من قاعدة البيانات UtilityCompany1

- إنشاء وتنفيذ العملية التي تحسب تكلفة 224 كيلو واط ساعي، حيث: 0.0900 دولار لكل kWh.
- إنشاء وتنفيذ العملية التي تضيف إلى 8.50 دولار إلى نتيجة جداء 148 كيلو واط ساعي إلى 0.0650 kWh.

المتغيرات وأنواع البيانات

المتغيرات الأساسية

استخدمنا في الدرس السابق قسم لبعض الأنواع من المتغيرات مثل 242 أو ' James Knight، هذه الأنواع من القيم تعتبر ثابتة (Constant) لأننا نعرفها قبل استعمالها ولم نغير في محتواها، إذا كنت تريد استخدام نوع معين من القيم مرارا وتكرارا يمكنك حجز جزء من الذاكرة لتلك القيمة، وبذلك يمكنك تغيير هذه قيمة باستمرار وبسهولة ومتى شئت. يتم ذلك بحجز قيمة في منطقة من الذاكرة حيث تم تخزين قيمة المتغير.

لاستخدام نفس المنطقة من الذاكرة لتخزين وحذف القيم حسب الحاجة، يحتاج مترجم SQL معلومتين أساسيتين: الاسم وحجم المساحة من الذاكرة القادر على تخزين القيمة.

الإعلان عن المتغيرات

المتغير هو حيز من الذاكرة يستخدم لتخزين القيم التي تستخدم في البرنامج، قبل استخدام متغير يجب عليك إبلاغ المترجم بوجود المتغير، وهذا ما يعبر عنه بالإعلان عن المتغير. للإعلان عن متغير في SQL Server استخدم كلمة **DECLARE** وفق الصيغة التالية:

DECLARE Options

الكلمة **DECLARE** تعلم المترجم بإعلان عن متغير، يتبعها اسم المتغير، يقوم Transact-SQL بالتعرف على اسم المتغير بالعلامة @، ويسمح اسم المتغير بتحديد مساحة الذاكرة حيث تخزن قيمة المتغير.

بخلاف لغات البرمجة الأخرى C/C++, Pascal, Java, C# التي تفرض قواعد صارمة لأسماء المتغيرات؛ Transact-SQL مرنة للغاية حيث يمكن للاسم أن يأخذ أرقاما فقط، مثال:

DECLARE @264

لكن هذا الاسم يمكن أن يتداخل مع عدد طبيعي.

يمكن أن يأخذ اسم المتغير عدة كلمات كما يمكن أيضا أن يكون في كلمة واحدة.

أسماء المتغيرات:

لتجنب الخلط سنراعي في هذه الدروس ما يلي:

- يبدأ اسم المتغير بالعلامة _ أو بحرف لاتيني، مثال: @_n ، @act ، @Second .
- بعد الحرف الأول يأخذ اسم المتغير تركيبة من حروف أو أرقام أو العلامة _ ، مثال: @_24 ، @act_52t .
- لن يأخذ اسم المتغير أيًا من هذه الرموز: !, @, #, \$, %, ^, &, * .
- إذا كان اسم المتغير في أكثر من كلمة فسيكتب أول حرف من كل كلمة بالأحرف الكبيرة (Upper Case)، مثال: @DateHired, @_RealSport, @DriversLicenseNumber .

للإعلان عن متغير، بعد إعطائه اسماً مناسباً، يجب تحديد مساحة الذاكرة التي يحتاجها المتغير، وهو ما يعرف بنوع البيانات، ويتم ذلك وفق الصيغة التالية:

```
DECLARE @VariableName DataType;
```

يمكنك أيضاً الإعلان عن أكثر من متغير واحد، للقيام بذلك أضف رمز الفاصلة بين المتغيرات على النحو التالي:

```
DECLARE @Variable1 DataType1, @Variable2 DataType2, @Variable_n  
DataType_n;
```

وخلافاً للعديد من لغات البرمجة الأخرى، إذا أعلنت عن العديد من المتغيرات التي تستخدم نفس نوع البيانات يجب أن يتبع اسم كل متغير نوع البيانات الخاصة به.

تهيئة متغير (Initializing a Variable)

بعد الإعلان عن متغير يقوم المترجم بحجز مساحة من ذاكرة الكمبيوتر لذلك المتغير، ولكن المساحة المحجوزة لا تكون كافية بالضرورة، ففي هذا الوقت يكون المتغير عديم القيمة (Null)، لتغيير ذلك يجب إرفاق قيمة ابتدائية للمتغير، وهذا ما يعرف بتهيئة المتغير.

تذكر أن اسم المتغير يبدأ بالرمز @ وكلما أردت الإشارة إلى المتغير يجب عليك أن تتأكد من إدراج العلامة @، لتهيئة المتغير -متى استلزم الأمر- استخدم الكلمة SELECT أو SET متبوعة باسم المتغير ثم رمز إرفاق القيمة "=" تليها قيمة مناسبة، والصيغة:

```
SELECT @VariableName = DesiredValue
```

أو

```
SET @VariableName = DesiredValue
```

بمجرد تهيئة متغير يمكنك عرض قيمته بكتابة اسم المتغير على يمين الأمر SELECT أو .PRINT

أنواع البيانات:

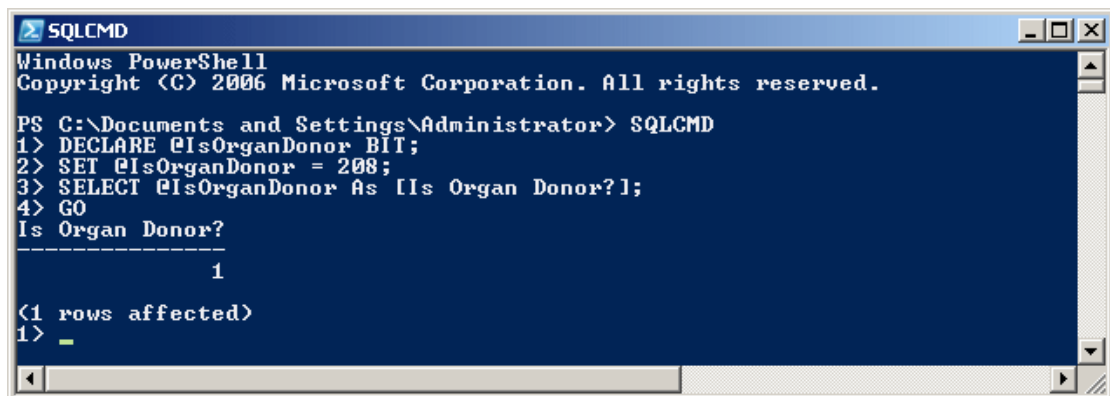
بعد تحديد اسم متغير يجب أن تحدد مقدار الذاكرة الذي يستخدم لتخزين قيمة المتغير، نظرا لوجود أنواع مختلفة من المعلومات في قاعدة البيانات التي يمكن التعامل معها، توفر SQL مجموعة هامة من أنواع البيانات.

المتغيرات المنطقية:

القيمة المنطقية هي المعلومة التي يمكن كتابتها على وجهين متناظرين: صحيح/خاطئ، نعم/لا، 1/0...الخ، للإعلان عن متغير منطقي يستخدم النوع bit كما يلي:

```
DECLARE @IsOrganDonor bit;
```

بعد الإعلان عن المتغير المنطقي، يمكنك تهيئته بقيمة 0 أو قيمة أخرى، إذا كانت قيمة المتغير عند التهيئة 0 فسيتلقى القيمة المنطقية خاطئ (False)، وإذا تلقى عدد آخر فإنه سيحمل القيمة المنطقية صحيح (True)، مثال¹:



```
SQLCMD
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Documents and Settings\Administrator> SQLCMD
1> DECLARE @IsOrganDonor BIT;
2> SET @IsOrganDonor = 208;
3> SELECT @IsOrganDonor As [Is Organ Donor?];
4> GO
Is Organ Donor?
-----
1
<1 rows affected>
1> -
```

تطبيق عملي: استخدام متغير منطقي

1- افتح نافذة استعلام جديدة واكتب ما يلي:

```
DECLARE @IsMarried bit
SET @IsMarried = 1
```

¹ في هذا المثال تم استخدام SQLCMD برنامج للتعامل مع SQL Server من خلال سطر الأوامر باستخدام PowerShell.


```
SELECT @IsMarried AS [Is Married?];  
GO
```

2- نفذ الاستعلام.

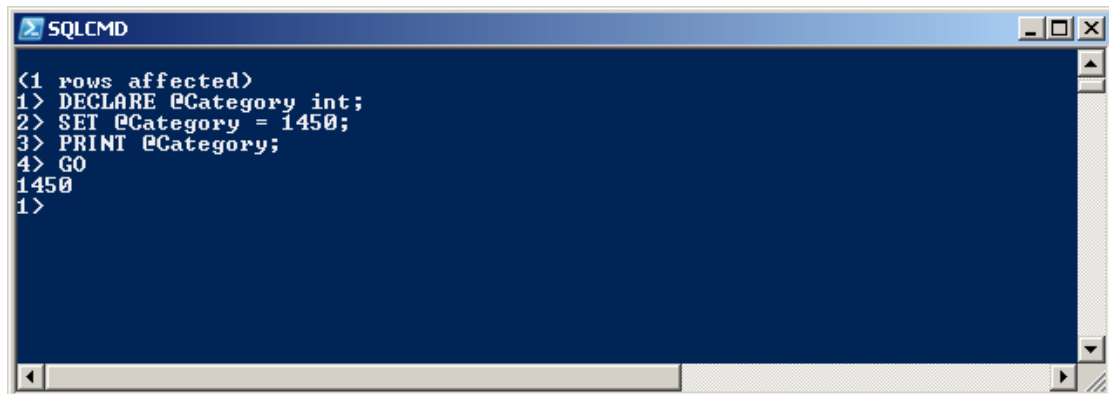
المتغير العددي Integer:

النوع Integer هو مجموعة الأعداد الصحيحة المحصورة بين القيمتين -2,147,483,648 و 2,147,483,648 ، أي أنها تقبل قيم سالبة أيضا ولا تقبل قيم عشرية، وبالتالي لا يمكن إضافة رمز الفاصلة بين الأرقام كما لا يمكن إضافة رمز تمييز الآلاف، مثلا العدد: 79,435,794 غير مقبول في SQL لأنه يحمل فواصل بين الأرقام.

يتم الإعلان عن متغير Integer في SQL Server بالرمز **int**، مثال:

```
DECLARE @Category int;  
SET @Category = 1450;  
PRINT @Category;  
GO
```

وهذا من شأنه أن يعرض القيمة 1450:



```
SQLCMD  
<1 rows affected>  
1> DECLARE @Category int;  
2> SET @Category = 1450;  
3> PRINT @Category;  
4> GO  
1450  
1>
```

يحجز النوع int مساحة 4 bytes.

إذا كنت ترغب في استخدام متغير لحجز أعداد صغيرة جدا لأعمار الأشخاص مثلا، أو عدد صفحات جريدة، استخدم نوع البيانات **tinyint**، نوع البيانات **tinyint** يمكن أن يخزن عدد موجب تتراوح قيمته بين 0 إلى 255، مثال:

```

SQLCMD
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Documents and Settings\Administrator> SQLCMD
1> DECLARE @StudentAge tinyint;
2> SET @StudentAge = 14;
3> SELECT @StudentAge AS [Student's Age];
4> GO
Student's Age
-----
          14

<1 rows affected>
1> -

```

نوع البيانات **smallint** يتبع نفس القواعد والمبادئ لنوع البيانات **int** إلا أنه يستخدم لتخزين أعداد صغيرة تتراوح قيمها بين 32.767 و -32.768، مثال:

```

DECLARE @NumberOfPages SMALLINT;
SET @NumberOfPages = 16;
SELECT @NumberOfPages AS [Number of Pages];
GO

```

نوع البيانات **bigint** يتبع نفس القواعد والمبادئ للنوع **int** إلا أنه يستخدم مع أعداد كبيرة جداً؛ من -9.223.372.036.854.775.808 إلى 9.223.372.036.854.775.807، مثال:

```

DECLARE @CountryPopulation BigInt;
SET @CountryPopulation = 16500000;
SELECT @CountryPopulation AS 'Country Population';
GO

```

نوع البيانات الثنائية (binary data) يستخدم مع المتغيرات التي تحمل الأعداد الست-عشرية (hexadecimal)، مثل 0xFA36, 0x7238، استخدم نوع البيانات الثنائي إذا كانت لجميع قيم المتغير نفس الحجم (أو الكمية)، إذا كنت تتوقع أن بعض الإدخالات ستكون مختلفة عن غيرها، استخدم نوع البيانات **varbinary** كبديل، إن النوع **varbinary** يستخدم أيضاً للبيانات الست-عشرية لكنه يسمح بإدخال قيم متباينة ما دامت كل القيم ست-عشرية.

تطبيق عملي: استخدام المتغيرات Integer

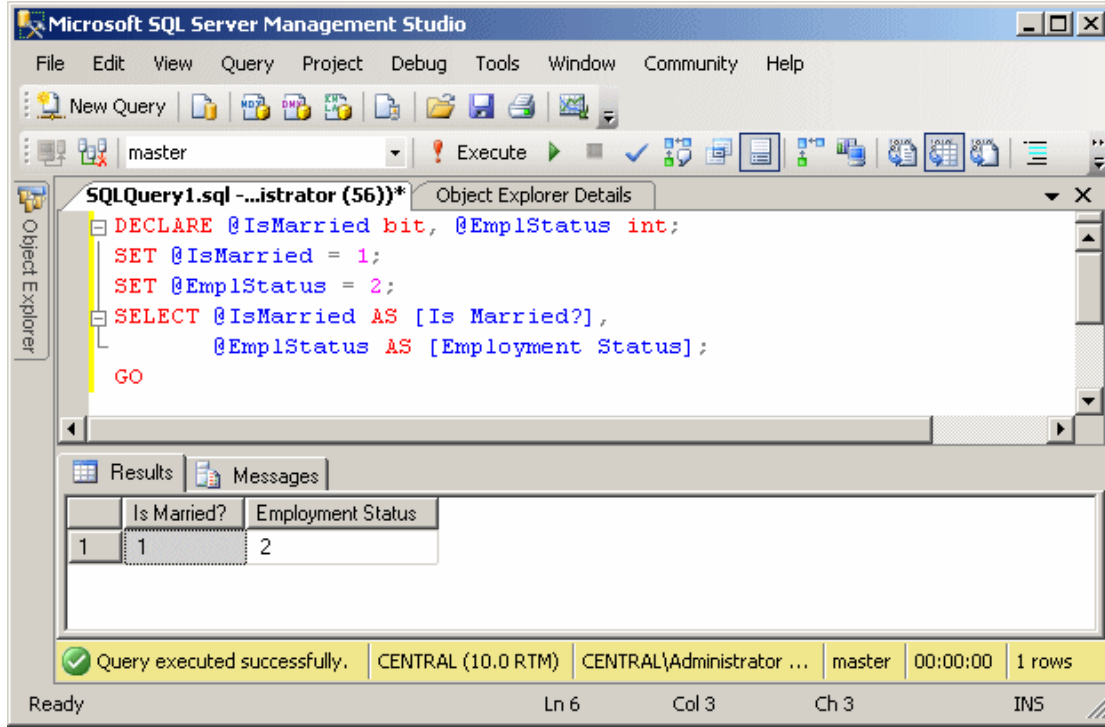
1- في نافذة استعلام جديدة اكتب استعلاماً على النحو التالي:

```

DECLARE @IsMarried bit, @EmplStatus int;
SET @IsMarried = 1;
SET @EmplStatus = 2;
SELECT @IsMarried AS [Is Married?],
       @EmplStatus AS [Employment Status];
GO

```

2- نفذ الاستعلام (F5)



متغيرات الأعداد العشرية (Decimal):

العدد العشري هو العدد الذي يمكن أن يتضمن فاصلة، (أو الرمز الذي يستخدم كفاصلة عشرية حسب ما هو محدد في الإعدادات الإقليمية للنظام من Control Panel) بين الرقمين، مثل 12.625 و 44.80، تشمل هذه الأعداد قيم سالبة وموجبة. يمكن التحكم في عدد الأرقام العشرية، كما يمكن تحديد النوع إما **numeric** أو **decimal**، مثال:

```
DECLARE @Distance DECIMAL;
SET @Distance = 648.16;
PRINT @Distance;
GO
```

سيعيد هذا الاستعلام النتيجة 648 فقط، ذلك لأننا لم نحدد مقدار الجزء العشري، لتحديد الجزء العشري نضيف رقمين بين قوسين بعد كتابة اسم النوع، الرقم الأول يمثل حجم العدد (مثلا العدد 648.16 حجمه 5 أرقام)، والثاني يمثل مقدار الجزء العشري (مثلا العدد 648.16 مقدار جزئه العشري رقمين)، مثال:

```
DECLARE @Distance decimal(5,2);
SET @Distance = 648.12
SELECT @Distance;
GO
```

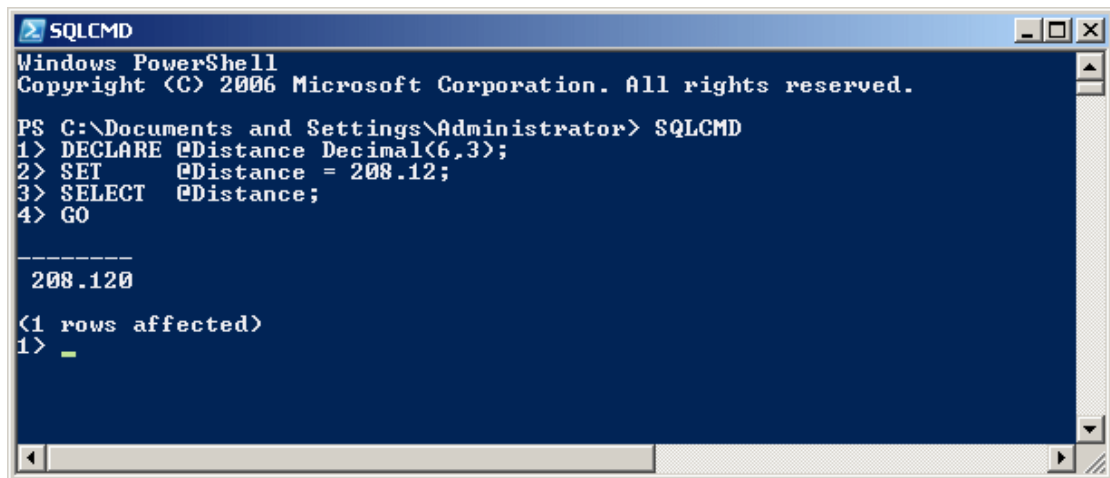
هذا الاستعلام يعيد النتيجة 648.12.

الأعداد ذات الفاصلة العائمة (floating-point) هي الأعداد ذات قيم كسرية، مثلها مثل الأعداد العشرية decimal، للإعلان عن مثل هذا المتغير استخدم الكلمة **Float**، مثال:

```
DECLARE @Radius FLOAT;  
SET @Radius = 48.16;  
SELECT @Radius AS Radius;  
GO
```

دقة العدد هي عدد الأرقام المستخدمة لعرض الجزء الصحيح، مثلا العدد 42005 دقته 5؛ والعدد 226 دقته 3، إذا كان نوع البيانات محدد على هيئة متغير عددي (int وفرعه) أو أعداد ذات فاصلة عائمة (float و real) فالدقة تحدد من قبل قاعدة البيانات ويمكنك فقط قبول القيمة التي حددها مترجم SQL Server، للرقم العشري (decimal كان أم numeric) يسمح لك SQL Server بتحديد المقدار الذي تريد من الدقة، لكن القيمة يجب أن يكون بين 1 و 38 (28 إذا كنت تستخدم SQL Server 7).

العدد العشري هو العدد التي يحوي قيم بعد الفاصلة، مثل 12.05 و 1450.4227، حجم الجزء العشري هو عدد الأرقام على يمين العدد، حجم الجزء العشري يستخدم فقط مع الأعداد التي تقبل أرقام بعد الفاصلة، وتشمل النوع النقدي (money و smallmoney) والعشرية (numeric و decimal)، إذا أعلنت عن متغير عشري يمكنك تحديد الجزء العشري منه، بحيث لا يزيد عن 18 ولا يقل عن 0، مثال:



```
SQLCMD  
Windows PowerShell  
Copyright (C) 2006 Microsoft Corporation. All rights reserved.  
  
PS C:\Documents and Settings\Administrator> SQLCMD  
1> DECLARE @Distance Decimal(6,3);  
2> SET @Distance = 208.12;  
3> SELECT @Distance;  
4> GO  
  
-----  
208.120  
  
<1 rows affected>  
1> -
```

تطبيق عملي: استخدام المتغيرات العشرية

1- غير الاستعلام إلى ما يلي:

```
DECLARE @IsMarried bit,  
        @EmplStatus int,  
        @WeeklyHours Decimal(6,2);  
SET @IsMarried = 1;  
SET @EmplStatus = 2;  
SET @WeeklyHours = 36.50;  
SELECT @IsMarried AS [Is Married?],  
       @EmplStatus AS [Employment Status],  
       @WeeklyHours AS Hours;  
GO
```

2- نفذ الاستعلام.

متغيرات العملة النقدية:

إذا كان المتغير يحمل قيمة نقدية يمكنك الإعلان عنه باستخدام الكلمة **money**، يدعم النوع **money** القيم السالبة والموجبة، من -922,337,203,685,477.5808 إلى +922,337,203,685,477.5807، مثال:

```
DECLARE @YearlyIncome Money;  
SET @YearlyIncome = 48500.15;  
SELECT @YearlyIncome AS [Yearly Income];  
GO
```

نوع البيانات **smallmoney** مثله مثل النوع **money** ويستخدم مع القيم التي لا يمكن أن تكون أقل من -214,748.3648 ولا أكبر من 214,748.3647.

دقة النوع **money** و **smallmoney** يحددها SQL Server في 4 أرقام.

تطبيق عملي: استخدام نوع العملة النقدية

1- أكتب الاستعلام التالي:

```
DECLARE @EmplStatus int,  
        @IsMarried bit,  
        @WeeklyHours Decimal(6,2),  
        @HourlySalary SmallMoney,  
        @WeeklySalary SmallMoney;  
SET @IsMarried = 1;  
SET @EmplStatus = 2;  
SET @WeeklyHours = 36.50;  
SET @HourlySalary = 15.72;  
SET @WeeklySalary = @WeeklyHours * @HourlySalary;  
SELECT @EmplStatus AS [Empl Status],  
       @IsMarried AS [Married?],  
       @WeeklyHours AS Hours,  
       @HourlySalary AS Hourly,
```

@WeeklySalary AS Weekly;

GO

2- نفذ الاستعلام.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query in a text editor. The query is as follows:

```
DECLARE @EmplStatus int,
        @IsMarried bit,
        @WeeklyHours Decimal(6,2),
        @HourlySalary SmallMoney,
        @WeeklySalary SmallMoney;

SET @IsMarried = 1;
SET @EmplStatus = 2;
SET @WeeklyHours = 36.50;
SET @HourlySalary = 15.72;
SET @WeeklySalary = @WeeklyHours * @HourlySalary;

SELECT @EmplStatus AS [Empl Status],
       @IsMarried AS [Married?],
       @WeeklyHours AS Hours,
       @HourlySalary AS Hourly,
       @WeeklySalary AS Weekly;

GO
```

Below the query editor, the Results pane shows a single row of data:

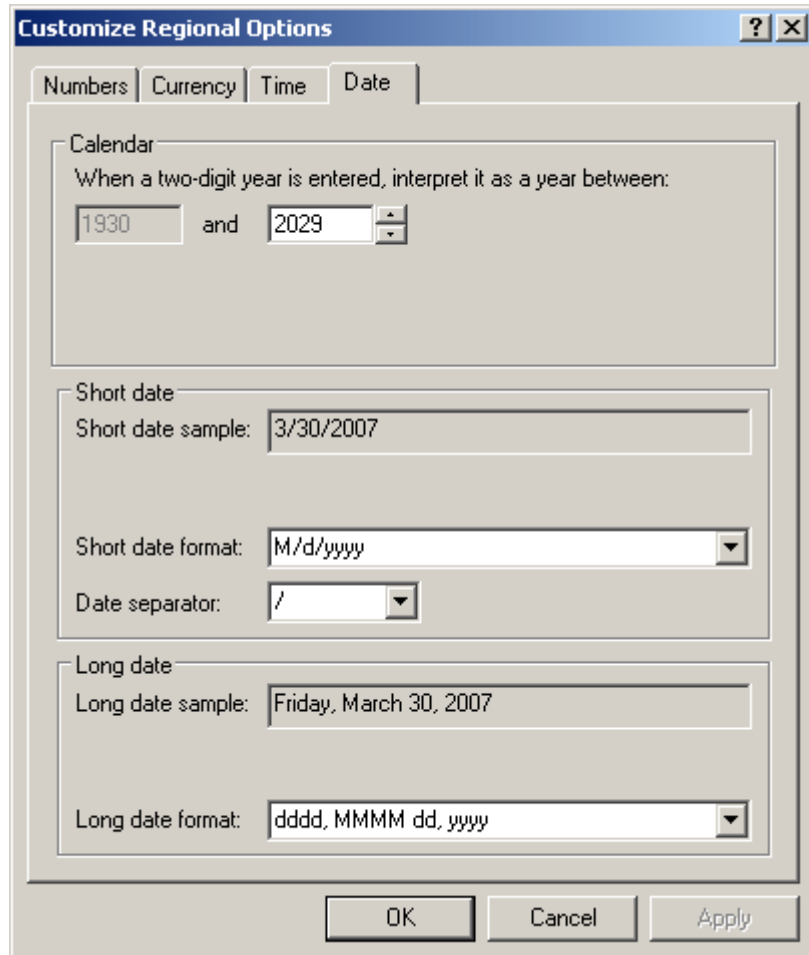
	Empl Status	Married?	Hours	Hourly	Weekly
1	2	1	36.50	15.72	573.78

The status bar at the bottom indicates: Query executed successfully. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 1 rows

المتغيرات الزمنية (التاريخ والتوقيت):

لتخزين قيم تحمل تاريخ أو/و توقيت يستخدم نوع البيانات DATETIME، ولا بد أن تكون المدخلات من وقت أو تاريخ ذات صيغة صحيحة وهنا يتميز SQL Server بأكثير من المرونة، حيث يمكن عرض التاريخ في أشكال غير التقليدية. يقبل النوع datetime القيم المحصورة بين 1 كانون الثاني / يناير من عام 1753 و 31 كانون الأول / ديسمبر من سنة 9999.

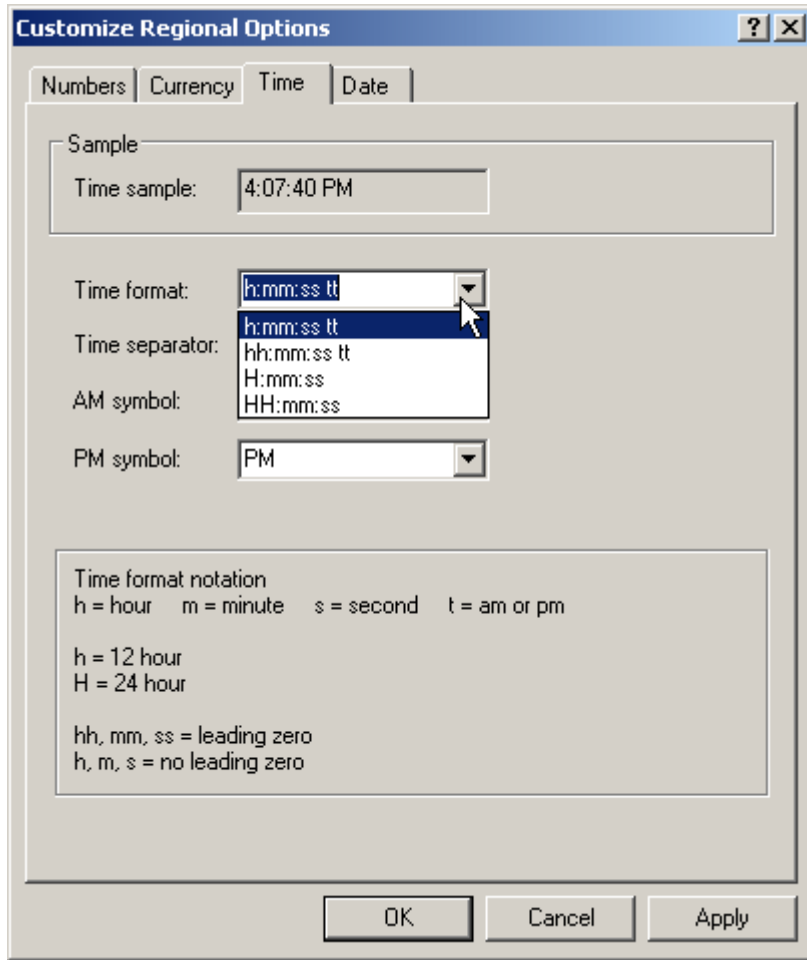
لتهيئة قيمة متغير من النوع DATETIME ضع قيمته بين علامتي الاقتباس (")، لعرض قيمة تاريخية افصل مكونات القيمة بالرمز المعترف به (Date Separator) في لوحة التحكم (Control Panel):



مثال:

```
DECLARE @IndependenceDay DATETIME;
SET @IndependenceDay = '01/01/1960';
SELECT @IndependenceDay AS [Independence Day];
GO
```

إذا كانت قيمة المتغير تمثل قيمة زمنية فيجب إدراجها بين علامتي الاقتباس، وبين علامتي الاقتباس اكتب القيمة الزمنية وفق الإعدادات الإقليمية في Control Panel:



مثال:

```
DECLARE @ArrivalTime datetime;
SET @ArrivalTime = '18:22';
SELECT @ArrivalTime AS [Arrival Time];
GO
```

نوع البيانات **smalldatetime** نوع بديل عن **datetime**، وهو يتبع نفس القواعد والمبادئ إلا أن التاريخ المقبول في هذا النوع محصور بين 1 من كانون الثاني / يناير لعام 1900 و 6 يونيو / جوان لعام 2079.

تطبيق عملي: استخدام متغيرات التاريخ / الوقت

1- اكتب الاستعلام التالي:

```
DECLARE @DateHired DateTime,
        @EmplStatus int,
        @IsMarried bit,
        @WeeklyHours Decimal(6,2),
        @HourlySalary SmallMoney,
        @WeeklySalary SmallMoney;
SET @DateHired = '12/05/1998';
```



```

SET @IsMarried = 1;
SET @EmplStatus = 2;
SET @WeeklyHours = 36.50;
SET @HourlySalary = 15.72;
SET @WeeklySalary = @WeeklyHours * @HourlySalary;
SELECT @DateHired AS [Date Hired],
       @EmplStatus AS [Empl Status],
       @IsMarried AS [Married?],
       @WeeklyHours AS Hours,
       @HourlySalary AS Hourly,
       @WeeklySalary AS Weekly;

```

GO

1- نفذ الاستعلام.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query with the following code:

```

DECLARE @DateHired DateTime,
        @EmplStatus int,
        @IsMarried bit,
        @WeeklyHours Decimal(6,2),
        @HourlySalary SmallMoney,
        @WeeklySalary SmallMoney;

SET @DateHired = '12/05/1998';
SET @IsMarried = 1;
SET @EmplStatus = 2;
SET @WeeklyHours = 36.50;
SET @HourlySalary = 15.72;
SET @WeeklySalary = @WeeklyHours * @HourlySalary;

SELECT @DateHired AS [Date Hired],
       @EmplStatus AS [Empl Status],
       @IsMarried AS [Married?],
       @WeeklyHours AS Hours,
       @HourlySalary AS Hourly,
       @WeeklySalary AS Weekly;

GO

```

Below the query editor, the Results pane shows a single row of data:

	Date Hired	Empl Status	Married?	Hours	Hourly	Weekly
1	1998-12-05 00:00:00.000	2	1	36.50	15.72	573.78

The status bar at the bottom indicates: Query executed successfully. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 1 rows

المتغيرات الحرفية:

يمكن أن يضم الحقل النصي أي نوع من الرموز الحرفية مقروءة كانت أو غير مقروءة، إذا كنت تريد متغير بعدد ثابت من الحروف فقم بالإعلان عن متغير من النوع **char**، مثال:

```
DECLARE @Gender char;
```

افتراضيا يحجز نوع البيانات char متغير ذو حرف واحد، بعد الإعلان عن المتغير وعند

تهيئته توضع فيه القيمة بين علامتي اقتباس، هنا مثال:

```
DECLARE @Gender char;  
SET @GENDER = 'M';  
SELECT @Gender AS Gender;  
GO
```

إذا شمل المتغير أكثر من حرف واحد، فسيخزن الحرف الأول فقط، مثال:

```
DECLARE @Gender char;  
SET @GENDER = 'Male';  
SELECT @Gender AS Gender;  
GO
```

هذا الاستعلام يعيد الحرف M.

السلسلة الحرفية (String) هي حرف أو مجموعة من الحروف، إذا كان المتغير يخزن أطوال مختلفة من السلاسل الحرفية، فيجب إعلانه بالنوع **varchar**، سعة التخزين لهذا النوع لا تتعدى 8 kilobytes.

أحيانا تحتاج إلى تحديد عدد الحروف التي تستخدم في السلسلة الحرفية، فمثلا رغم اختلاف طول متغير الاسم ومتغير اسم الكتاب ينبغي استخدام النوع **varchar**.

لتحديد الحد الأقصى لعدد الحروف التي يمكن تخزينها في سلسلة حرفية، وعلى يمين المتغير char أو varchar أضف قوسين واكتب بينهما طول السلسلة الحرفية المرغوب فيه، لتهيئة المتغير إذا كنت تستخدم سطر الأوامر SQLCMD.EXE، ضعه بين شالنتين ""، مثال:

```

SQLCMD
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Documents and Settings\Administrator> SQLCMD
1> DECLARE @FirstName Char(20),
2> @LastName Char(20);
3> SET @FirstName = "François";
4> SET @LastName = "Décès";
5> SELECT @FirstName AS [First Name],
6> @LastName AS [Last Name];
7> GO
First Name          Last Name
-----
François            Décès

(1 rows affected)
1> -

```

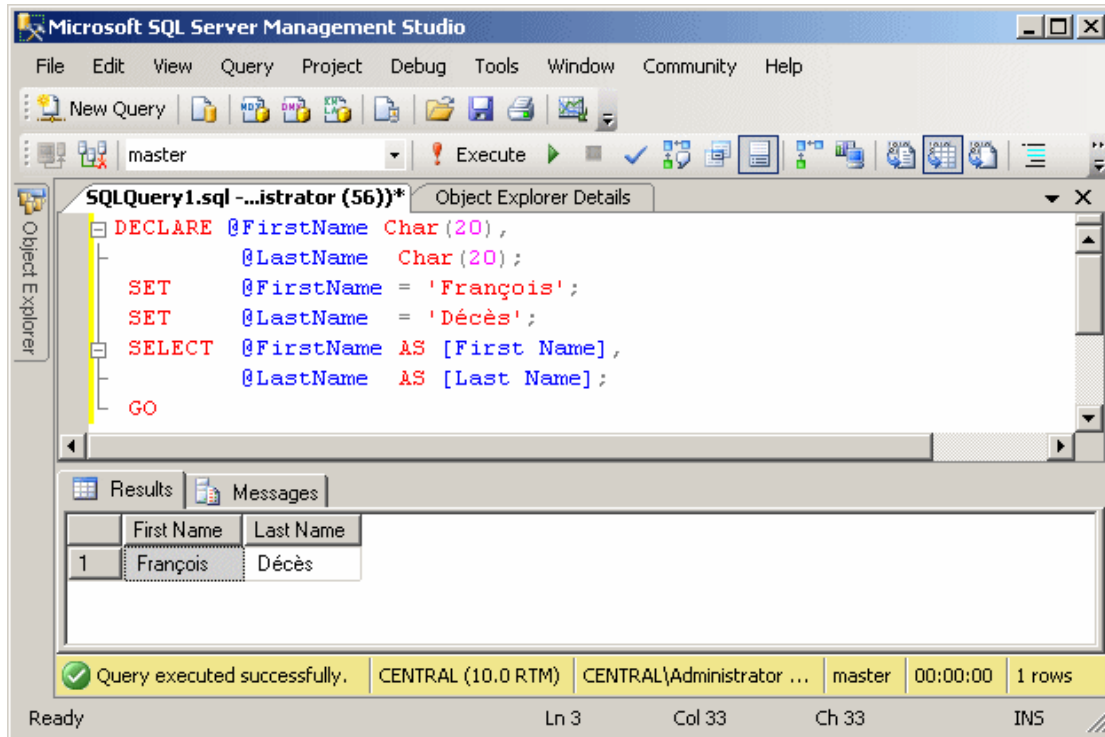
لا تكتب سلسلة حرفية بين شاليتين إذا كنت تستخدم نافذة الاستعلام، وإلا فستحصل على رسالة خطأ:

```

Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Community Help
New Query
master Execute
SQLQuery1.sql -...istrator (56))* Object Explorer Details
DECLARE @FirstName Char(20),
@LastName Char(20);
SET @FirstName = "François";
SET @LastName = "Décès";
SELECT @FirstName AS [First Name],
@LastName AS [Last Name];
GO
Messages
Msg 207, Level 16, State 1, Line 3
Invalid column name 'François'.
Msg 207, Level 16, State 1, Line 4
Invalid column name 'Décès'.
Query completed with errors. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 0 rows
Ready Ln 1 Col 1 Ch 1 INS

```

واكتف بعلمة الاقتباس فقط:



نوع البيانات **text** يستخدم على البيانات المتغيرة والتي ستتألف من حروف ASCII، كبدل عن النوع **varchar** يتسع النوع **text** إلى ما يزيد عن 8 kilobytes.

الأنواع **nchar**، **nvarchar**، و **ntext** تتبع نفس القواعد المشار إليها مع الأنواع **varchar**، **char** و **text**، إلا أنها يمكن أن تطبق على المتغيرات التي يمكن أن تشمل حروف اللغات الدولية، وهي حروف اللغات الأخرى غير الحروف اللاتينية، ويتم ذلك وفق قواعد معيار Unicode.

تطبيق عملي: استخدام المتغيرات الحرفية

1- اكتب في نافذة استعلام جديد، الاستعلام التالي:

```

DECLARE @FirstName varchar(20),
        @LastName varchar(20),
        @FullName varchar(40),
        @DateHired DateTime,
        @EmplStatus int,
        @IsMarried bit,
        @WeeklyHours Decimal(6,2),
        @HourlySalary SmallMoney,
        @WeeklySalary SmallMoney;

SET @FirstName      = 'Samuel';
SET @LastName       = 'Weinberg';
SET @FullName       = @LastName + ', ' + @FirstName;
SET @DateHired      = '12/05/1998';

```

```
SET @IsMarried      = 1;
SET @EmplStatus     = 2;
SET @WeeklyHours    = 36.50;
SET @HourlySalary   = 15.72;
SET @WeeklySalary   = @WeeklyHours * @HourlySalary;
SELECT @FullName As [Full Name],
       @DateHired AS [Date Hired],
       @EmplStatus AS [Empl Status],
       @IsMarried AS [Married?],
       @WeeklyHours AS Hours,
       @HourlySalary AS Hourly,
       @WeeklySalary AS Weekly;
GO
```

2- نفذ الاستعلام.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query in a text editor. The query declares several variables and sets their values, then selects the values of these variables into a table with aliases. The results pane shows a single row of data.

```

DECLARE @FirstName varchar(20),
        @LastName varchar(20),
        @FullName varchar(40),
        @DateHired DateTime,
        @EmplStatus int,
        @IsMarried bit,
        @WeeklyHours Decimal(6,2),
        @HourlySalary SmallMoney,
        @WeeklySalary SmallMoney;

SET @FirstName = 'Samuel';
SET @LastName = 'Weinberg';
SET @FullName = @LastName + ', ' + @FirstName;
SET @DateHired = '12/05/1998';
SET @IsMarried = 1;
SET @EmplStatus = 2;
SET @WeeklyHours = 36.50;
SET @HourlySalary = 15.72;
SET @WeeklySalary = @WeeklyHours * @HourlySalary;

SELECT @FullName AS [Full Name],
       @DateHired AS [Date Hired],
       @EmplStatus AS [Empl Status],
       @IsMarried AS [Married?],
       @WeeklyHours AS Hours,
       @HourlySalary AS Hourly,
       @WeeklySalary AS Weekly;

GO

```

	Full Name	Date Hired	Empl Status	Married?	Hours	Hourly	Weekly
1	Weinberg, Samuel	1998-12-05 00:00:00.000	2	1	36.50	15.72	573.78

Query executed successfully. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 1 rows

إحفظ الملف باسم Variables في المجلد My Documents لحاسوبك.

ملخص الدرس: تمارين متنوعة

المطلوب:

- كتابة استعلام يعيد الراتب السنوي للعامل، وأجرة كل ساعة (باعتبار الراتب السنوي يحتوي على 40 ساعة في الأسبوع).

- كتابة استعلام يحسب طول القاعدة والقيمة لمثلث، ثم يحسب المساحة (مساحة المثلث هي: القاعدة \times الارتفاع / 2).

تمرين (شركة المرافق):

- افتح نافذة الاستعلام من قاعدة البيانات UtilityCompany1
- عرف متغير من النوع عملة نقدية باسم BaseCharge ثم هيئه بالقيمة 8.50.
- أعلن عن متغير باسم First700 وهيئه بالعدد 6.50.
- أنشئ ونفذ استعلام يحسب جداء العدد 224 إلى المتغير First700 مع عرض النتيجة في شكل نسبة مئوية.
- أنشئ ونفذ الاستعلام الذي يضيف إلى النتيجة السابقة قيمة 8.50 للحصول على مبلغ الفواتير لشهر.

أنواع العبارات SQL

عبارات المقارنة:

عن طريق SQL يمكن إنشاء عبارات مقارنة بين مجموعة من القيم، أو المتغيرات، والمعاملات، يدعم Transact-SQL مجموعة من العمليات على غرار التي رأيناها فيما سبق.

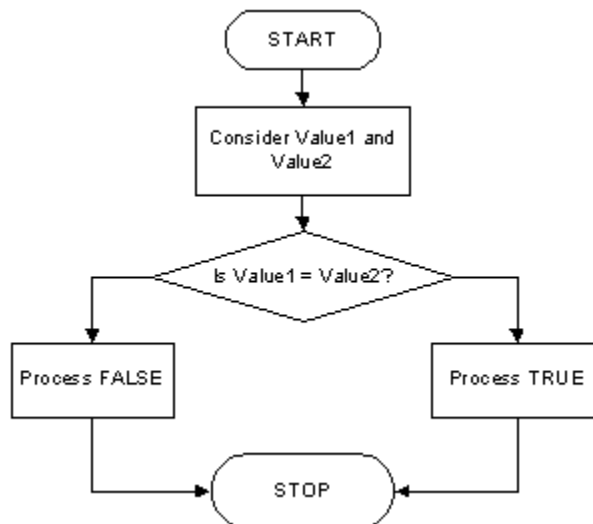
عملية المقارنة هي عملية منطقية تنتج أحد القيمتين True أو False اعتمادا على القيم الواردة في المقارنة، وتتم المقارنة بين قيمتين من النوع نفسه (رقمين، حرفين، سلسلتين نصيتين...)، ويدعم Transact-SQL لأجل ذلك كل عبارات المقارنة الضرورية.

المقارنة بالمساواة (=):

للمقارنة بين أي قيمتين استخدام رمز المساواة (=)، صيغته العامة:

`Value1 = Value2`

تستخدم المساواة بين قيمتين للتحقق من تساويهما، من الصيغة أعلاه يقارن SQL بين قيمة Value1 و Value2، فإذا كانتا متساويتين فسينتج المترجم القيمة True، وإذا كانتا مختلفتين تصبح قيمة المقارنة False، ويمكن توضيح ذلك على النحو التالي:

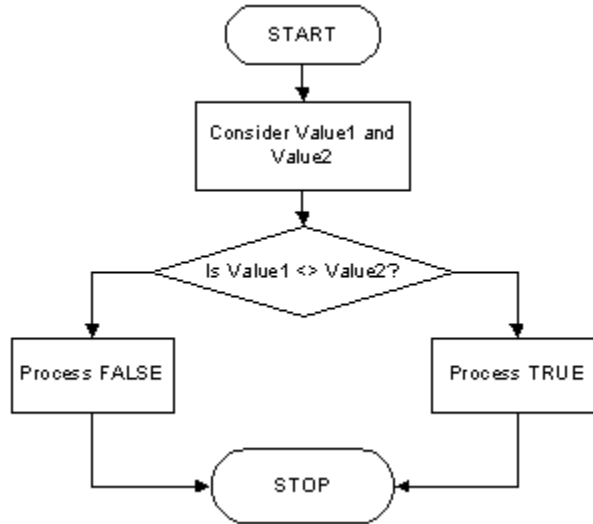


المقارنة بالاختلاف (<>):

بدلا من المساواة لمعرفة للتحقق من تساوي القيم استخدم الرمز <> بصيغته العامة:

Value1 <> Value2

رمز المقارنة المنطقي <> يستخدم للمقارنة بين القيم، حيث يعيد المترجم القيمة True إذا كانت القيمتين مختلفتين ويعيد False إذا كانتا متساويتين وفق المخطط التالي:



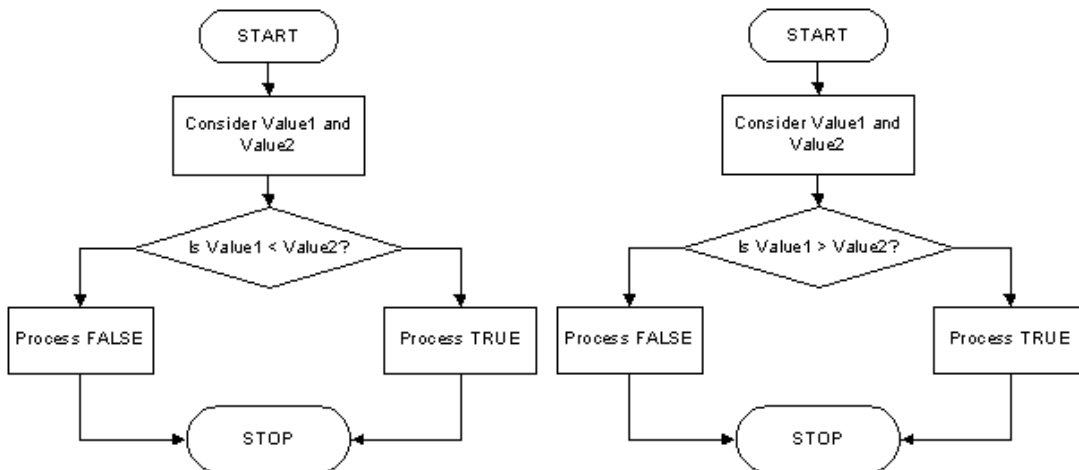
المقارنة بأصغر من (<) وأكبر من (>):

كباقي لغات البرمجة يدعم Transact-SQL صيغة المقارنة بالرمز < أو >، الصيغة:

Value1 < Value2

Value1 > Value2

في المقارنة السابقة يعيد المترجم أحد القيمتين True أو False حسب قيم المتغيرات.



المقارنة بأقل أو يساوي وأكبر أو يساوي (<= و >=):

عملية المقارنة <= أو >= تجمع بين المقارنتين = و < أو > صيغتها:

```
Value1 >= Value2
Value1 <= Value2
```

إذا تساوت القيمتين Value1 و Value2 أو كانتا تحققان المقارنة < أو > فنتيجة المقارنة صحيحة True، وإذا اختلفتا أو لم تحقق عملية المقارنة < أو > فالنتيجة خاطئة False.

العمليات الشرطية:

شرط المقارنة هو تقييم لوضعية عبارة SQL، تنتج عملية المقارنة أحد القيمتين True أو False، ويمكنك بناء على نتيجة المقارنة اتخاذ الإجراءات اللازمة. ينبغي دائماً كتابة شرط المقارنة بأبسط ما يمكن ليقبلها المترجم، وحتى لو لم يحصل خلط عند المترجم فإن ذلك يسهل عليك إيجاد محل الخطأ وتصحيحه بسهولة.

فيما سيأتي سنتطرق إلى الكلمات المفتاحية والصيغ التي يتيحها Transact-SQL لمساعدتك من أجل صياغة عبارات واضحة، عادة ما تبدأ العبارات بكلمات رئيسية ثم محتوى الشرط، وبعد عبارة الشرط يمكنك تكليف المترجم بما ينبغي أن يفعله، الشكل العام على النحو التالي:

```
Keyword Expression
Statement
```

العبارتين BEGIN و END:

بعد كتابة عبارة SQL (*Expression*) يمكنك كتابة جمل SQL (*Statement*) في سطر واحد، لكنك قد تحتاج في معظم الحالات إلى أكثر من سطر وهنا استخدم العبارتين BEGIN و END لكتابة مجموعة من الجمل في كتلة واحدة.

يُعتبر المترجم ما يأتي بعد عبارة SQL وحدة فقط إذا كانت الجملة التي بعدها في سطر وحيد، لتبنيه المترجم بحالة وجود عدة أكثر من سطر نضيف الأمر BEGIN عند بداية التعليمات ونغلق الكتلة بالأمر END. وتكون في هذه الحالة صيغة العبارة على النحو التالي:

```
Keyword Expression
BEGIN
    Statement Line 1
    Statement Line 2
    Statement Line n
END
```

بإمكانك استخدام BEGIN و END حتى إذا كان الاستعلام يغطي سطر واحد، مثل:

```
Keyword Expression
BEGIN
    Statement
END
```

استخدام عبارات محصورة بين BEGIN و END يجعل الاستعلام أسهل في القراءة لأنه يشير بوضوح إلى موضع البداية والانتهاؤ.

العبارة الشرطية IF:

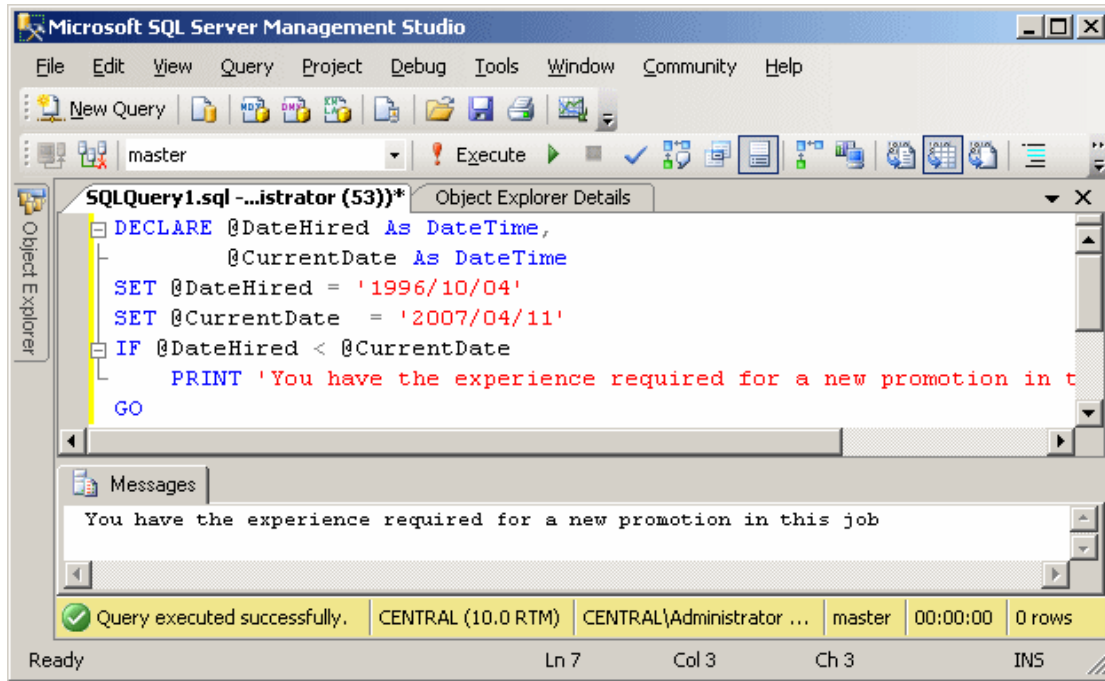
للتحقق من صحة أي عبارة شرطية يمكن استخدام العبارة الشرطية IF التي يتوفر عليها Transact-SQL، الصيغة العامة:

```
IF Condition
    Statement
```

عند إنشاء استعلام باستخدام الشرط IF يجب التحقق أن نتيجة الشرط تكون صحيحة True أو خاطئة False، مثال:

```
DECLARE @DateHired As DateTime,
        @CurrentDate As DateTime
SET @DateHired = '1996/10/04'
SET @CurrentDate = '2007/04/11'
IF @DateHired < @CurrentDate
    PRINT 'You have the experience required for a new promotion
in this job'
GO
```

نتيجة هذا الاستعلام:



العبارة الشرطية IF..ELSE:

الاستعلام IF مناسب فقط في حالة ما إذا كان الشرط صحيح، وللتحقق من الحالتين (صحة الشرط من عدمها) نضيف الأمر ELSE. في حالات كثيرة لا يوجد شيء يجب القيام به إذا لم تتحقق صحة التعبير. الصيغة العامة:

```
IF Condition Statement1 ELSE Statement2
```

مثال:

```
DECLARE @DateHired As DateTime,
        @CurrentDate As DateTime
SET @DateHired = '1996/10/04'
SET @CurrentDate = '1995/04/11'
IF @DateHired < @CurrentDate
    PRINT 'You have the experience required for a new promotion
in this job'
ELSE
    PRINT 'Sorry, you have NOT the experience required'
GO
```

العبارة الشرطية CASE...WHEN...THEN:

الكلمة CASE تستخدم لفحص حالة متغير تبعاً لكل قيمة، صيغتها العامة:

```
CASE Expression
    WHEN Value1 THEN Result
    WHEN Value2 THEN Result
    ...
    WHEN Value_n THEN Result
END
```

في المثال التالي إذا كان الحرف CharGender يحمل القيمة M فسيتم إنشاء سلسلة حرفية بالقيمة Male، إذا كان الحرف F فسيتم إنشاء سلسلة بالقيمة Female:

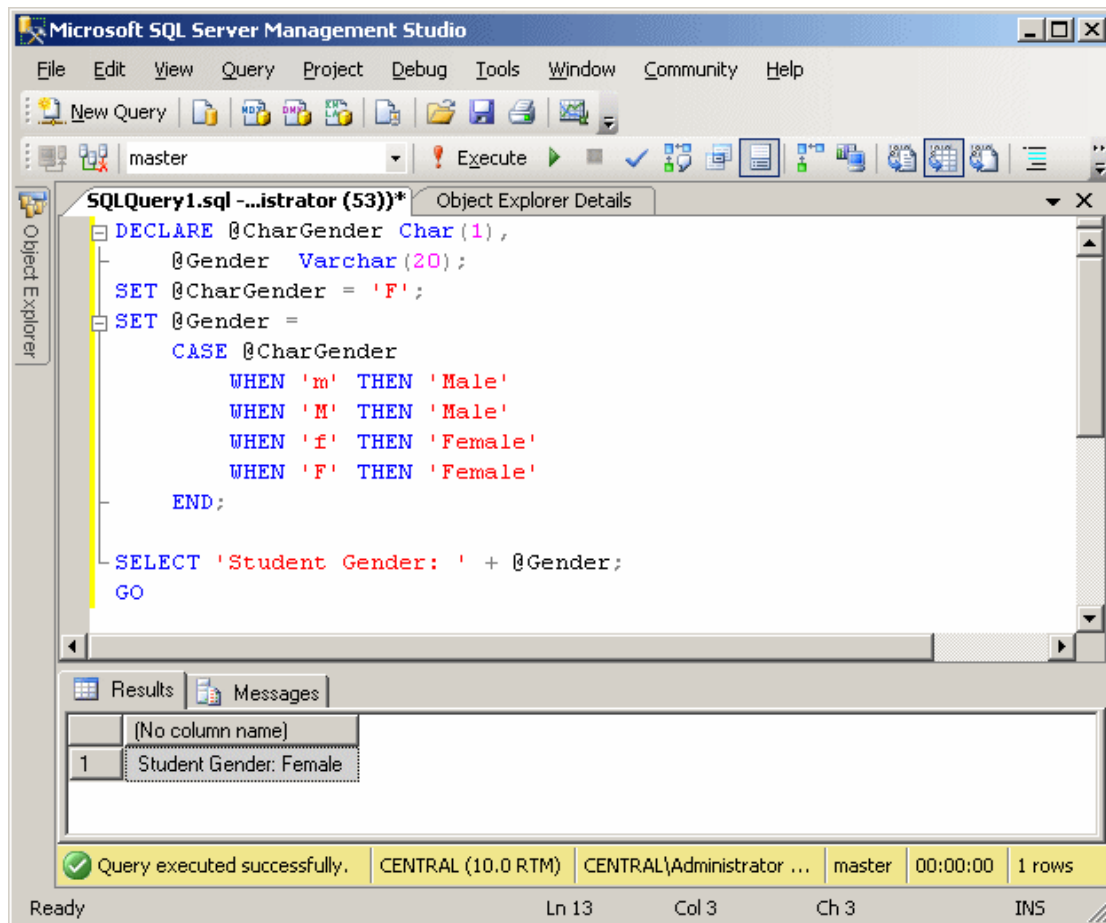
```

DECLARE @CharGender Char(1),
        @Gender Varchar(20);
SET @CharGender = 'F';
SET @Gender =
    CASE @CharGender
        WHEN 'm' THEN 'Male'
        WHEN 'M' THEN 'Male'
        WHEN 'f' THEN 'Female'
        WHEN 'F' THEN 'Female'
    END;

SELECT 'Student Gender: ' + @Gender;
GO

```

نتيجة تنفيذ الاستعلام:



العبارة الشرطية CASE...WHEN...THEN...ELSE :

في حالات كثيرة يمكنك التنبؤ أو معرفة بعض القيم التي ستنتجها القضية، وفي حالات أخرى يمكن أن تنتج قيمة غير متوقعة، وفي هذه الحالة استخدم الكلمة ELSE لكتابة الأمر المتعلق بقيمة غير القيم المتوقعة، وشكلها العام كما يلي:

```
CASE Expression
  WHEN Value1 THEN Result
  WHEN Value2 THEN Result
  WHEN Value_n THEN Result

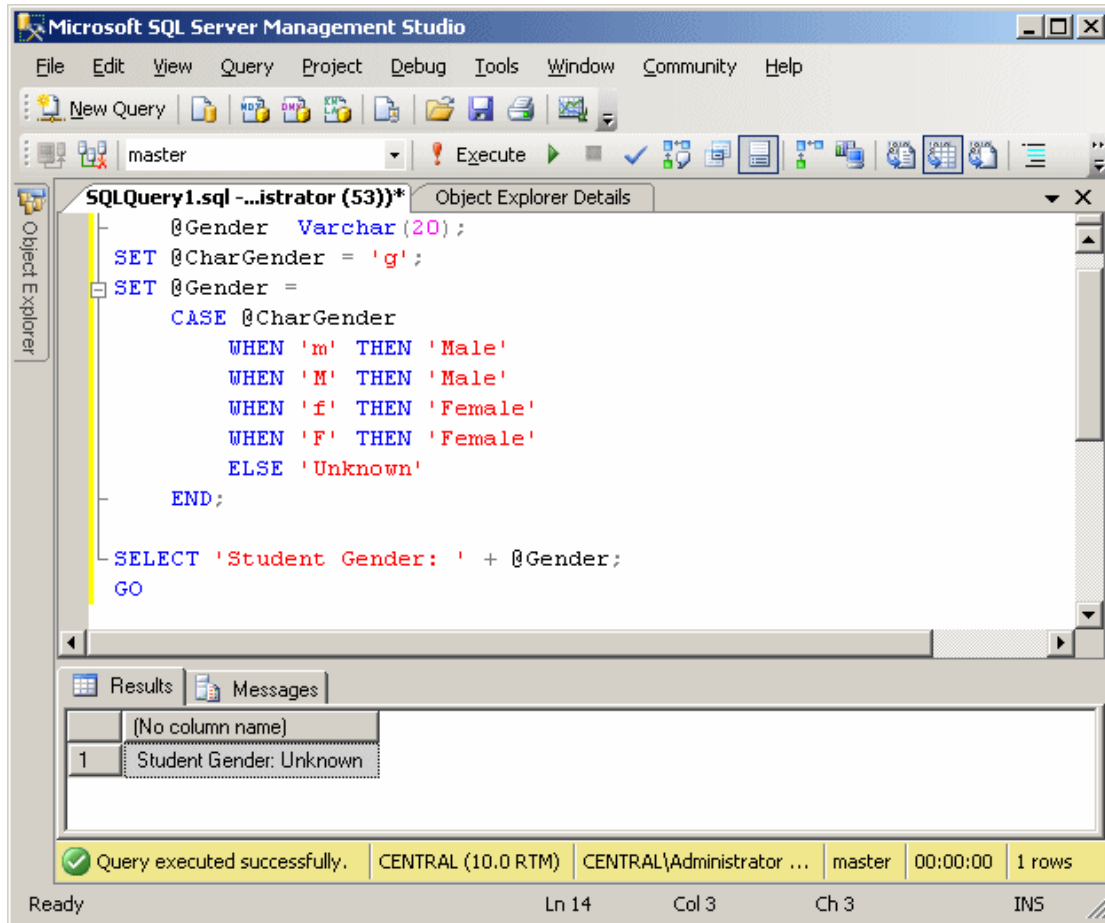
  ELSE Alternative
END
```

تستخدم الكلمة ELSE عندما تأخذ العبارة الشرطية قيمة غير متوقعة، مثال:

```
DECLARE @CharGender Char(1),
        @Gender Varchar(20);
SET @CharGender = 'g';
SET @Gender =
    CASE @CharGender
        WHEN 'm' THEN 'Male'
        WHEN 'M' THEN 'Male'
        WHEN 'f' THEN 'Female'
        WHEN 'F' THEN 'Female'
        ELSE 'Unknown'
    END;

SELECT 'Student Gender: ' + @Gender;
GO
```

وهذا من شأنه أن ينتج:



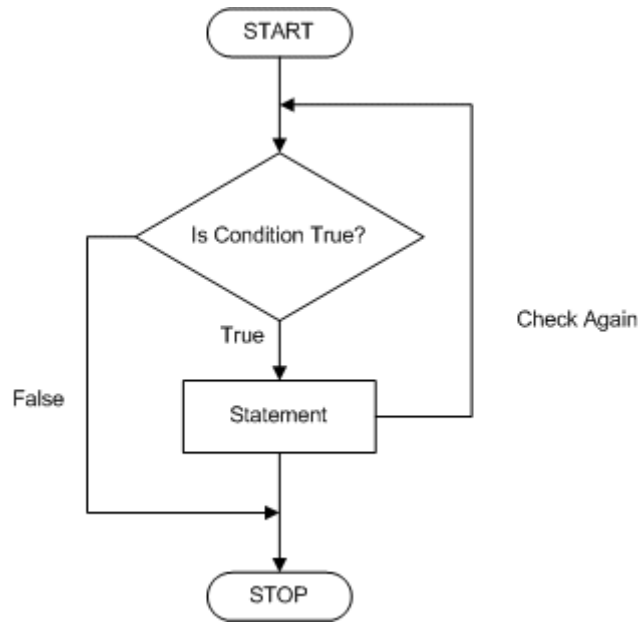
إذا لم تستعمل ELSE في العبارة الشرطية وكانت النتيجة خارج القيمة المقترحة، فسيأخذ المتغير القيمة المعدومة NULL، وهذا يعني أن ELSE تقوم بحماية العبارة الشرطية CASE.

الحلقة WHILE:

لدراسة وتقييم عبارة SQL قبل اتخاذ أي إجراء، يمكنك استخدام الكلمة WHILE، صيغتها العامة:

WHILE Expression Statement

لاستخدام هذا الاستعلام قم بإضافة عبارة (*Expression*) بعد الكلمة WHILE ثم الأمر الذي ترغب في تكراره، وهنا ينبغي أن ترجع العبارة أحد القيمتين True أو False، وإذا تحقق شرط العبارة فسيقوم المترجم بتنفيذ الأمر ثم يرجع مرة أخرى إلى التحقق من صحة العبارة، ويبقى المترجم على تلك الحال حتى يصبح شرط العبارة غير صحيح False وعندها يتوقف المترجم عن الفحص، يمكن توضيح ذلك على النحو التالي:

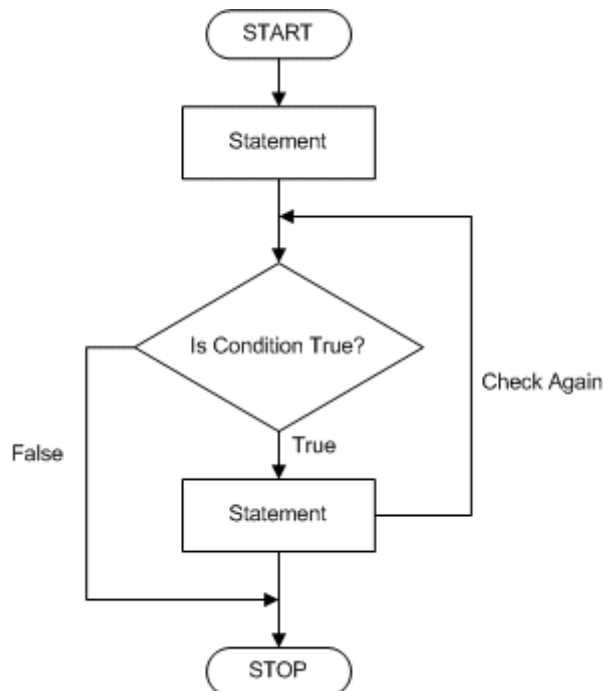


مثال:

```

DECLARE @Number As int
WHILE @Number < 5
    SELECT @Number AS Number
GO
  
```

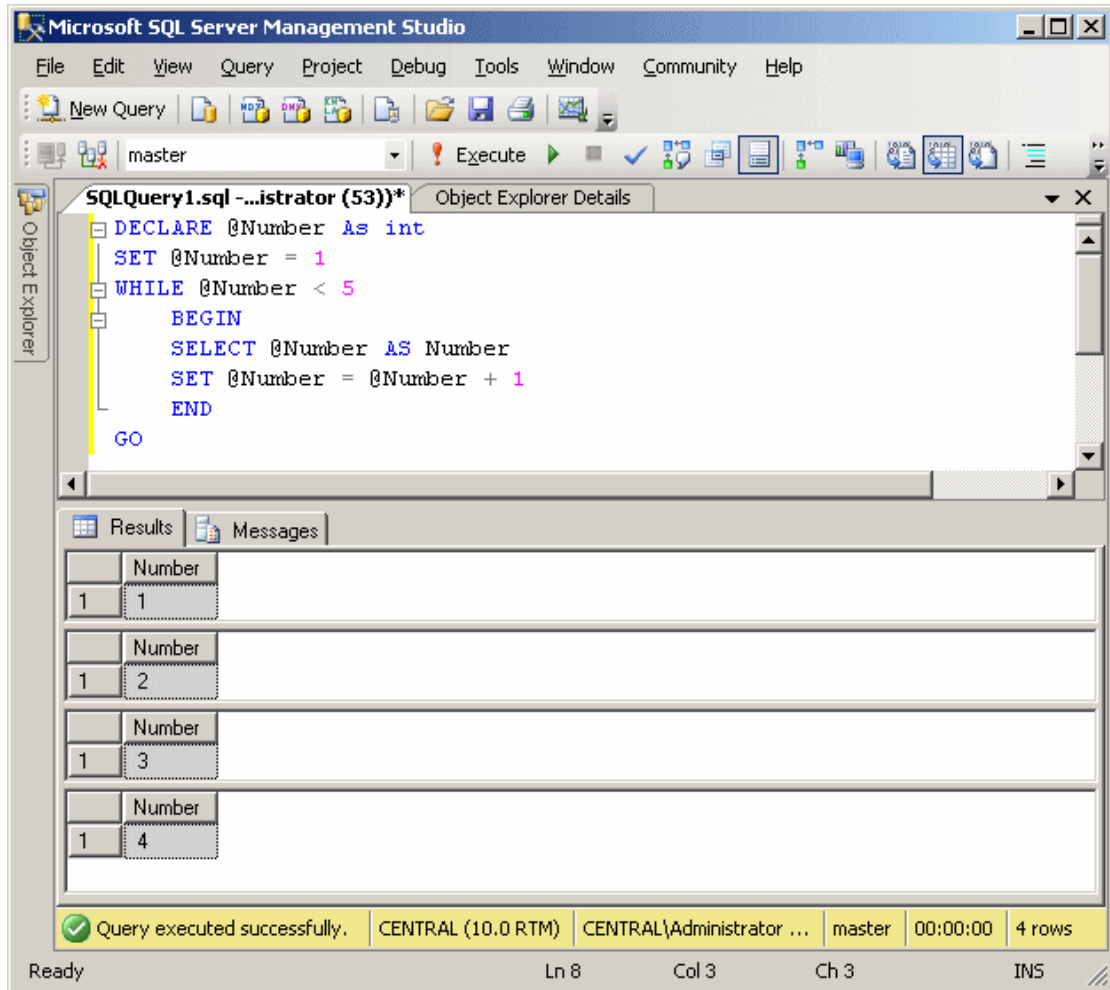
عند تنفيذ الشرط WHILE، يجب عليك أن تتأكد من توفير آلية تسمح للمترجم بالحصول على قيمة مرجعية لحالة المتغير أو التعبير المفحوص، فقد يحدث هذا في بعض الأحيان في شكل متغير مهياً مسبقاً أو قد ينتج عن عبارة أخرى، يمكن توضيح ذلك فيما يلي:



هذه المرة سينفذ الاستعلام على النحو التالي:

```
DECLARE @Number As int
SET @Number = 1
WHILE @Number < 5
    BEGIN
        SELECT @Number AS Number
        SET @Number = @Number + 1
    END
GO
```

ونتيجه:



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a T-SQL query in a text editor. The query is as follows:

```
DECLARE @Number As int
SET @Number = 1
WHILE @Number < 5
    BEGIN
        SELECT @Number AS Number
        SET @Number = @Number + 1
    END
GO
```

Below the query editor, the Results pane shows the output of the query. It displays four rows of data, each with a column header 'Number' and a value from 1 to 4. The status bar at the bottom indicates that the query was executed successfully, returning 4 rows.

Number
1
2
3
4

Query executed successfully. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 4 rows

ثوابت المنطق:

توفر بيانات البرمجة وقواعد البيانات عوامل يمكنك استخدامها لتحليل وصيانة البيانات، تدعى هذه العوامل بالعوامل المنطقية لأنها تستخدم لأداء المقارنات التي تنتج نتيجة صحيحة أو خاطئة (لا نتيجة أخرى بينهما).

القيمتين الثابتتين TRUE وFALSE:

القيمة الصحيحة TRUE يمكن أن يعبر عنها بالرقم 1 أو الكلمة Yes، ومثل ذلك FALSE التي يعبر عنها بالقيمة الخاطئة أو بالرقم 0، فالتحقق مثلا من وجود قيمة في حقل معين يمكنك فقط معرفة ما إذا كانت قيمة الحقل موجودة أم لا.

عادة ما تطبق المقارنة بالقيمتين True أو False على الحقول التي تحمل قيمة منطقية، فإذا سجلت القيمة 1 فيعتبر الحقل يحمل قيمة صحيحة True، وإذا سجلت فيه قيمة 0 فالحقل يحمل قيمة خاطئة False.

القيمة الثابتة المعدومة NULL:

بعد الإعلان عن متغير SQL يحجز المترجم مساحة من ذاكرة الحاسب لكنه لا يضع أي شيء في تلك الذاكرة، وفي ذلك الوقت لا تحمل الذاكرة أية قيمة للمتغير، وحينها يقال أن المتغير يحمل القيمة المعدومة NULL.

جُهِز Transact-SQL بقيمة الثابت المعلوم NULL، وعادة ما تستخدم لأغراض المقارنة، وهذا لا يعني أن قيمته تساوي 0.

المعامل IS:

للتحقق من إمكانية عمل شيء ما، يمكنك استخدام المعامل IS، كما يمكنك استخدام التعبير IS NULL مثلا كما يلي:

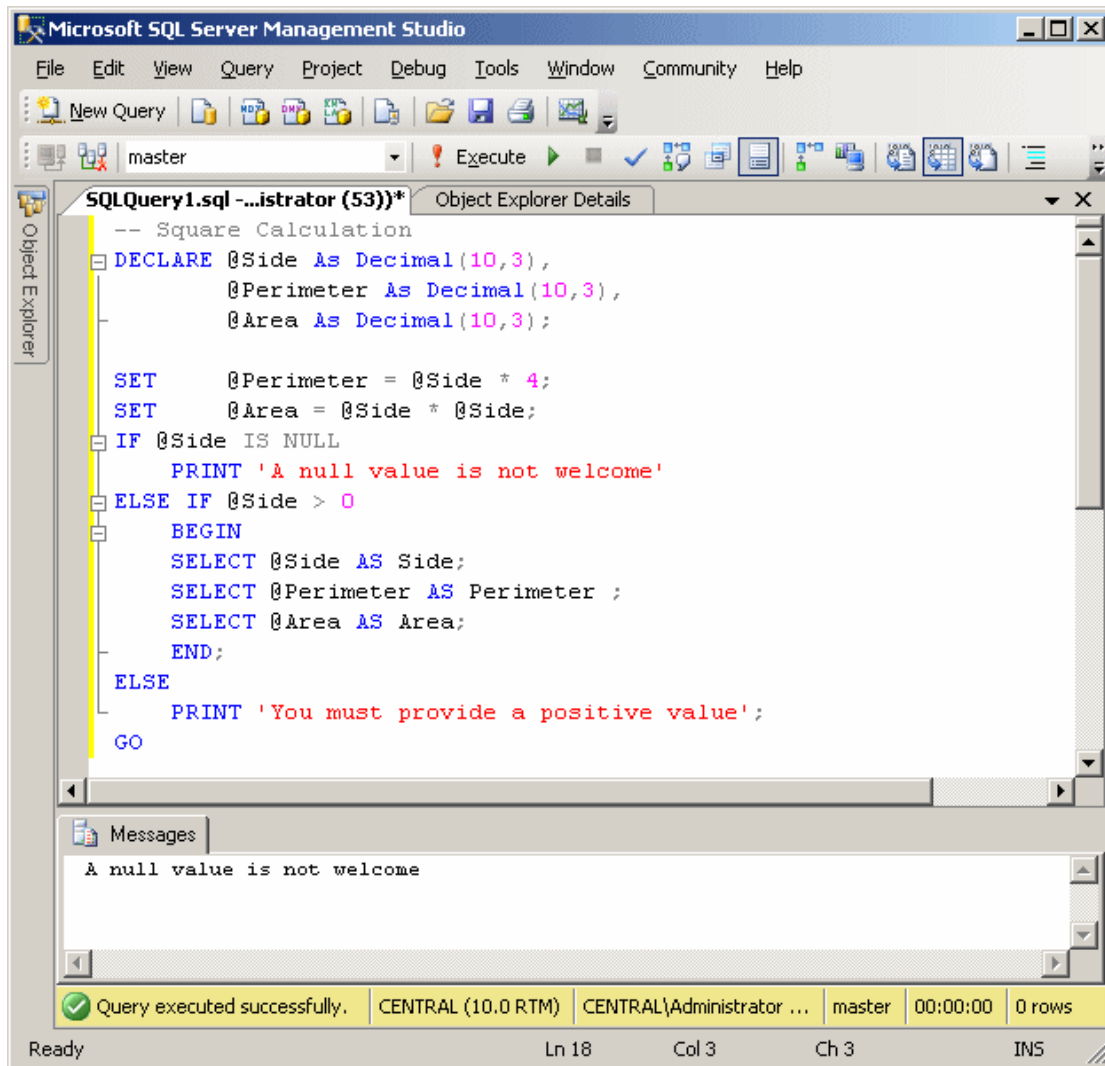
```
-- Square Calculation
DECLARE @Side As Decimal(10,3),
        @Perimeter As Decimal(10,3),
        @Area As Decimal(10,3);

SET     @Perimeter = @Side * 4;
SET     @Area = @Side * @Side;
IF @Side IS NULL
    PRINT 'A null value is not welcome'
ELSE IF @Side > 0
    BEGIN
        SELECT @Side AS Side;
        SELECT @Perimeter AS Perimeter ;
        SELECT @Area AS Area;
    END;
ELSE
```

```
PRINT 'You must provide a positive value';
```

GO

ونتيجة ذلك:



لاجتناب وجود القيمة المعدومة (NULL)، يمكنك تهيئة المتغير أو أن تعين قيمة له،

مثال:

```
-- Square Calculation
DECLARE @Side As Decimal(10,3),
        @Perimeter As Decimal(10,3),
        @Area As Decimal(10,3);

SET     @Side = 48.126;
SET     @Perimeter = @Side * 4;
SET     @Area = @Side * @Side;
IF @Side IS NULL
    PRINT 'A null value is not welcome'
ELSE IF @Side > 0
    BEGIN
        SELECT @Side AS Side;
        SELECT @Perimeter AS Perimeter ;
        SELECT @Area AS Area;
```

```

END;
ELSE
    PRINT 'You must provide a positive value';
GO

```

ونتيجة هذا الاستعلام:

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query titled "SQLQuery1.sql -...istrator (53)*". The query is as follows:

```

-- Square Calculation
DECLARE @Side As Decimal(10,3),
        @Perimeter As Decimal(10,3),
        @Area As Decimal(10,3);
SET     @Side = 48.126;
SET     @Perimeter = @Side * 4;
SET     @Area = @Side * @Side;
IF @Side IS NULL
    PRINT 'A null value is not welcome'
ELSE IF @Side > 0
    BEGIN
        SELECT @Side AS Side;
        SELECT @Perimeter AS Perimeter ;
        SELECT @Area AS Area;
    END;
ELSE
    PRINT 'You must provide a positive value';
GO

```

Below the query editor, the Results pane shows the output of the query. It contains three tables:

Side
1 48.126

Perimeter
1 192.504

Area
1 2316.112

The status bar at the bottom indicates: "Query executed successfully. CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 1 rows".

معامل النفي NOT:

لفحص وجود أو توفر قيمة متغير، يمكنك استخدام المعامل NOT، هذا المعامل يستخدم في أكثر أحواله مع العبارات المنطقية.

ملخص الدرس:

تمارين متنوعة:

- كتابة الاستعلام الذي يمكن أن يتعرف ويعرض كل عدد يقبل القسمة على 3،
وينحصر بين 1 و 30.
- اكتب الاستعلام الذي من شأنه أن يحسب القطر، المحيط، والمساحة لأي دائرة.
- بنفس الطريقة مع شكل أسطوانة.

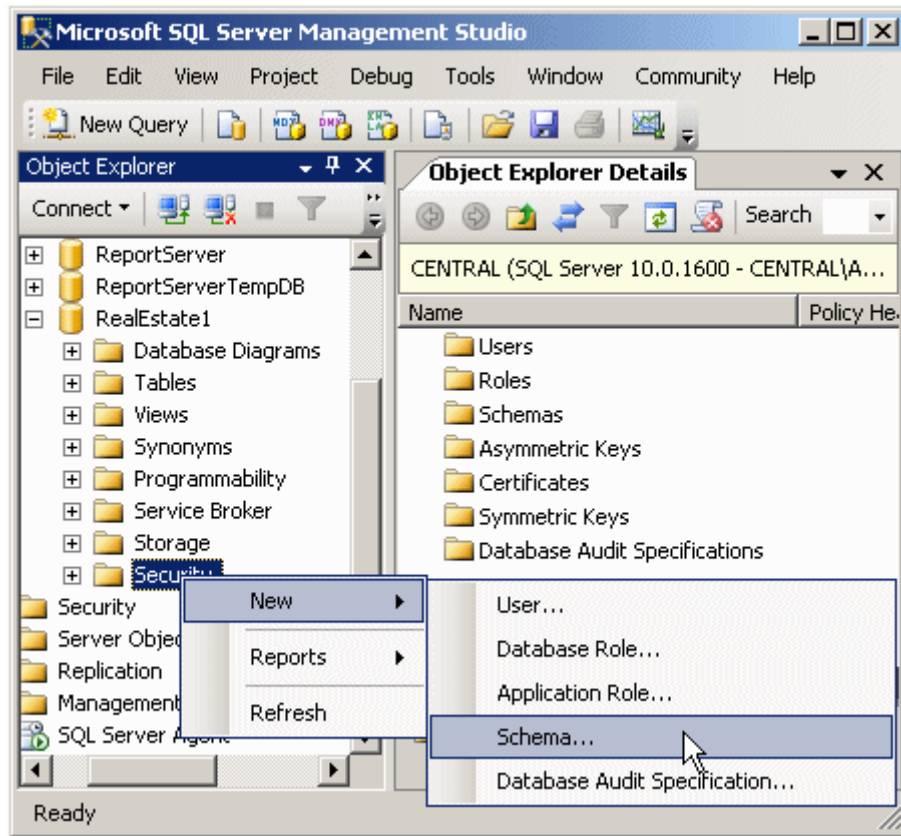
مدخل إلى الدوال في SQL

أساسيات الدوال:

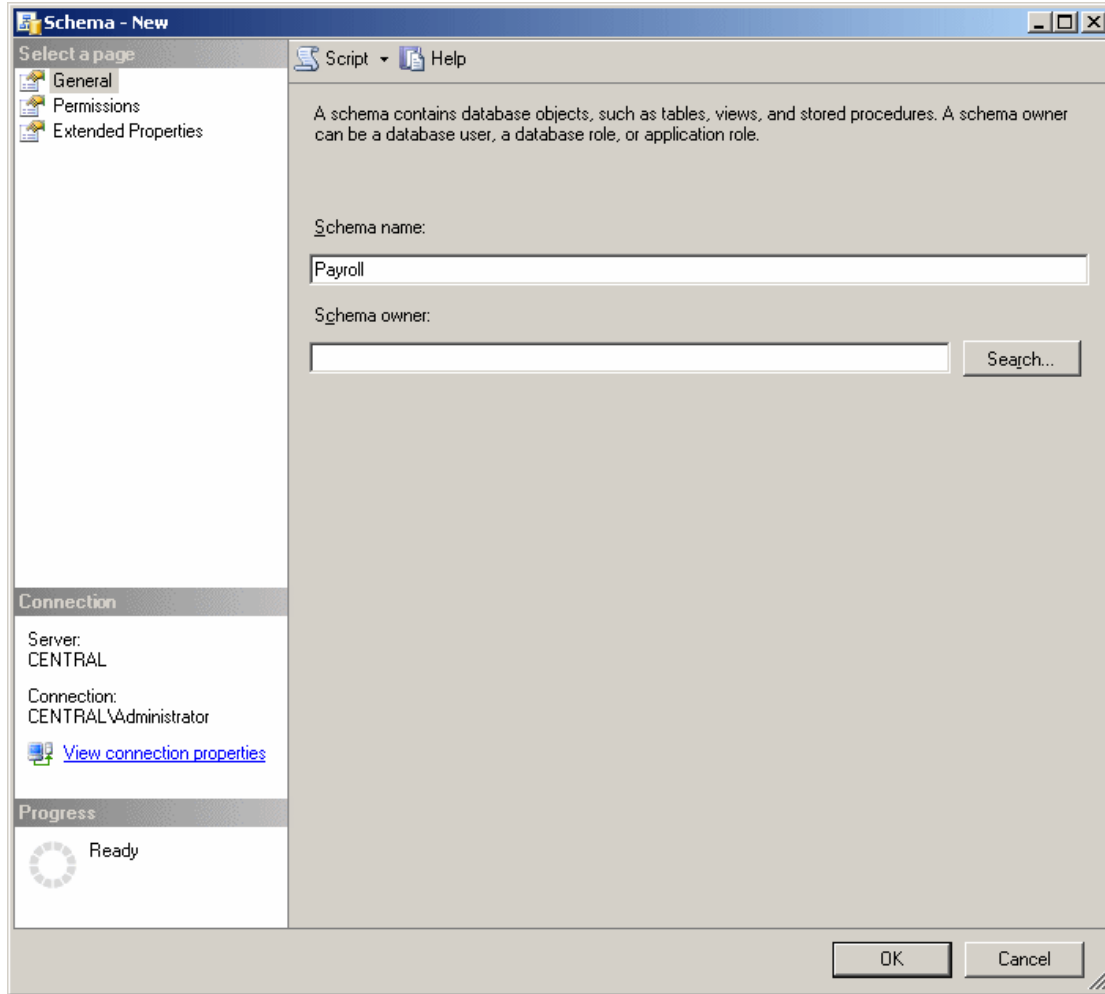
التابع أو الدالة Function هي وظيفة أو مهمة منفردة صغيرة نسبياً بحيث تسمح بالحصول في أي وقت على نتائجها، تعتبر الدالة function في Transact-SQL غرضاً يجب إنشاؤه قبل استخدامه، وبالتالي تصبح الدالة جزءاً من قاعدة البيانات وغرضاً يمكن الوصول إليه.

تطبيق عملي: استخدام الدوال

- 1- تأكد من استعمالك اسم مستخدم pkatts واسم الحساب الذي أنشئ في بداية الدروس.
- 2- شغل SQL Server Management Studio
- 3- لإنشاء قاعدة بيانات جديدة انقر بالزر الأيمن على Databases واختر New Database...
- 4- في مربع النص لاسم قاعدة البيانات اكتب RealEstate1
- 5- حدد مالك لقاعدة البيانات ComputerName\pkatts (في نظام Windows XP أو Vista) أو DomainName\pkatts (في نظام Windows Server).
- 6- حدد المسار C:\Microsoft SQL Server Database Development، ثم انقر OK.
- 7- في متصفح الكائنات قم بتوسيع Databases من البند RealEstate1، ثم انقر بالزر الأيمن على Security، واختر New، ثم انقر على Schema... لإنشاء مخطط جديد.



8- اكتب اسم المخطط Payroll واضغط Enter.



أساسيات إنشاء الدوال:

هناك عدة طرق لإنشاء الدوال اختر أحدها:

- من متصفح الكائنات حدد قاعدة البيانات المرغوبة ثم من العنصر Programmatically اختر Functions، انقر بالزر الأيمن على Scalar-Valued Function، ثم انقر على New Scalar-Valued Function...، وحينها سيُخَرِّج نموذج لشيفرة SQL لكتابة الدالة، يمكنك عندئذ تعديلها ثم استخدامها.
 - افتح نافذة استعلام جديدة، ثم اختر من متصفح النماذج (Templates Explorer) البند Function، قم بسحب وإفلات البند Scalar-Valued Function إلى نافذة الاستعلام.
- يمكنك مباشرة فتح نافذة استعلام جديدة والبدء في الكتابة استعلام الدالة الخاصة بك.

الصيغة العامة لإنشاء دالة:


```
CREATE FUNCTION FunctionName()
```

تسمية الدالة:

سننتج في هذه الدروس القواعد التالية لتسمية الدوال:

- أول رمز من الاسم يكون حرف أو الرمز _، مثال: are_n, act, أو Second.
- بعد أول رمز تركيبية من الحروف و الأرقام الرمز _، مثال: n24, act_52_t.
- لن تشمل الأسماء أيًا من هذه الرموز: @، #، \$، %، ^، &، *، !
- علينا تجنب استخدام الفراغات في الاسم إلا في حالات خاصة.
- إذا كان الاسم مركب من عدة كلمات فكل كلمة تبدأ بالأحرف الكبيرة، أمثلة
DateHired، _RealSport، DriversLicenseNumber.

القيمة المسترجعة من الدالة:

لتكون الدالة مفيدة يجب أن تسفر عن نتيجة، فعند إنشاء دالة لابد من تحديد نوع القيمة التي ستعيدها الدالة، لأجل ذلك اكتب بعد اسم الدالة الكلمة المفتاحية RETURNS متبوعة بنوع البيانات، مثال:

```
CREATE FUNCTION Addition()  
RETURNS Decimal(6,3)
```

بعد تحديد اسم ونوع القيمة التي ستعود بها الدالة، يمكنك كتابة جسم الدالة، جسد الدالة يبدأ الكلمة BEGIN وينتهي بالكلمة END، مثال:

```
CREATE FUNCTION Addition()  
RETURNS Decimal(6,3)  
BEGIN  
  
END
```

يمكنك إضافة AS كخيار إضافي:

```
CREATE FUNCTION Addition()  
RETURNS Decimal(6,3)  
AS  
BEGIN  
  
END
```

بين BEGIN و END حيث جسم الدالة، يمكنك استخدام المتغيرات وتخزين قيم فيها وكتابة عبارات SQL أخرى وكل ما يجب أن تقوم به الدالة، وبعد بكل ذلك وقبل الكلمة END، يجب أن تحدد القيمة التي ترجعها الدالة، ويتم ذلك عن طريق الكلمة المفتاحية RETURN متبوعة بالعباراة المناسبة وفق الصيغة التالية:

```
CREATE FUNCTION Addition()  
RETURNS Decimal(6,3)  
AS  
BEGIN  
    RETURN Expression  
END
```

مثال:


```
CREATE FUNCTION GetFullName()  
RETURNS varchar(100)  
AS  
BEGIN  
    RETURN 'Doe, John'  
END
```

تطبيق عملي: إنشاء دالة

1- من متصفح الكائنات انقر بالزر الأيمن على RealEstate1 واختر New Query...

2- لإنشاء دالة اكتب الاستعلام الآتي:

```
CREATE FUNCTION CalculateWeeklySalary()  
RETURNS Decimal(8, 2)  
AS  
BEGIN  
    RETURN 880.44  
END;  
GO
```

3- لتنفيذ الاستعلام انقر من شريط الأدوات على زر التنفيذ .

4- إحتفظ الاستعلام في ملف SQL من الزر حفظ Save.

5- من متصفح الكائنات قم بتوسيع بند قاعدة البيانات RealEstate1، ثم

Functions ثم Programmability وافتح Scalar-Valued Functions لتلاحظ العنصر

CalculateWeeklySalary الذي يمثل الدالة التي قمنا بإنشائها.

استدعاء الدالة:

بعد إنشاء الدالة يمكن استدعاؤها لاستخدام القيم التي تعيدها، للقيام بذلك اكتب اسم قاعدة البيانات التي أنشئت فيها الدالة تليها نقطة (.) ثم اسم المخطط (المخطط النظامي dbo) ثم نقطة أخرى ثم اسم الدالة متبوعا بقوسيتها، بهذه الصيغة:

```
DatabaseName.dbo.FunctionName()
```

بما أن الدالة تعيد قيمة فيمكنك استخدام تلك القيمة على النحو الذي تراه مناسباً، مثلاً يمكنك استخدام أمر الطباعة (PRINT) أو الانتقاء (SELECT) لعرض قيمة الاستعلام، هنا مثال لاستدعاء الدالة GetFullName() التي أنشأناها سابقاً:

```
PRINT Exercise.dbo.GetFullName();
```

بطريقة أخرى لاستدعاء الدالة انقر بزر اليمين على اسمها من متصفح الكائنات، حدد Script Function As ثم اختر SELECT To وانق على New Query Editor Window.

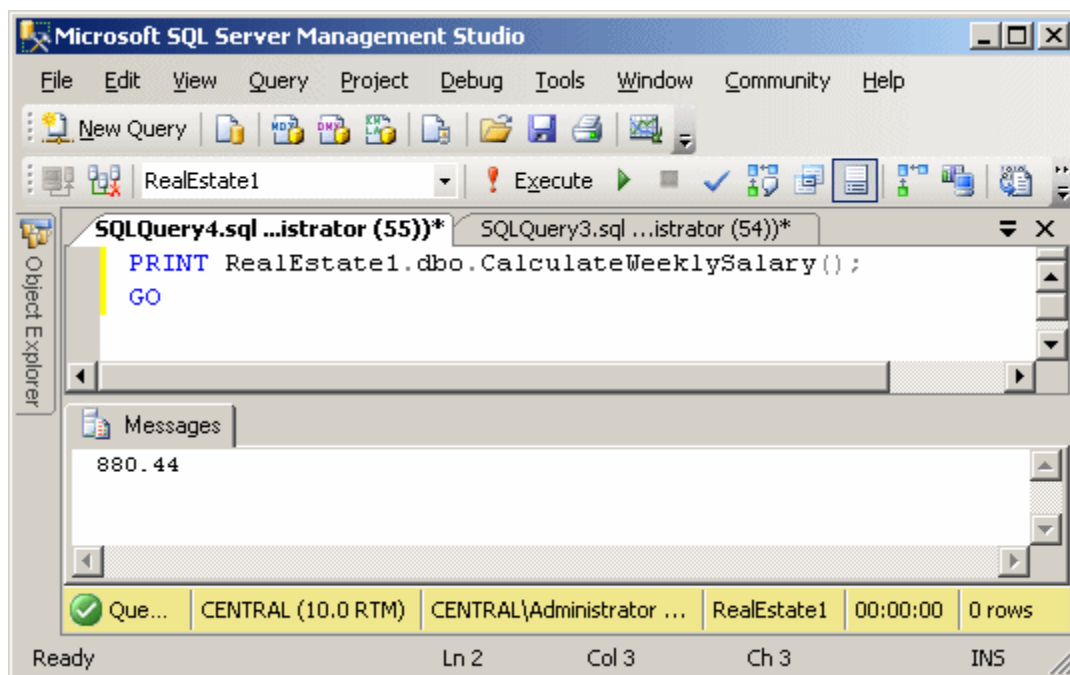
تطبيق عملي: استدعاء الدالة

1- انقر من متصفح الكائنات بزر اليمين على RealEstate1 وانقر New Query

2- لتنفيذ الدالة التي أنشأناها سابقاً نفذ الاستعلام الآتي:

```
PRINT RealEstate1.dbo.CalculateWeeklySalary();
GO
```

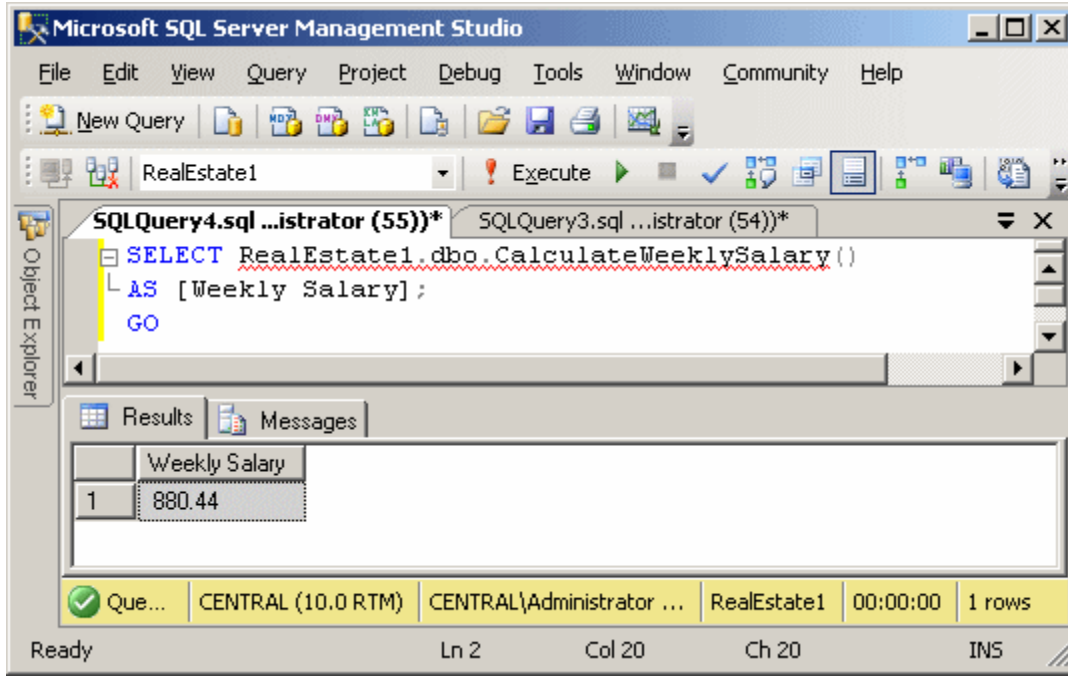
النتائج:



3- لتحديد اسم العمود الذي يعرض قيمة الدالة قم بتغيير الاستعلام على النحو التالي:

```
SELECT RealEstate1.dbo.CalculateWeeklySalary() AS [Weekly Salary];  
GO
```

النتائج:



يمكنك من على شريط الأدوات حفظ الاستعلام في ملف sql من الزر Save.

صيانة الدالة:

بما أن Transact-SQL يعتبر الدالة غرض كباقي الأغراض من قاعدة البيانات فقد تحتاج إلى الصيانة والتعديل، ونقصد بذلك: إعادة التسمية، التعديل، الحذف.

تغيير اسم الدالة:

إذا قمت بإنشاء دالة فسيتم تخزينها تحت البند Scalar-Valued Functions بالاسم الذي أعطيتها، يمكنك إذا أردت تغيير هذا الاسم مع الحفاظ على وظائف الدالة، للقيام بذلك، انقر بالزر الأيمن على اسم الدالة واختر Rename، اكتب الاسم الجديد المرغوب فيه واضغط على Enter.

حذف الدالة:

إذا قمت بإنشاء دالة ولم تعد بحاجة إليها، يمكنك حذفها بإحدى الطرق التالية:

- من متصفح الكائنات حدد الدالة التي تريد حذفها ثم انقر بالزر الأيمن على Delete، سيظهر مربع الحوار الذي يطالبك بتأكيد خيارك، انقر على OK للموافقة.
- برمجياً باستخدام SQL، اكتب العبارة DROP FUNCTION متبوعة باسم الدالة التي تريد حذفها و نفذ الاستعلام من محرر الاستعلامات.
- من متصفح الكائنات انقر بالزر الأيمن على اسم الدالة، حدد Script Function As اختر DROP To وانقر على New Query Editor Window ثم نفذ الاستعلام.
- افتح نافذة استعلام جديدة مرتبطة بقاعدة البيانات التي تحتوي على الدالة، من متصفح القوالب (Templates Explorer) قم بتوسيع البند Function ثم اسحب العنصر Drop Function إلى نافذة الاستعلام، غير البيانات الضرورية ثم نفذ الاستعلام.

تطبيق عملي: حذف دالة

1- في متصفح الكائنات حدد الدالة dbo.CalculateWeeklySalary وانقر بالزر الأيمن على Delete.

2- من مربع الحوار حذف كائن انقر على موافق (OK).

تعديل الدالة:

كما ذكرنا سابقاً في جسم الدالة عليك تحديد -بناية- الوظيفة التي ستقوم بها الدالة، كحد أدنى يجب أن تعيد الدالة ولو قيمة عدد بسيط، اكتب من اليمين الكلمة RETURN مثال:

```
CREATE FUNCTION Addition()  
RETURNS int  
BEGIN  
    RETURN 1  
END
```

وهنا يمكنك الإعلان عن متغيرات جديدة في جسم الدالة، تدعى هذه المتغيرات بالمتغيرات المحلية (Local variable)، بعد إنشاء المتغير يمكنك استخدامه كأبي متغير، مثال:

```
CREATE FUNCTION Addition()  
RETURNS int  
BEGIN  
    DECLARE @Number1 int  
    SET @Number1 = 588
```

```

RETURN @Number1 + 1450
END

```

تطبيق عملي : الإعلان عن المتغيرات المحلية في الدالة

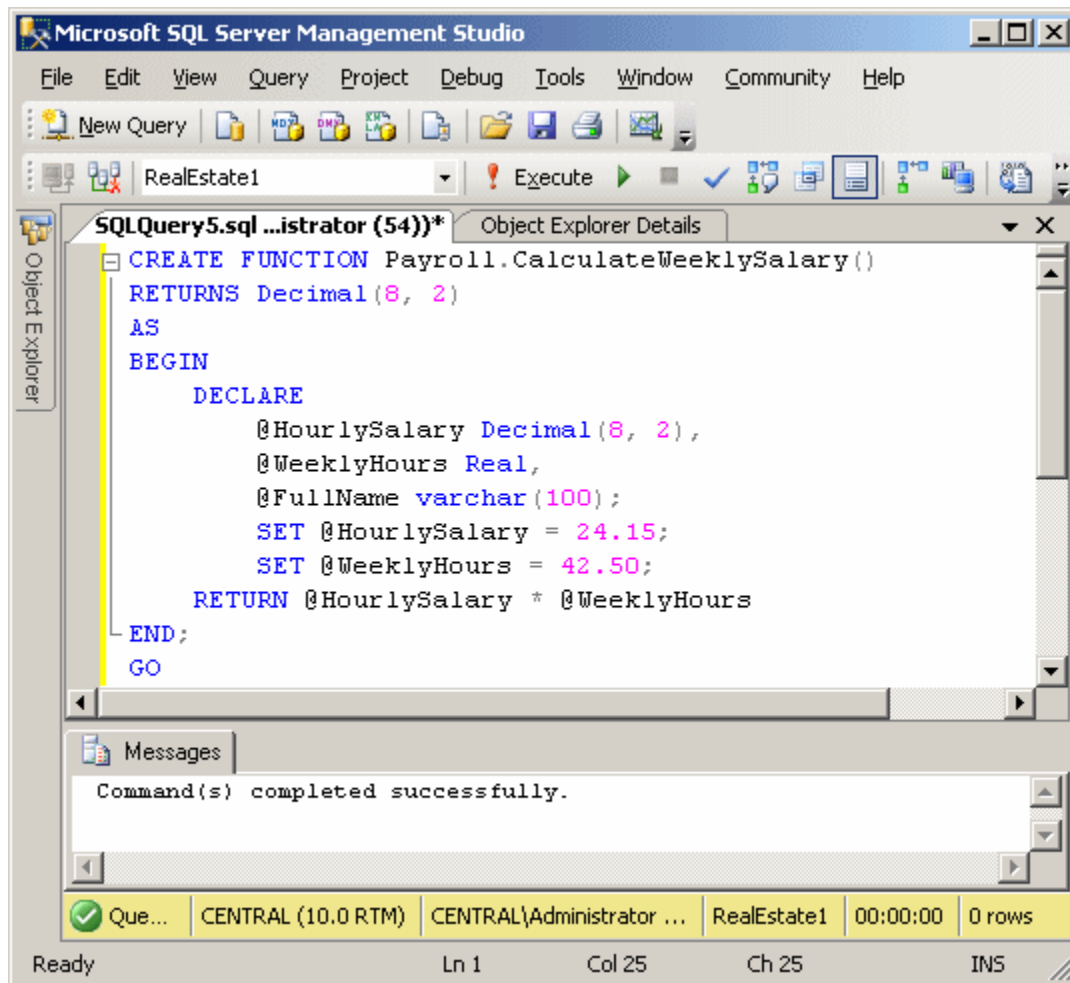
1- في نافذة محرر استعلامات، غير الاستعلام إلى ما يلي (لاحظ المخطط المستعمل، Payroll الذي أنشأناه في البداية):

```

CREATE FUNCTION Payroll.CalculateWeeklySalary()
RETURNS Decimal(8, 2)
AS
BEGIN
    DECLARE
        @HourlySalary Decimal(8, 2),
        @WeeklyHours Real,
        @FullName varchar(100);
    SET @HourlySalary = 24.15;
    SET @WeeklyHours = 42.50;
    RETURN @HourlySalary * @WeeklyHours
END;
GO

```

2- اضغط على F5 لتنفيذ الاستعلام.



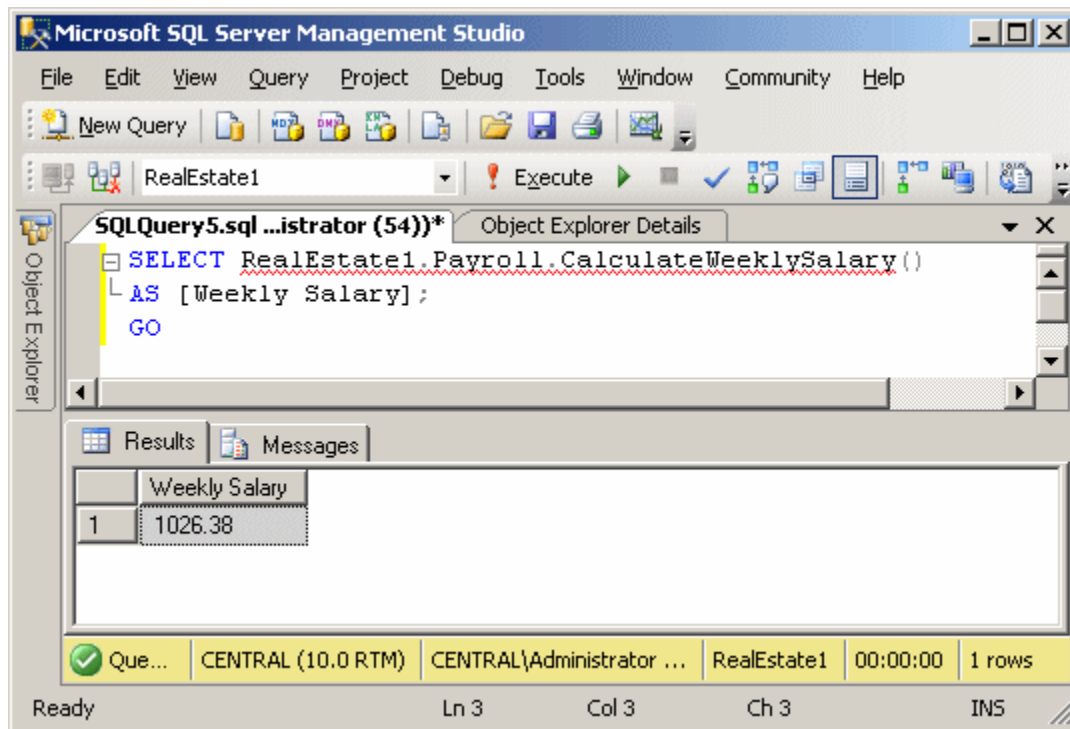
هذه الدالة لم يتم إنشاؤها على مخطط النظام الافتراضي dbo ولكن على المخطط Payroll، وبالتالي عند استدعائها أو استعمالها يجب تحديد اسم المخطط قبل كتابة اسم الدالة، مثال حالة الحذف:

```
DROP FUNCTION [Payroll].[CalculateWeeklySalary]
GO
```

3- لاستدعاء الدالة، امسح الاستعلام واكتب ما يلي:

```
SELECT RealEstate1.Payroll.CalculateWeeklySalary()
AS [Weekly Salary];
GO
```

4- اضغط على F5 لتنفيذ الاستعلام.



معالم الدالة:

لتنفيذ مهمتها يمكن أن تزود الدالة ببعض القيم وهي التي تعرف بالمعالم أو المَعْلَمَات (Arguments أو Parameters)، عند إنشاء دالة بدلا من استخدام المتغيرات المحلية قد ترغب في استخدام متغيرات خارج الدالة لأداء المهمة اللازمة، مثال، تريد إيجاد الوظيفة التي من شأنها توليد عناوين البريد الإلكتروني للموظفين، عندما يقوم المستخدم بإدخال الاسم، حينها لا

يمكنك التنبؤ بأسماء الموظفين بمن فيهم أولئك الذين لم يعينوا حتى الآن، في هذه الحالة يمكنك كتابة الدالة بكاملها ما عدا اسم الموظف الذي سيحجز له متغير ليستعمل عند طلب الوظيفة.

القيمة التي تأتي من خارج الدالة تسمى مَعْلَم، حيث يمكن للدالة أن تشتمل على أكثر من معلم، لذلك عند إنشاء دالة عليك أن تحدد عدد المعالم التي تحتاجها، مع اختيار الأنواع المناسبة لها.

الدالة ذات المعالم:

لقد رأينا أن اسم الدالة يجب أن يُتبع بقوسين، إذا كانت الدالة لا تستخدم قيم خارجية يمكن ترك القوسين فارغين، أما إذا كانت ستستخدم قيمة خارجية فعندئذ يجب عند إنشائها تحديد اسم ونوع كل معلم، المَعْلَم هو متغير وبالتالي ينشأ بالعلامة @ متبوعة باسمه ثم نوعه، مثال:

```
CREATE FUNCTION Addition(@Number1 Decimal(6,2))
```

عندما تأخذ الدالة معلما، يمكنك استخدام المعلم في جسم الدالة كما لو كنت أعلنت عنه،

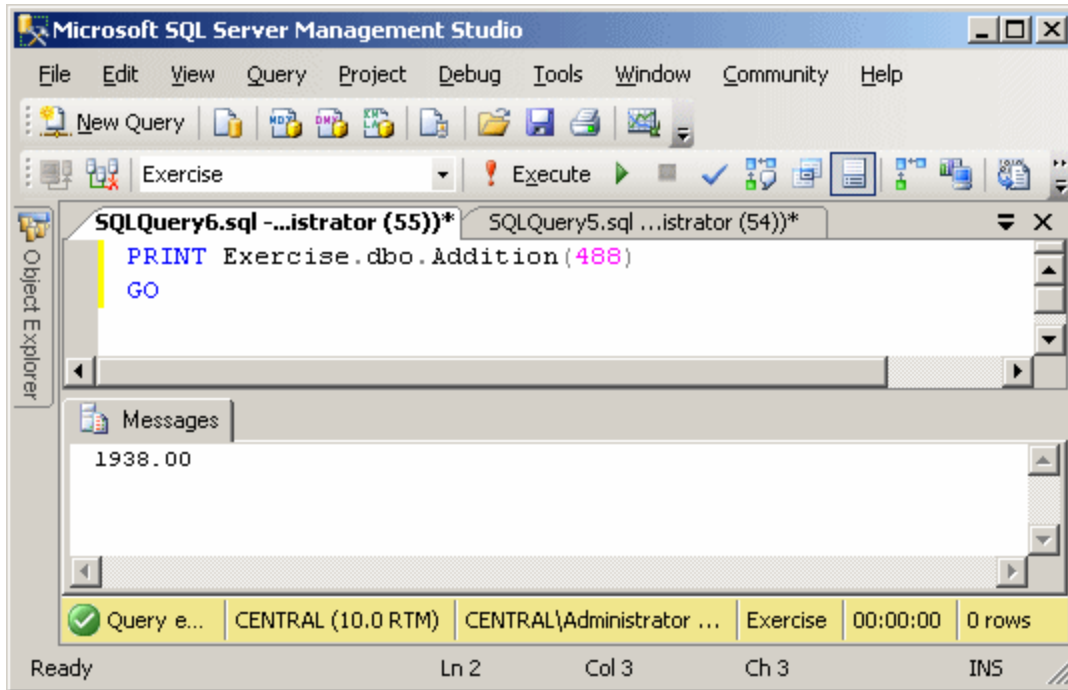
مع احترام نوع تلك القيمة، مثال:

```
CREATE FUNCTION Addition(@Number1 Decimal(6,2))
RETURNS Decimal(6,2)
BEGIN
    RETURN @Number1 + 1450
END
```

استدعاء دالة ذات معالم:

عند استدعاء دالة مزودة بمعالم، يجب إرفاق قيمة لكل معلم، لأجل ذلك اكتب قيمة المعلم

في الدالة بين قوسين، مثال:



دالة متعددة المعالم:

يمكنك إنشاء دالة تأخذ أكثر من معلم، وفي هذه الحالة يجب الفصل بين المعالم بفاصلة،
مثال:

```
CREATE FUNCTION Addition(@Number1 Decimal(6,2), @Number2
Decimal(6,2))
```

وفي جسم الدالة يمكنك استخدام المعالم كما لو كنت قد أعلنت عنها، كما يمكنك الإعلان
عن متغيرات محلية وإشراكها مع المعالم وفق ما تراه مناسباً، مثال:

```
CREATE FUNCTION Addition(@Number1 Decimal(6,2),
                          @Number2 Decimal(6,2))
RETURNS Decimal(6,2)
BEGIN
    DECLARE @Result Decimal(6,2)
    SET @Result = @Number1 + @Number2
    RETURN @Result
END;
GO
```

عند استدعاء الدالة التي تأخذ أكثر من معلم بين القوسين، يجب تحديد قيمة لكل معلم،
على نفس الترتيب الوارد في تعريف الدالة، مثال:

```
PRINT Variables1.dbo.Addition(1450, 228);
```

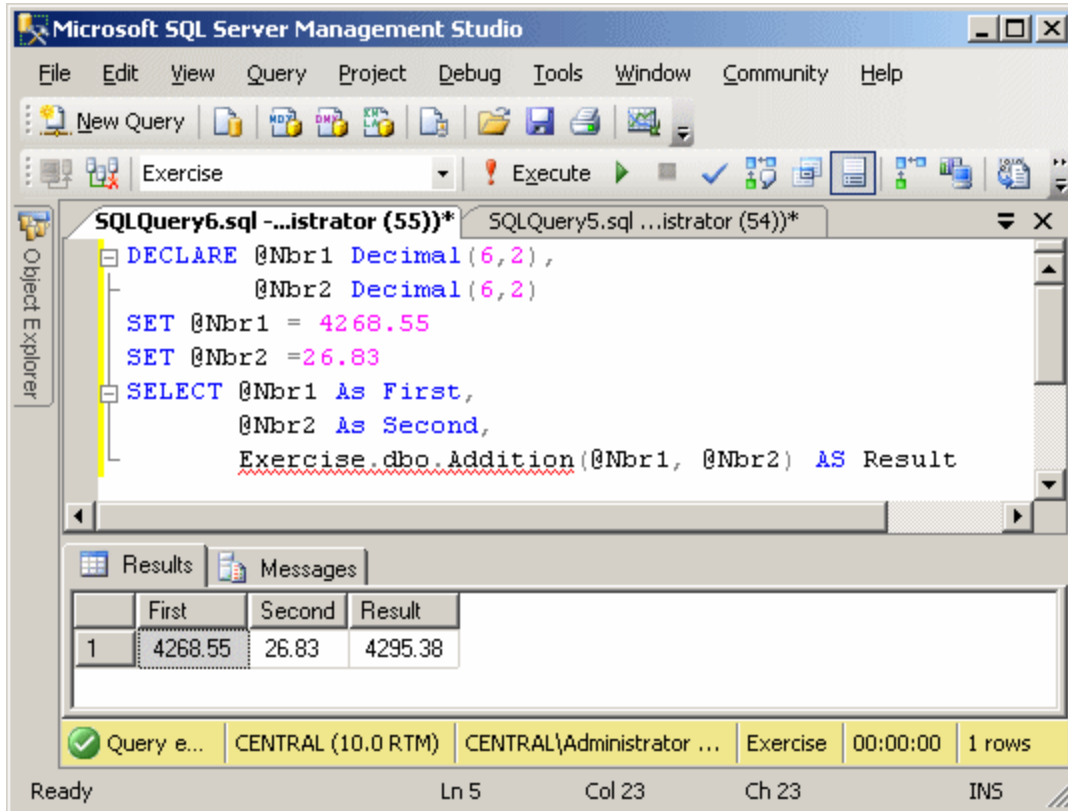
يمكنك أيضاً تمرير أسماء متغيرات سبق أن أعلنت عنها وهيئتها، مثال استدعاء الدالة
السابقة:

```

DECLARE @Nbr1 Decimal(6,2),
        @Nbr2 Decimal(6,2)
SET @Nbr1 = 4268.55
SET @Nbr2 =26.83
SELECT @Nbr1 As First,
       @Nbr2 As Second,
       Variables1.dbo.Addition(@Nbr1, @Nbr2) AS Result

```

هذا من شأنه أن ينتج ما يلي:



تطبيق عملي: إنشاء دالة بعدة معالم

- 1- في متصفح الكائنات تحت البند Payroll.CalculateWeeklySalary انقر بالزر الأيمن على Payroll.CalculateWeeklySalary وانقر Delete.
- 2- في مربع الحوار حذف الكائن، انقر على OK.
- 3- لإضافة معلم، قم بتغيير العبارة التي بين قوسين () على النحو التالي:

```

CREATE FUNCTION Payroll.CalculateWeeklySalary(@WeeklyHours
Decimal(6,2),
                                           @HourlySalary SmallMoney)
RETURNS Decimal(8, 2)
AS
BEGIN
    DECLARE @Weekly SmallMoney
    SELECT @Weekly = @WeeklyHours * @HourlySalary

```

```

RETURN @Weekly
END;
GO

```

4- اضغط على F5 لإنشاء الدالة.

5- امسح الاستعلام من المحرر واكتب ما يلي:

```

DECLARE @Hours Decimal(5,2),
        @Hourly SmallMoney
SELECT @Hours = 42.50
SELECT @Hourly = 18.62
SELECT 'Hermine Singh' As [Employee Name],
       @Hours As [Weekly Hours],
       @Hourly As [Hourly Salary],
       RealEstate1.Payroll.CalculateWeeklySalary(@Hours, @Hourly)
       AS [Weekly Salary];
GO

```

6- اضغط على F5 لتنفيذ الاستعلام.

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query in the 'SQLQuery7.sql ...istrator (54)*' tab. The query is as follows:

```

DECLARE @Hours Decimal(5,2),
        @Hourly SmallMoney
SELECT @Hours = 42.50
SELECT @Hourly = 18.62
SELECT 'Hermine Singh' As [Employee Name],
       @Hours As [Weekly Hours],
       @Hourly As [Hourly Salary],
       RealEstate1.Payroll.CalculateWeeklySalary(@Hours, @Hourly)
       AS [Weekly Salary];
GO

```

Below the query editor, the 'Results' tab is active, showing a table with the following data:

Employee Name	Weekly Hours	Hourly Salary	Weekly Salary
1 Hermine Singh	42.50	18.62	791.35

The status bar at the bottom indicates 'Query executed suc...', 'CENTRAL (10.0 RTM)', 'CENTRAL\Administrator ...', 'RealEstate1', '00:00:00', and '1 rows'.

7- أغلق نافذة الاستعلام دون حفظ الملف، من متصفح الكائنات حدد قاعدة البيانات

RealEstate1، انقر بالزر الأيمن عليها ثم اختر Delete، في نافذة التأكيد انقر OK.

ملاحظة:

في محرر الاستعلام يمكنك تحديد جزء من الاستعلام وتنفيذه بشكل منفصل كاستعلام جديد، للقيام بذلك قم بتظليل جزء من الاستعلام ثم اضغط على F5.

حوصلة الدرس:

تمارين متنوعة:

- 1- أكتب دالة لحساب الراتب السنوي للعامل إذا عُلِمَ راتب الساعة، بحيث لا تؤدي أي عملية إذا أعطي راتب الساعة بقيمة سالبة.
- 2- أكتب الدالة التي تحسب مساحة دائرة إذا تُلقت نصف قطرها.
- 3- أكتب دالة تحسب: قطر القاعدة، محيط القاعدة، مساحة القاعدة، والحجم، إذا تُلقت طول نصف القطر والارتفاع لاسطوانة.

الدوال المدمجة Built-In Functions

الدوال المدمجة الأساسية

من أهم الوظائف التي يقوم بها مصمم قواعد البيانات مساعدة المستخدمين لأداء مختلف مهامهم على النحو الذي ينبغي، ومن الأدوات المساعدة على ذلك الدوال المدمجة التي يمكن تؤدي عمليات أكثر تعقيدا. يوفر Transact-SQL مجموعة هامة من الدوال المدمجة التي تم اختبارها، كل ما تحتاجه أن تكون على علم بهذه الدوال وصيغتها، والنتائج التي تعيدها. لمساعدتك على تحديد الدوال التي يمكنك استخدامها تم تصنيفها حسب أنواعها وأحيانا فائدتها.

بعض القيم يمكن التعرف عليها بسهولة لاستقرارها إذ هي قيم قطعية ثابتة، وهناك قيم أخرى لا يمكن معرفتها مسبقا لأنها تستند إلى ظروف تتغير باستمرار، مثل تاريخ بداية العام الدراسي الذي يتغير من سنة إلى أخرى رغم أنه دائم الحصول، وهذا يعني أننا نعرف أن ذلك سيحدث ولكن لا نعرف متى سيحدث، فهذه القيم تعتبر غير القطعية.

للحصول على القيم القطعية وغير القطعية يوفر Transact-SQL فئتين من الدوال، دوال تعيد دائما نفس القيمة وهي الدوال القطعية، ودوال تعيد قيمة تتوقف على شرط معين وهي الدوال غير القطعية.

تغليف قيمة باستخدام الدالة CAST:

في معظم الحالات يرسل المستخدم لقاعدة البيانات قيم نصية (String) في أول الأمر، وهذا مناسب إذا كنت ما تتوقعه، لكن إذا تلقيت قيمة أرسلها المستخدم وليست سلسلة نصية كعدد صحيح مثلا (Integer)، فقبل استخدام هذه القيم يجب أولا تحويلها إلى النوع المناسب.

يمكنك استخدام أحد الدالتين (CAST) أو (CONVERT) لتغليف القيم، وصيغتها العامة:

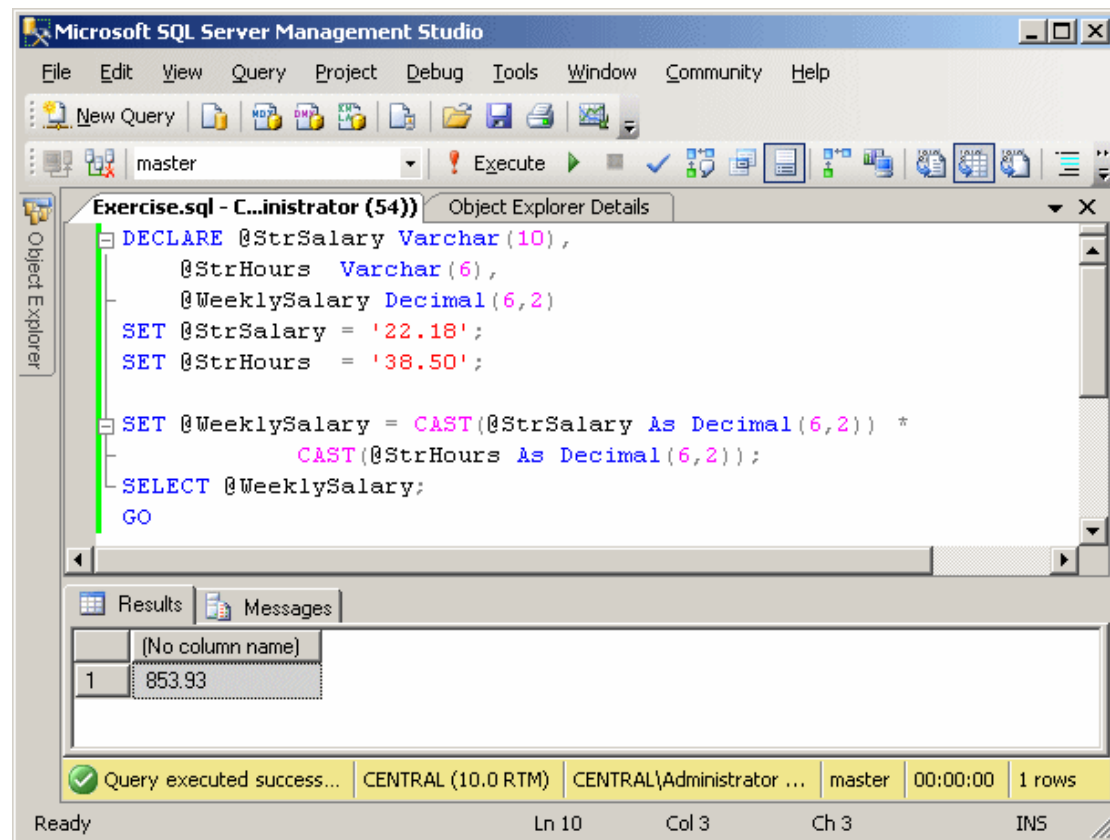
`CAST(Expression AS DataType)`

العبارة *Expression* تمثل القيمة التي المراد تحويلها، العامل *DataType* هو نوع القيمة التي تريد تحويل العبارة *Expression* إليها، نوع البيانات أحد الأنواع التي استعرضنا سابقاً، مثال:

```
DECLARE @StrSalary Varchar(10),
        @StrHours Varchar(6),
        @WeeklySalary Decimal(6,2)
SET @StrSalary = '22.18';
SET @StrHours = '38.50';

SET @WeeklySalary = CAST(@StrSalary As Decimal(6,2)) *
                    CAST(@StrHours As Decimal(6,2));
SELECT @WeeklySalary;
GO
```

وهذا تنفيذ ما ورد أعلاه:



تحويل القيمة باستخدام الدالة CONVERT:

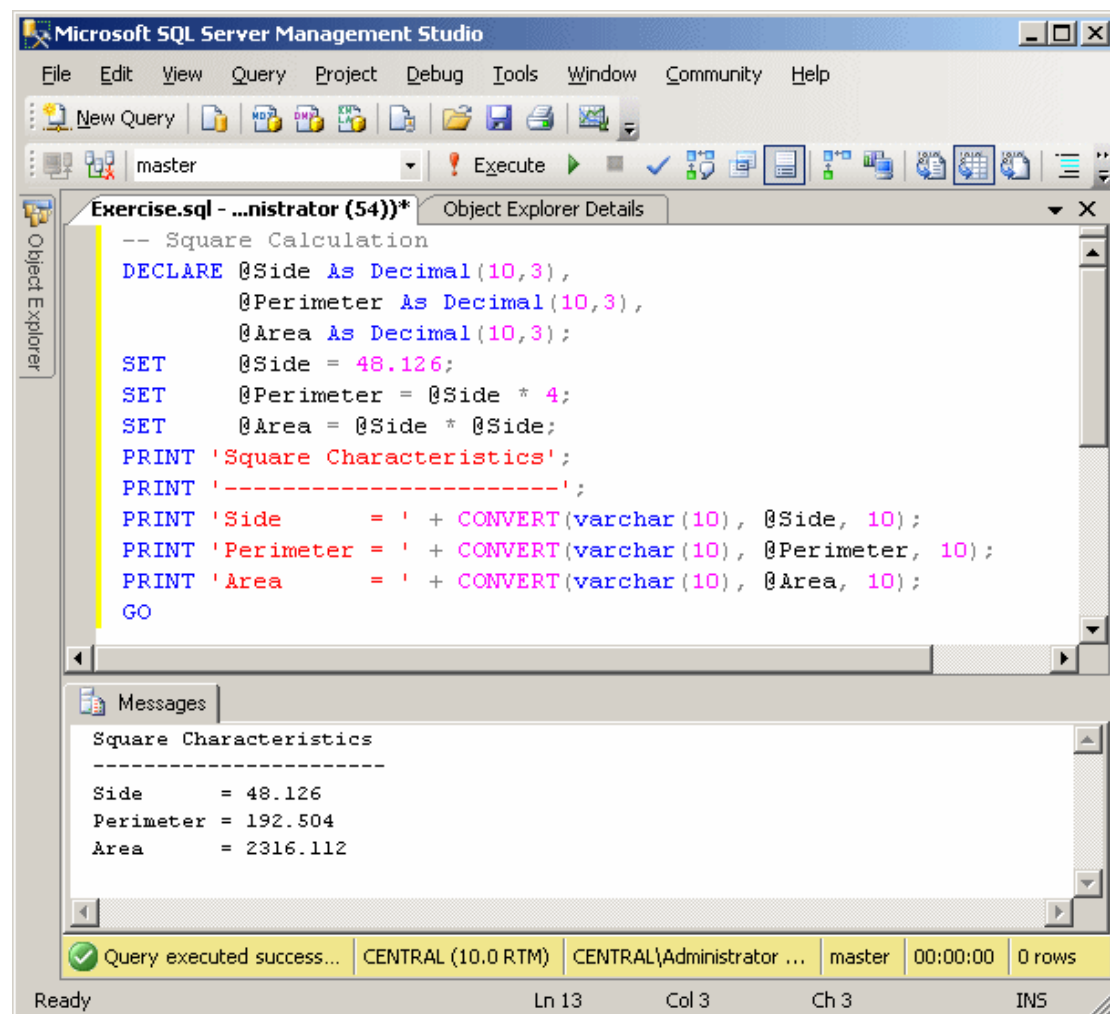
تماماً مثل الدالة *CAST()* تقوم الدالة *CONVERT()* بتحويل القيمة مع إمكانية استخدامها لتحويل القيمة من نوعها الأصلي إلى نوع محدد غير مماثل، مثلاً يمكنك استخدامها لتحويل عدد صحيح إلى سلسلة نصية والعكس، صيغتها العامة:

`CONVERT(DataType [(length)] , Expression [, style])`

العامل الأول يجب أن يكون معلوم النوع، إذا كنت تحويل قيمة في سلسلة نصية (varchar, nvarchar, char, nchar) أو النوع binary، يجب تحديد عدد الحروف المسموح به لنوع البيانات بين قوسين، كما رأينا مع الدالة CAST العبارة Expression هي التي يتم تحويلها، مثال:

```
-- Square Calculation
DECLARE @Side As Decimal(10,3),
        @Perimeter As Decimal(10,3),
        @Area As Decimal(10,3);
SET     @Side = 48.126;
SET     @Perimeter = @Side * 4;
SET     @Area = @Side * @Side;
PRINT 'Square Characteristics';
PRINT '-----';
PRINT 'Side      = ' + CONVERT(varchar(10), @Side, 10);
PRINT 'Perimeter = ' + CONVERT(varchar(10), @Perimeter, 10);
PRINT 'Area      = ' + CONVERT(varchar(10), @Area, 10);
GO
```

نتيجة الاستعلام:



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query titled "Exercise.sql - ...nistrator (54))*". The query is the same as the one shown in the previous code block. Below the query, the "Messages" window shows the output of the query execution:

```
Square Characteristics
-----
Side      = 48.126
Perimeter = 192.504
Area      = 2316.112
```

The status bar at the bottom indicates "Query executed success...", "CENTRAL (10.0 RTM)", "CENTRAL\Administrator ...", "master", "00:00:00", and "0 rows".

دوال السلاسل النصية String:

من أهم أنواع البيانات في قواعد البيانات السلسلة النصية، ذلك لأنها الاعتبار الأول لأي قيمة. للتعامل مع النصوص في SQL Server يوفر هذا الأخير توابع مختلفة لمساعدتك في إدارة قواعد البيانات وأغراضها.

السلسلة النصية هي أبسط القيم، وهي النوع الرئيسي المستخدم في عرض البيانات، ذلك لأن أي قيمة، قبل أن تعالج بشكل خاص، هي في المقام الأول تعتبر سلسلة نصية، يمكنك من خلال تطبيقك تمرير واستقبال قيم نصية على شكل ثوابت أو متغيرات، كالقيم التي أدخلها مستخدم البرنامج.

يوفر Transact-SQL مجموعة هامة من الدوال التي تسمح بالتحكم وإجراء عمليات على السلاسل النصية، ويمكن تقسيم هذه الدوال على فئات مختلفة تشمل التحويل، الإضافة... الخ.

تطبيق عملي: تمهيد لاستخدام الدوال المدمجة

- 1- شغل SQL Server Management Studio وقم بالاتصال بالخادم.
- 2- انقر بالزر الأيمن على Databases وانقر على New Database...
- 3- أكتب اسم قاعدة البيانات: Exercise1، وحدد المسار التالي:

C:\Microsoft SQL Server Database Development

- 4- انقر على OK ثم افتح البند Databases واختر قاعدة البيانات Exercise1، افتح منها البند Programmability ومنه افتح Functions.

طول السلسلة النصية:

بعض العمليات التي تجرى على السلاسل النصية تتطلب معرفة عدد حروفها، ذلك لأن بعض العمليات تتطلب حداً أدنى لعدد الأحرف، وهو ما يعرف بطول السلسلة النصية Length of String.

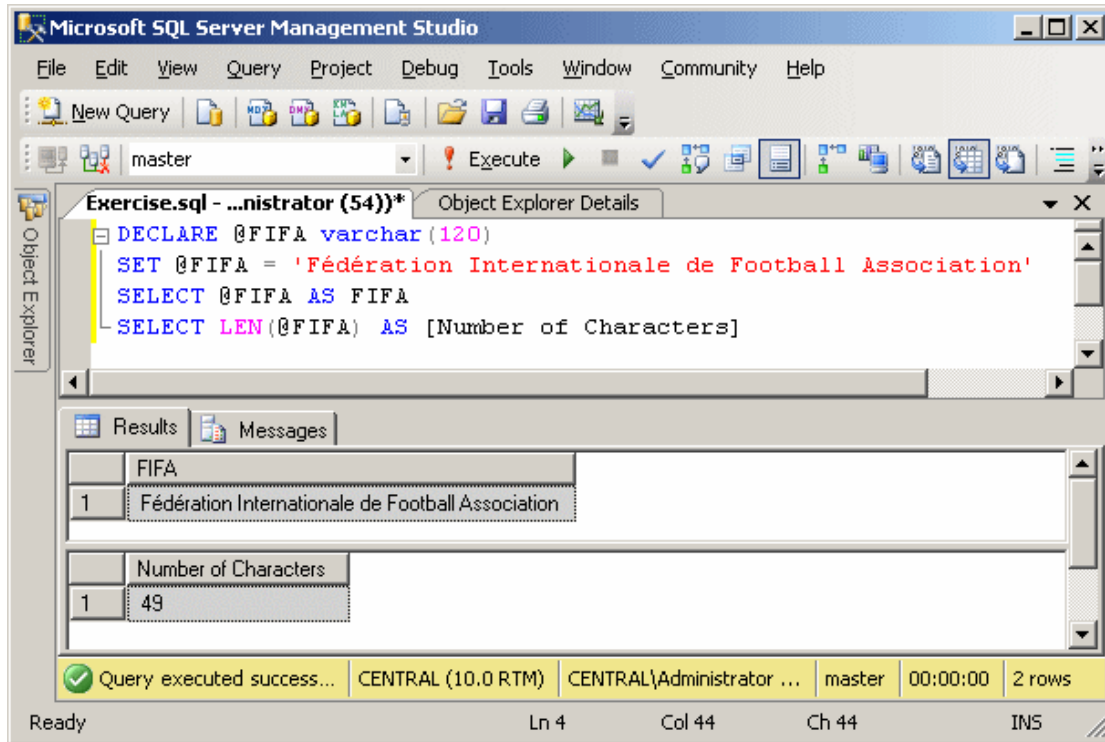
للحصول على طول سلسلة يمكنك استخدام الدالة LEN، وصيغتها:


```
int LEN(String)
```

تأخذ هذه الدالة معامل وحيد يمثل السلسلة النصية، وتعيد عدد الأحرف التي تحويها، مثال:

```
DECLARE @FIFA varchar(120)
SET @FIFA = 'Fédération Internationale de Football Association'
SELECT @FIFA AS FIFA
SELECT LEN(@FIFA) AS [Number of Characters]
```

وهذا من شأنه أن ينتج:



التحويلات على السلسلة النصية: التحويل من عدد صحيح إلى ASCII

من المعلوم أن السلسلة مكون من حرف أو مجموعة من الحروف، هذه الحروف هي قيم [ASCII](#)¹، إذا كانت لديك سلسلة نصية وتريد الحصول على رمز ASCII للحرف الأول من اليسار، يمكنك استخدام الدالة (ASCII)، وصيغتها كما يلي:

```
int ASCII(String)
```

تأخذ هذه الدالة عامل وحيد يمثل السلسلة النصية، وتعيد الرقم ASCII لأول حرف من السلسلة (من اليسار)، مثال:

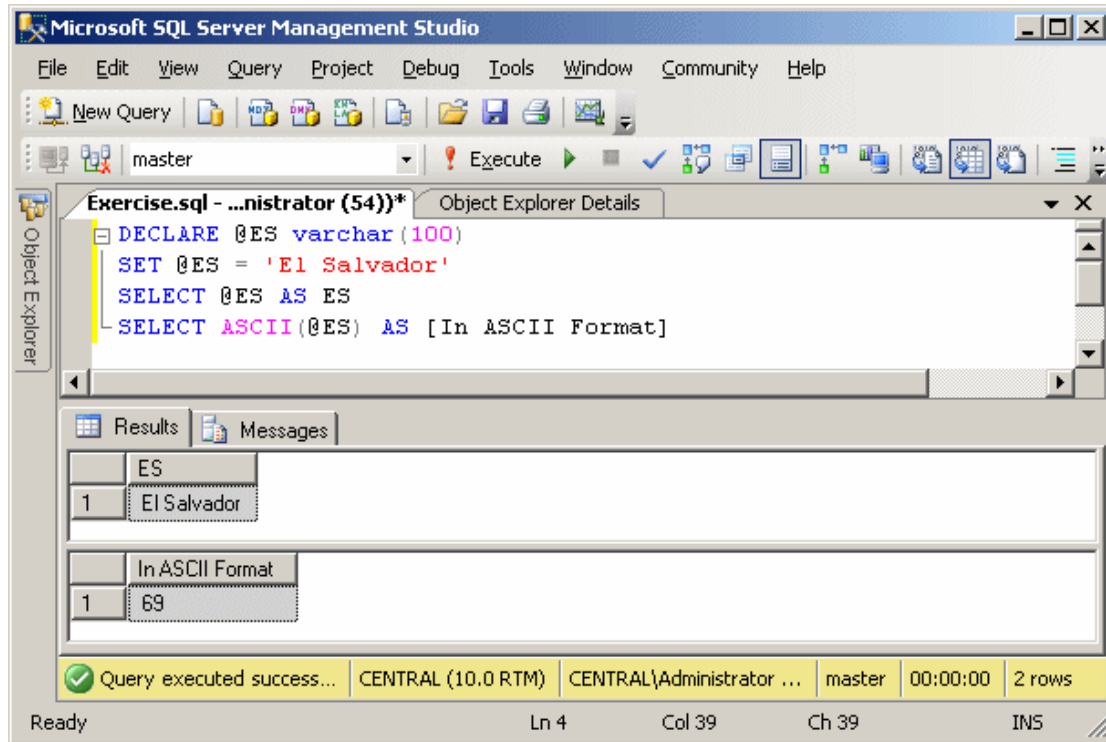
¹ ASCII أو American Standard Code for Information Interchange، هي المعيار العالمي الشهير لرموز الإعلام الآلي المتوافق مع أكثر لغات العالم.

```

DECLARE @ES varchar(100)
SET @ES = 'El Salvador'
SELECT @ES AS ES
SELECT ASCII(@ES) AS [In ASCII Format]

```

والنتيجة:



التحويل من ASCII إلى حرف:

لتحويل عدد ASCII إلى حرف وفق نظام ASCII، يمكن استخدام الدالة CHAR() بالصيغة التالية:

```
char CHAR(int value)
```

تأخذ الدالة CHAR قيمة وحيدة تمثل رمز الحرف وفق ASCII وتعيد قيمة بالحرف المطابق، مثال:

```

DECLARE @num int
SET @num = 69
SELECT CHAR(@num)
GO

```

النتيجة هي الحرف: E

التحويل إلى الحرف الصغير (Lowercase):

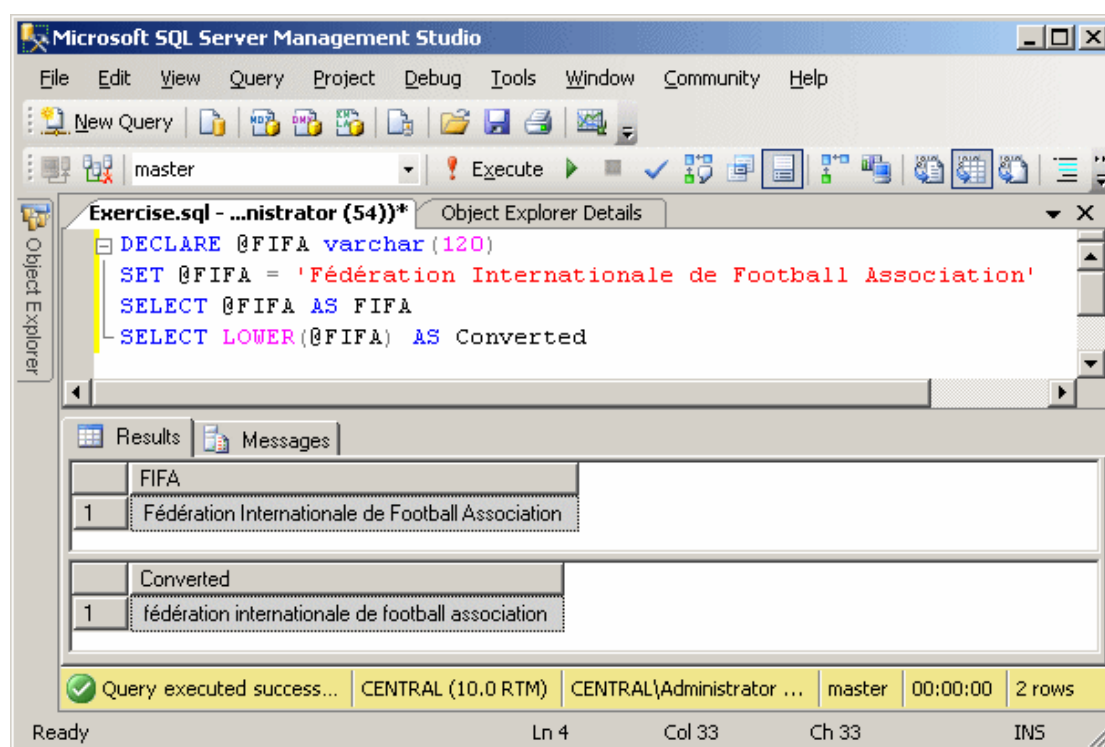
لتحويل جميع حروف سلسلة نصية (باستثناء الرموز) إلى حروف صغيرة يمكن استخدام الدالة LOWER() بالصيغة:

`varchar LOWER(String)`

تأخذ هذه الدالة سلسلة نصية وتعيدها بالأحرف الصغيرة مع الاحتفاظ بالأحرف الصغيرة والرموز على حالها، مثال:

```
DECLARE @FIFA varchar(120)
SET @FIFA = 'Fédération Internationale de Football Association'
SELECT @FIFA AS FIFA
SELECT LOWER(@FIFA) AS Converted
```

النتيجة:



تطبيق عملي: تحويل سلسلة إلى الحرف الصغير

1- من متصفح الكائنات انقر بالزر الأيمن على Exercise1 وانقر New Query...

2- لإنشاء دالة اكتب الاستعلام التالي:

```
-- =====
-- Function: GetUsername
-- =====
```

```
CREATE FUNCTION GetUsername
    (@FirstName varchar(40),
    @LastName varchar(40))
```

```

RETURNS varchar(50)
AS
BEGIN
    DECLARE @Username AS varchar(50);
    SELECT @Username = LOWER(@FirstName) + LOWER(@LastName);
    RETURN @Username;
END
GO

```

3- اضغط F5 لتنفيذ الاستعلام، ثم احفظه في ملف .sql.

4- لاستعمال الدالة التي أنشأناها احذف الاستعلام السابق واكتب ما يلي:

```

SELECT Exercise1.dbo.GetUsername('Francine', 'Moukoko');
GO

```

5- نفذ الاستعلام.

6- من متصفح الكائنات افتح قاعدة البيانات Exercice1 ثم افتح البند Programmability ثم

Functions، اختر Scalar-Valued Functions وانقر بالزر الأيمن على

dbo.GetUserName ثم انقر على Delete، وافق على الحذف بالنقر على OK.

فروع السلسلة النصية (Sub-Strings): بداية السلسلة النصية

فرع سلسلة نصية هو جزء منها، والهدف منه عزل حرف أو مجموعة من الحروف

لأداء عملية من العمليات.

الدالة LEFT() تأخذ سلسلة نصية في المعلم الأول وترجع حرف أو مجموعة من

الحروف حسب قيمة المعلم الثاني، صيغتها كما يلي:

```
varchar LEFT(String, NumberOfCharacters)
```

تطبيق عملي: إنشاء سلسلة فرعية من اليسار

1- اكتب في نافذة استعلام جديدة ما يلي:

```

-- =====
-- Function: GetUsername
-- =====

CREATE FUNCTION GetUsername
    (@FirstName varchar(40),
    @LastName varchar(40))
RETURNS varchar(50)
AS
BEGIN
    DECLARE @Username AS varchar(50);
    SELECT @Username = LOWER(LEFT(@FirstName, 1)) +
        LEFT(LOWER(@LastName), 4)

```

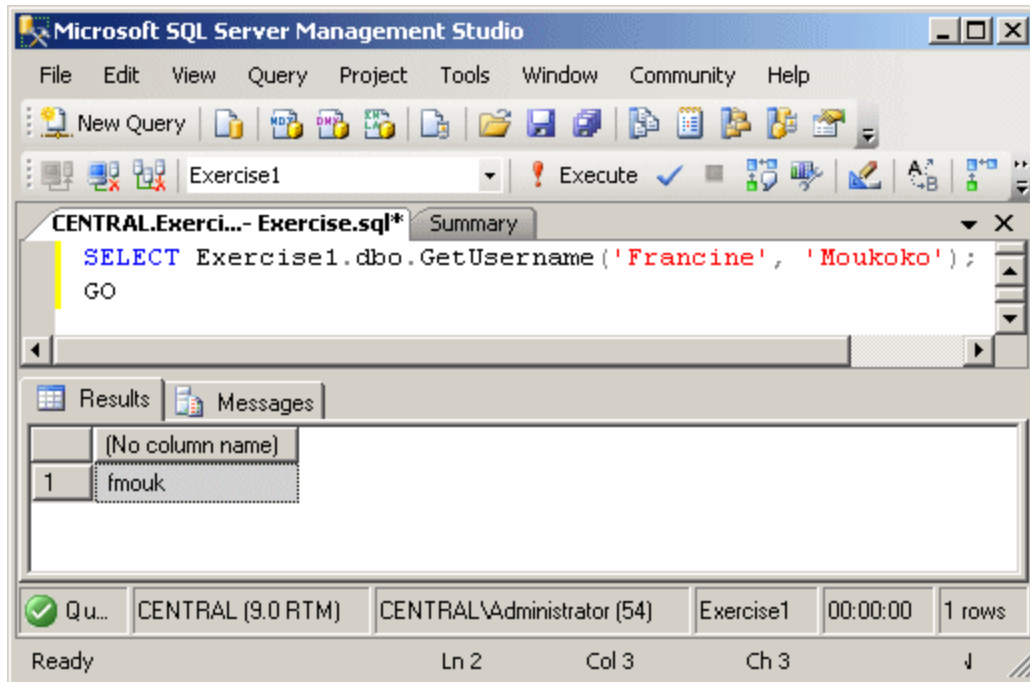
```
RETURN @Username;  
END  
GO
```

2- اضغط F5 لتنفيذ الاستعلام.

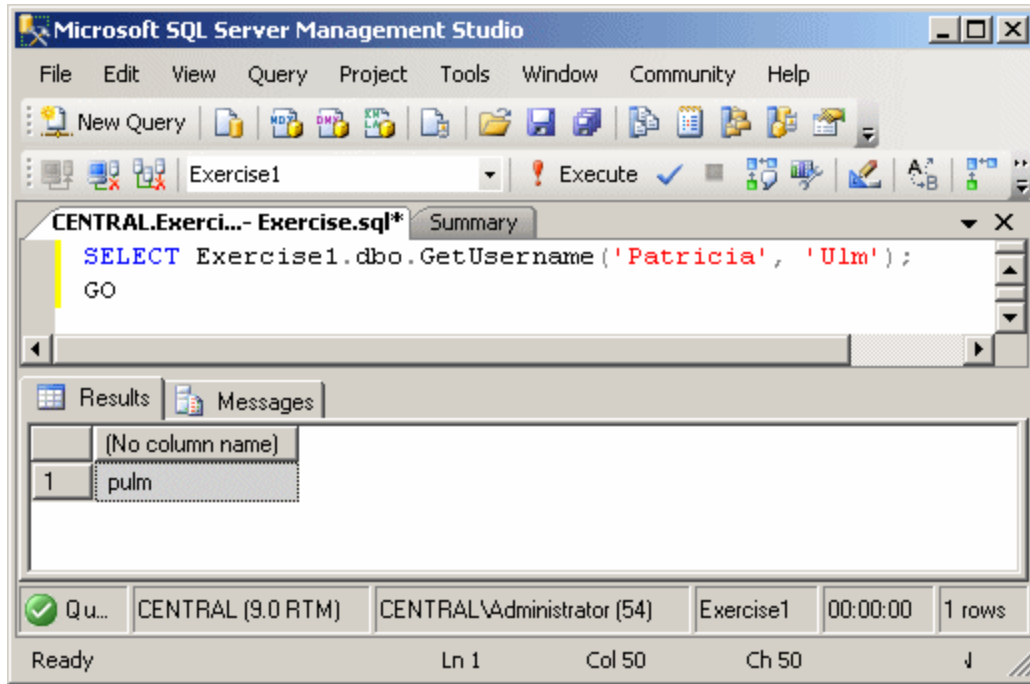
3- اسمح الاستعلام واكتب:

```
SELECT Exercisel.dbo.GetUsername('Francine', 'Moukoko');  
GO
```

4- لتنفيذ ما جاء في الاستعلام اضغط على F5:



5- غير الاسم بآخر أقل من 5 أحرف مثل "Um" ونفذ الاستعلام، مثال:



فروع السلسلة النصية (Sub-Strings): نهاية السلسلة النصية

بنفس الطريقة السابقة الدالة **RIGHT()** تأخذ كمعلم أول السلسلة النصية، وتعيد سلسلة نصية طولها يتوقف على قيمة المعلم الثاني الذي يمثل عدد الأحرف، صيغتها:

`varchar RIGHT(String, NumberOfCharacters)`

تطبيق عملي: الحصول على الحروف من اليمين

1- احذف من النافذة الاستعلام السابق، واكتب الاستعلام:

```
-- =====
-- Function: Last4DigitsOfSSN
-- =====

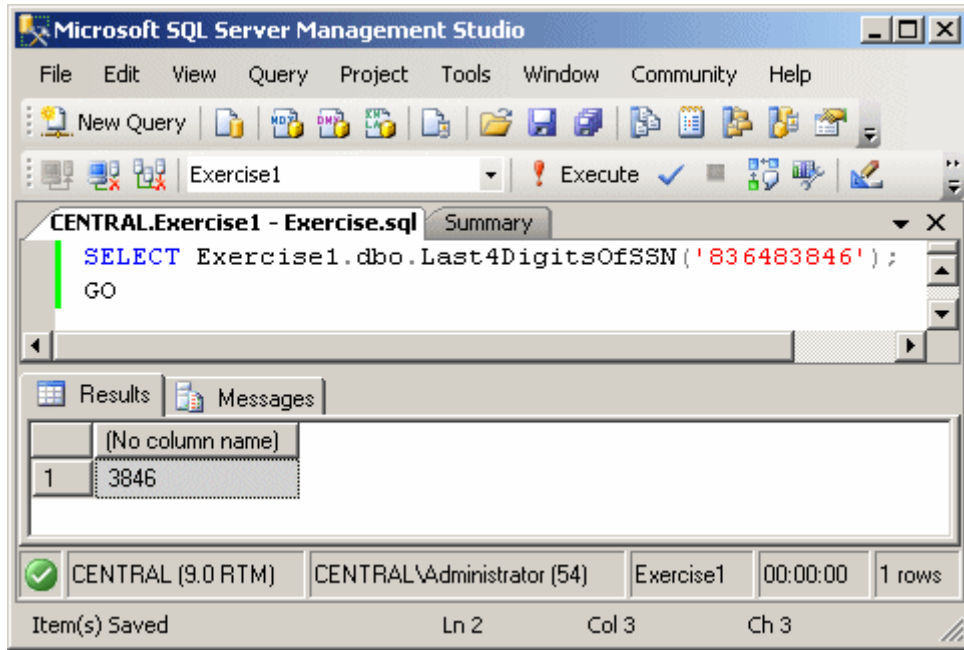
CREATE FUNCTION Last4DigitsOfSSN(@SSN varchar(12))
RETURNS char(4)
AS
BEGIN
    RETURN RIGHT(@SSN, 4);
END
GO
```

2- اضغط على F5 لتنفيذ الاستعلام.

3- امسح ما في المحرر واكتب الاستعلام التالي:

```
SELECT Exercisel.dbo.Last4DigitsOfSSN('836483846');
GO
```

4- قم بتنفيذ الاستعلام:



فروع السلسلة النصية (Sub-Strings): استبدال سلسلة من النص

كثيرا ما يحتاج المبرمج إلى استبدال جزء من سلسلة نصية بسلسلة أخرى، فمثلا يمكن إدخال رقم الهاتف على عدة صيغ: مثل 000-000-000، أو 0000000000 أو (000) 000-000، وكلها مقبولة وصحيحة ولكن لا يمكن التنبؤ بأي منها أدخل المستخدم، أحد الطرق لحل هذا الإشكال يتمثل في استبدال رمز من العبارة النصية.

لاستبدال سلسلة من أخرى سلسلة يمكنك استخدام الدالة (REPLACE)، الصيغة:

```
varchar REPLACE(String, FindString, ReplaceWith)
```

أو

```
binary REPLACE(String, FindString, ReplaceWith)
```

تأخذ هذه الدالة ثلاث معالم، الأول هو السلسلة الأم، الثاني *FindString* هو السلسلة الفرعية التي سيتم البحث عنها في السلسلة الأم، إذا وجدت السلسلة الثانية في الأولى فسيتم تعويضها بقيمة المعلم الثالث *ReplaceWith*.

تطبيق عملي: استبدال حروف أو سلسلة فرعية

1- من متصفح الكائنات اختر إنشاء استعلام جديد واكتب التالي:

```
-- =====
-- Function: Last4DigitsOfSSN
```

```

-- =====
CREATE FUNCTION Last4DigitsOfSSN(@SSN varchar(12))
RETURNS char(4)
AS
BEGIN
    DECLARE @StringWithoutSymbol As varchar(12);
    -- First remove empty spaces
    SET @StringWithoutSymbol = REPLACE(@SSN, ' ', '');
    -- Now remove the dashes "-" if they exist
    SET @StringWithoutSymbol = REPLACE(@StringWithoutSymbol, '-',
''');
    RETURN RIGHT(@StringWithoutSymbol, 4);
END
GO

```

2- نفذ الاستعلام F5

3- احذف الاستعلام من المحرر ثم اكتب:

```

SELECT Exercisel.dbo.Last4DigitsOfSSN('244-04-8502');
GO

```

4- نفذ الاستعلام.

الدوال الرياضية:

إشارة العدد:

لتحديد إشارة العدد يمكن استخدام الدالة (SIGN)، التي تعيد أحد القيم: 1 إذا كان العدد موجبا، 0 إن كان معدوماً، -1 إذا كان سالبا، وذلك وفق الصيغة:

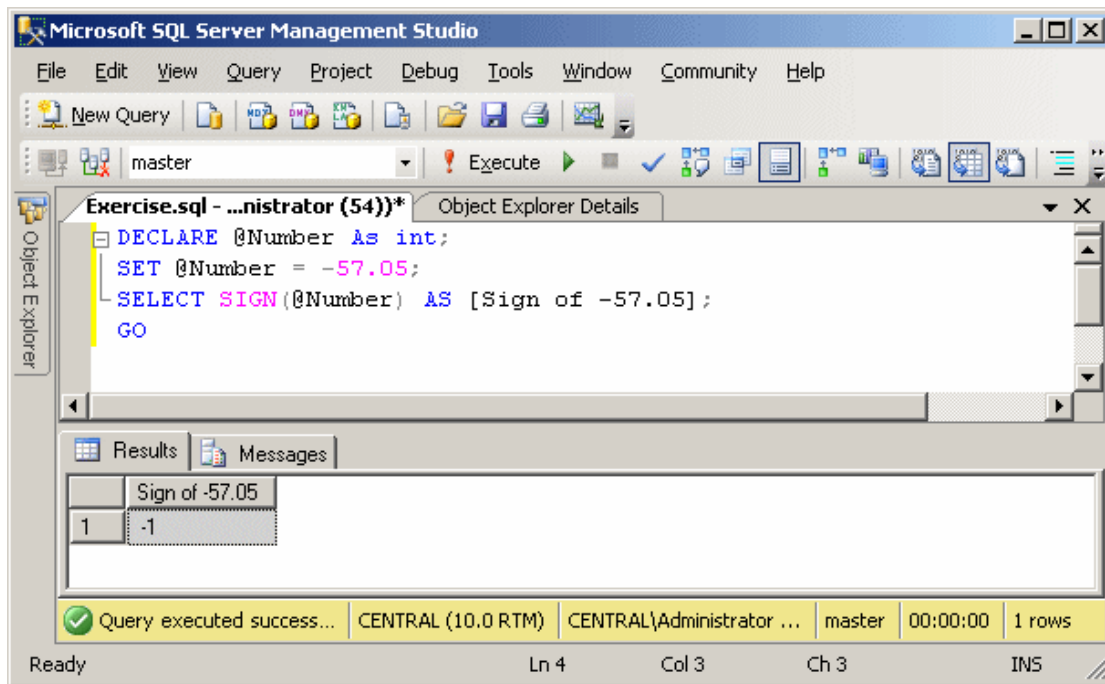
```
SIGN(Expression)
```

تأخذ هذه الدالة معلم واحد مثل العدد الذي تريد اختباره، مثال:

```

DECLARE @Number As int;
SET @Number = -57.05;
SELECT SIGN(@Number) AS [Sign of -57.05];
GO

```

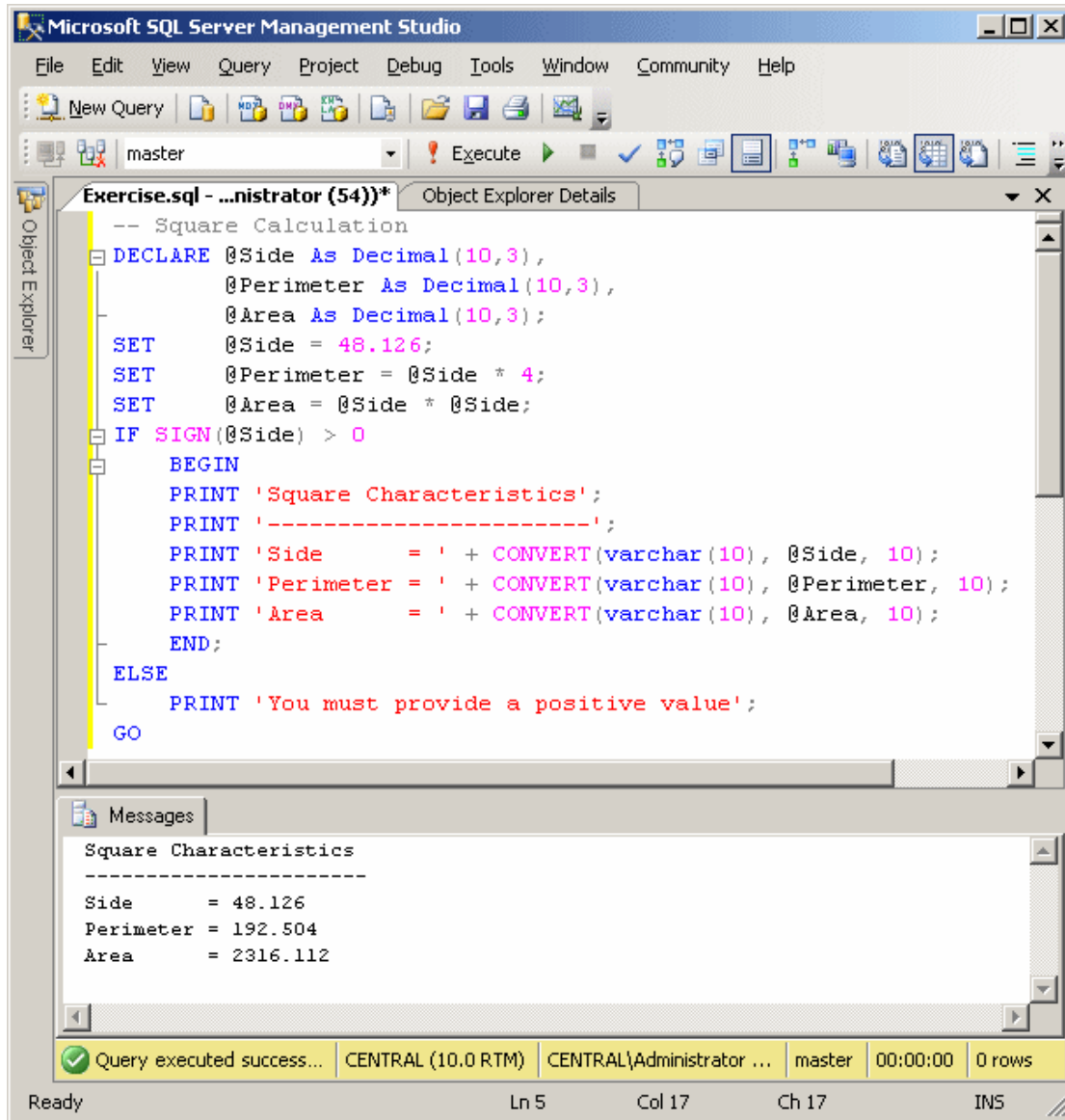
يمكن إدراجها في عبارة شرطية IF، للتحقق إشارة متغير، مثال:

```

-- Square Calculation
DECLARE @Side As Decimal(10,3),
        @Perimeter As Decimal(10,3),
        @Area As Decimal(10,3);
SET     @Side = 48.126;
SET     @Perimeter = @Side * 4;
SET     @Area = @Side * @Side;
IF SIGN(@Side) > 0
    BEGIN
        PRINT 'Square Characteristics';
        PRINT '-----';
        PRINT 'Side      = ' + CONVERT(varchar(10), @Side, 10);
        PRINT 'Perimeter = ' + CONVERT(varchar(10), @Perimeter, 10);
        PRINT 'Area      = ' + CONVERT(varchar(10), @Area, 10);
    END;
ELSE
    PRINT 'You must provide a positive value';
GO

```

بعد تنفيذ الاستعلام:



القيمة المطلقة لعدد صحيح:

القيمة المطلقة لعدد هي التعبير عنه بقيمة موجبة أو معدومة، وتستخدم الدالة $ABS()$ لحسابها وفق الصيغة:

$ABS(\text{Expression})$

تأخذ هذه الدالة قيمة عددية وتعيد قيمتها المطلقة، مثال:

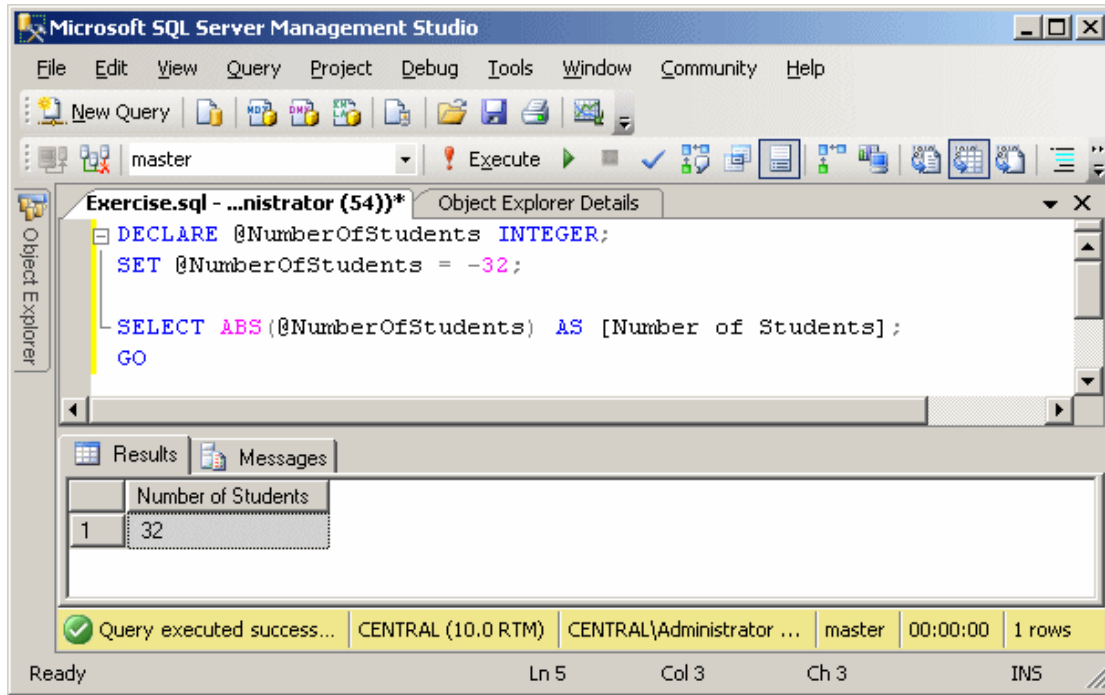
```

DECLARE @NumberOfStudents INTEGER;
SET @NumberOfStudents = -32;

SELECT ABS(@NumberOfStudents) AS [Number of Students];
GO

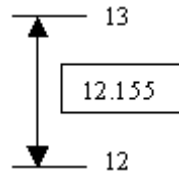
```

الحاصل:



القيمة المقربة بالزيادة لعدد عشري:

بالنظر إلى القيمة العشرية 12.155، نجد أنها تنحصر بين القيمتين الصحيحتين 12 و 13.



وأيضا القيمة العشرية -24.06 تنحصر بين القيمتين الصحيحتين -24 و -25، وهي أقرب إلى القيمة الصحيحة -24، فالقيمة المقربة بالزيادة هي أكبر أقرب قيمة صحيحة من العدد العشري، في الحالة الأولى (12.155) القيمة المقربة بالزيادة هي 13، وفي الحالة الثانية القيمة المقربة بالزيادة للقيمة -24.06 هي -24.

للحصول على القيمة المقربة بالزيادة لعدد عشري، يمكن استخدام الدالة `CEILING()` بالشكل:

`CEILING(Expression)`

تأخذ الدالة `CEILING` متغير وحيد يمثل القيمة العشرية وتعيد عدد صحيح بالقيمة المقربة بالزيادة، مثال:

`DECLARE @Number1 As Numeric(6, 2),`

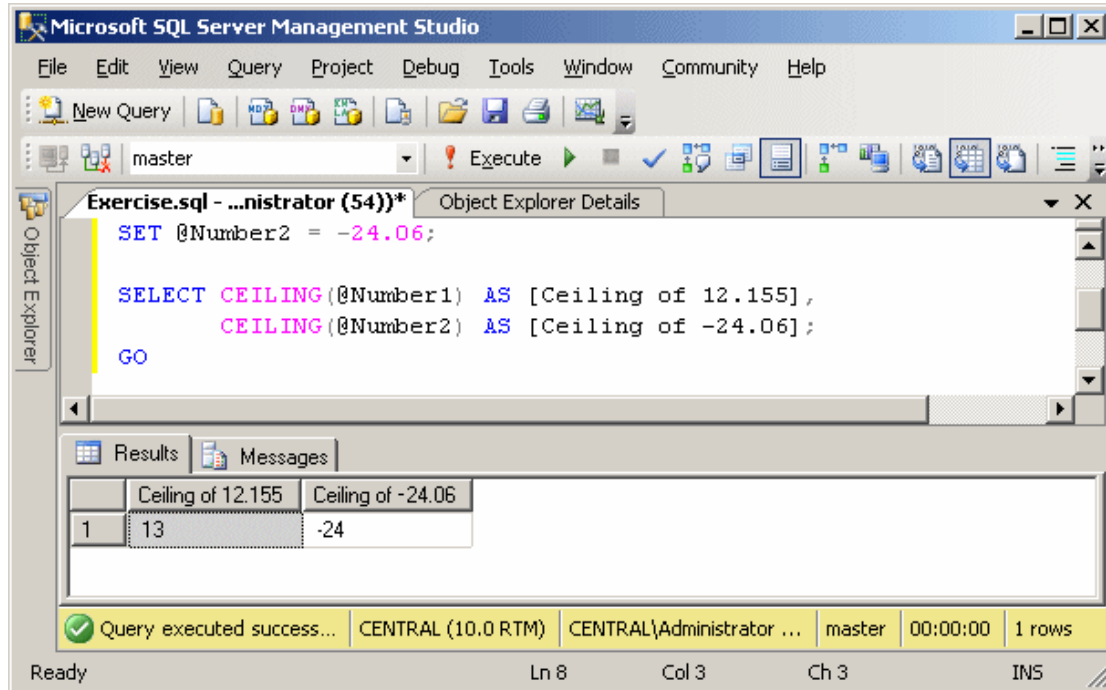
```

        @Number2 As Numeric(6, 2)
SET @Number1 = 12.155;
SET @Number2 = -24.06;

SELECT CEILING(@Number1) AS [Ceiling of 12.155],
       CEILING(@Number2) AS [Ceiling of -24.06];
GO

```

نتائج الاستعلام:



مثال بطريقة أخرى:

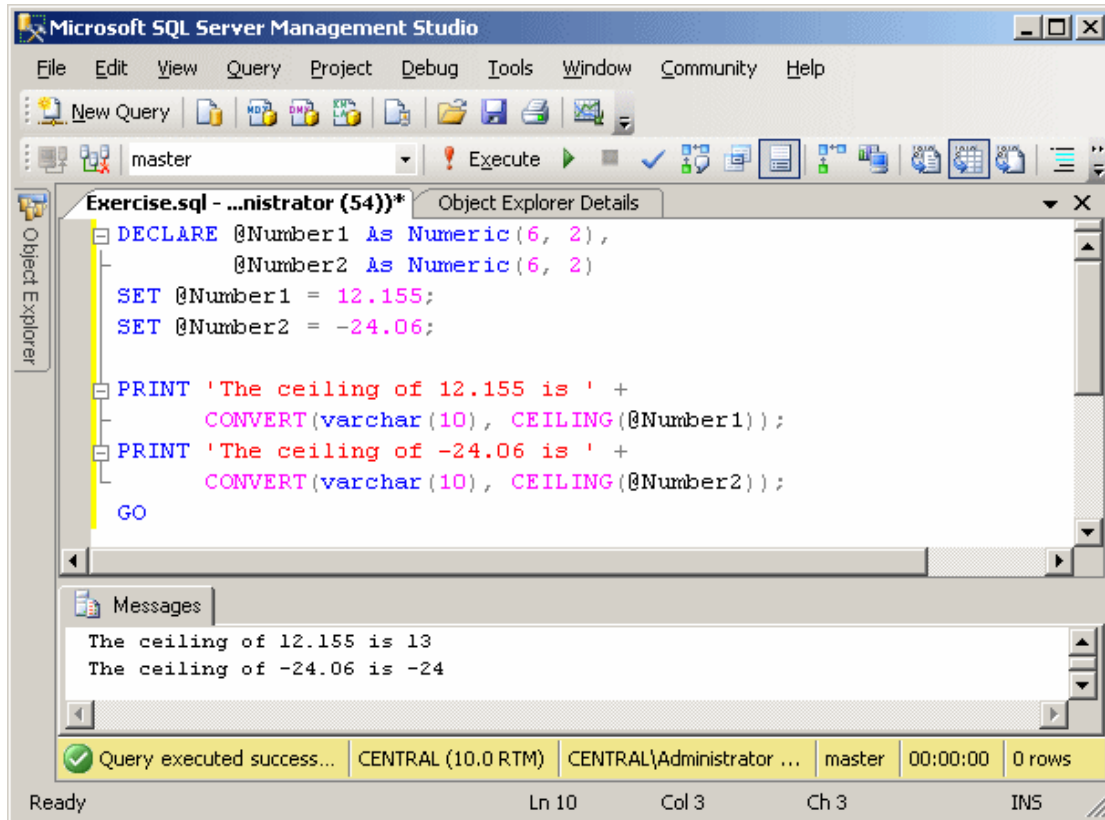
```

DECLARE @Number1 As Numeric(6, 2),
        @Number2 As Numeric(6, 2)
SET @Number1 = 12.155;
SET @Number2 = -24.06;

PRINT 'The ceiling of 12.155 is ' +
      CONVERT(varchar(10), CEILING(@Number1));
PRINT 'The ceiling of -24.06 is ' +
      CONVERT(varchar(10), CEILING(@Number2));
GO

```

نتائج المثال:



القيمة المقربة بالنقصان لقيمة عشرية:

بنفس طريقة الدالة السابقة، تعيد الدالة **FLOOR()** القيمة المقربة بالنقصان لعدد عشري، بحيث تأخذ الدالة متغير وحيد يمثل القيمة العشرية وتعيد القيمة المقربة في عدد صحيح، شكلها العام:

FLOOR(Expression)

مثال:

```

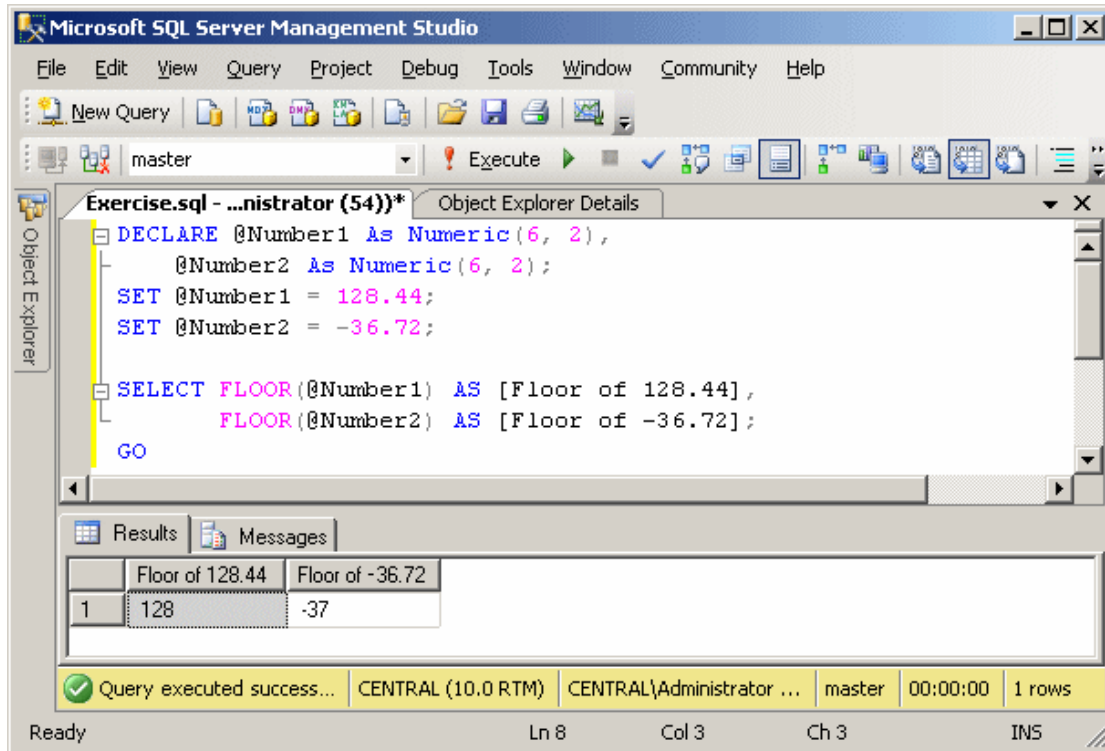
DECLARE @Number1 As Numeric(6, 2),
        @Number2 As Numeric(6, 2);
SET @Number1 = 128.44;
SET @Number2 = -36.72;

SELECT FLOOR(@Number1) AS [Floor of 128.44],
       FLOOR(@Number2) AS [Floor of -36.72];

GO

```

الناتج:



قوة عدد (الأس) لأساس اللوغاريتم النيبيري (e^x):

لحساب قوة عدد بأساس اللوغاريتم النيبيري (هـ) تستخدم الدالة **EXP()** بصيغتها العامة:

EXP(Expression)

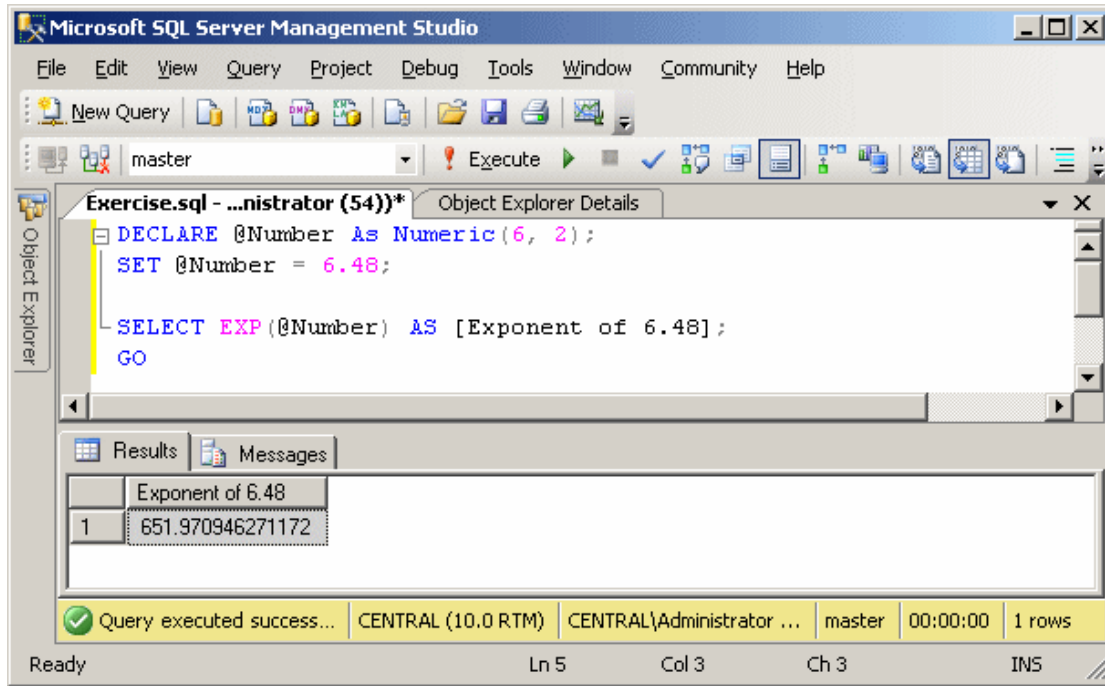
تأخذ الدالة متغير عددي **x** (عشري أم صحيح) وحيد وتعيد القيمة e^x ، مثال:

```

DECLARE @Number As Numeric(6, 2);
SET @Number = 6.48;
SELECT EXP(@Number) AS [Exponent of 6.48];
GO

```

نتيجة الاستعلام:



قوة عدد (الأس) x^y :

القيمة x^y تمثل مجموع جداءات العدد x مضروباً في نفسه y مرة، تستخدم لحسابها الدالة

POWER() وفق الصيغة التالية:

POWER(x, y)

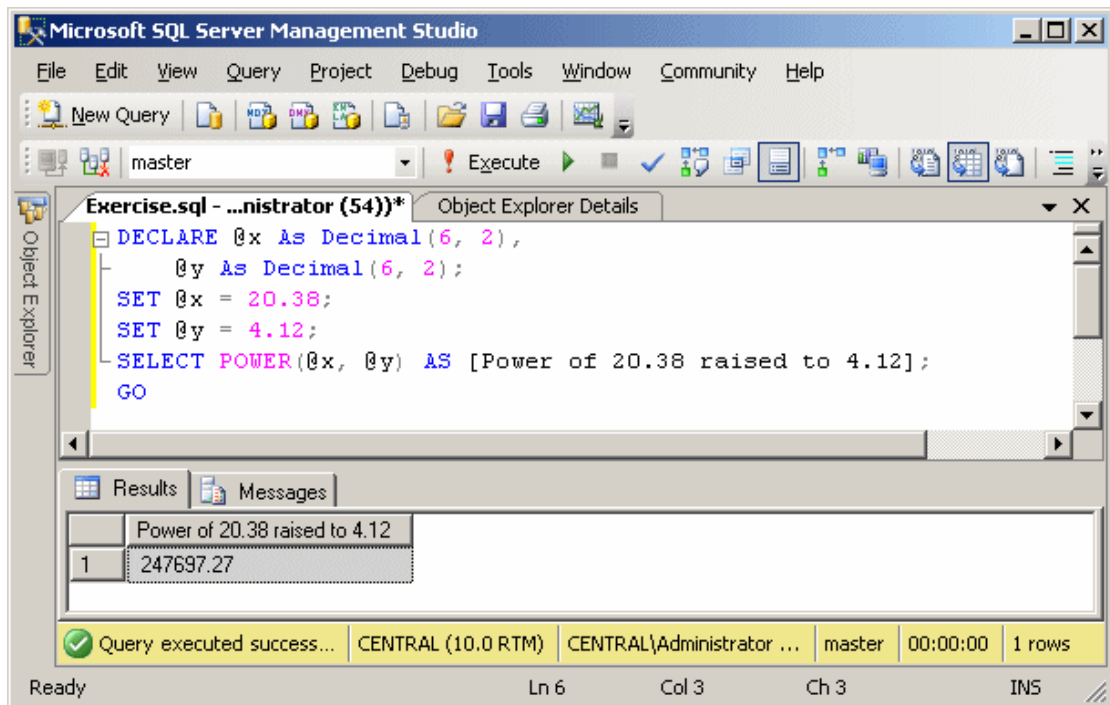
تأخذ الدالة **POWER** قيمتين عدديتين وتعيد قيمة عددية وحيدة، مثال:

```

DECLARE @x As Decimal(6, 2),
        @y As Decimal(6, 2);
SET @x = 20.38;
SET @y = 4.12;
SELECT POWER(@x, @y) AS [Power of 20.38 raised to 4.12];
GO

```

النتائج:



اللوغاريتم الطبيعي لعدد LOG¹:

لحساب اللوغاريتم الطبيعي لمتغير عدد يمكن استخدام الدالة LOG() وفق الصيغة:

LOG(Expression)

تأخذ الدالة LOG قيمة عددية وحيدة وتعيد القيمة اللوغاريتم الطبيعي لها، مثال:

```

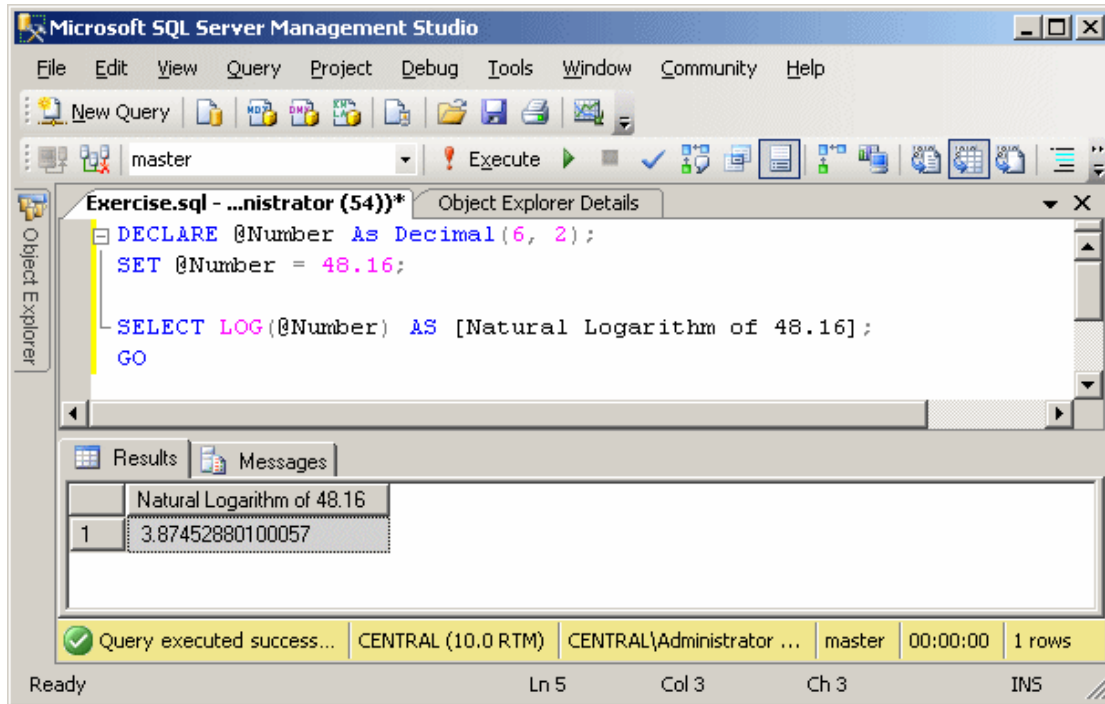
DECLARE @Number As Decimal(6, 2);
SET @Number = 48.16;

SELECT LOG(@Number) AS [Natural Logarithm of 48.16];
GO

```

نتائج الاستعلام:

¹ LOG هو اللوغاريتم النيبيري الذي أساسه القيمة e، حيث $e \approx 2.17828$ ، ويرمز له أيضا Ln.



لوغاريتم الأساس 10¹:

لحساب لوغاريتم الأساس 10 لمتغير عددي تستخدم الدالة LOG10 وفق الصيغة:

LOG10(Expression)

تأخذ الدالة قيمة عددية وحيدة وتعيد قيمتها وفق اللوغاريتم ذو الأساس 10، مثال:

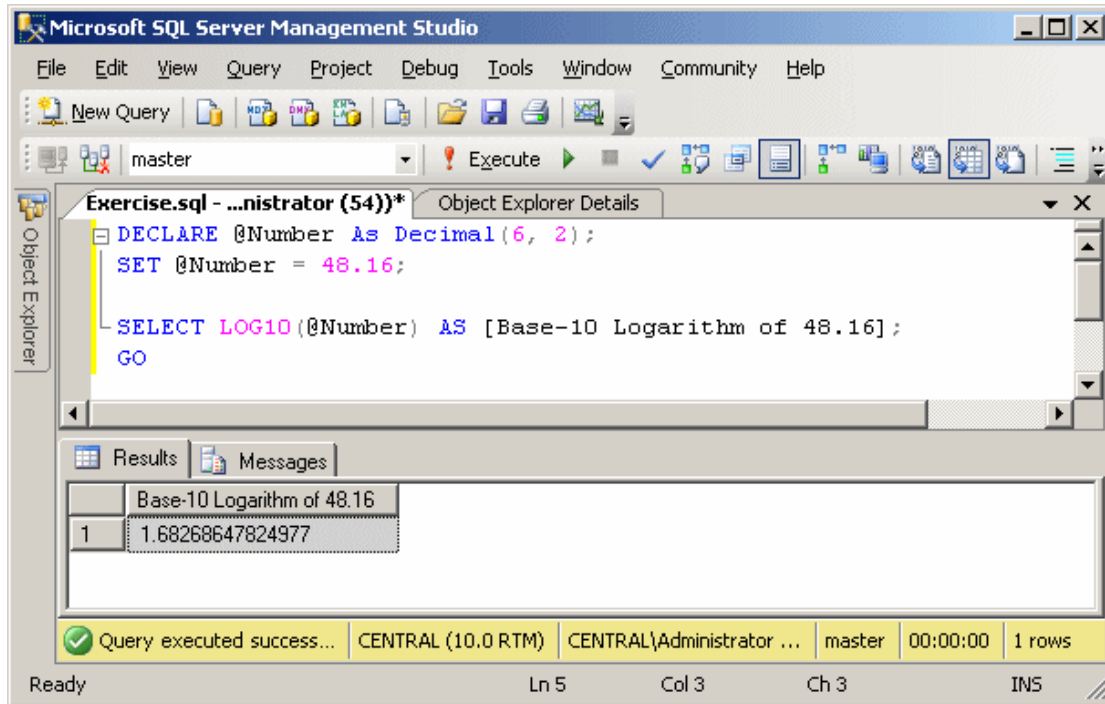
```

DECLARE @Number As Decimal(6, 2);
SET @Number = 48.16;
SELECT LOG10(@Number) AS [Base-10 Logarithm of 48.16];
GO

```

ناتج المثال:

¹ ويرمز له أيضا Log، حيث العبارتان التاليتان متكافئتان: $x = 10^y \Leftrightarrow y = \log_{10}x$.



الجذر التربيعي لعدد موجب:

لحساب الجذر التربيعي لأي عدد موجب تستخدم الدالة **SQRT()** وفق الصيغة:

SQRT(Expression)

تأخذ الدالة **SQRT** عدد عشري **موجب** وتعيد جذره التربيعي، مثال:

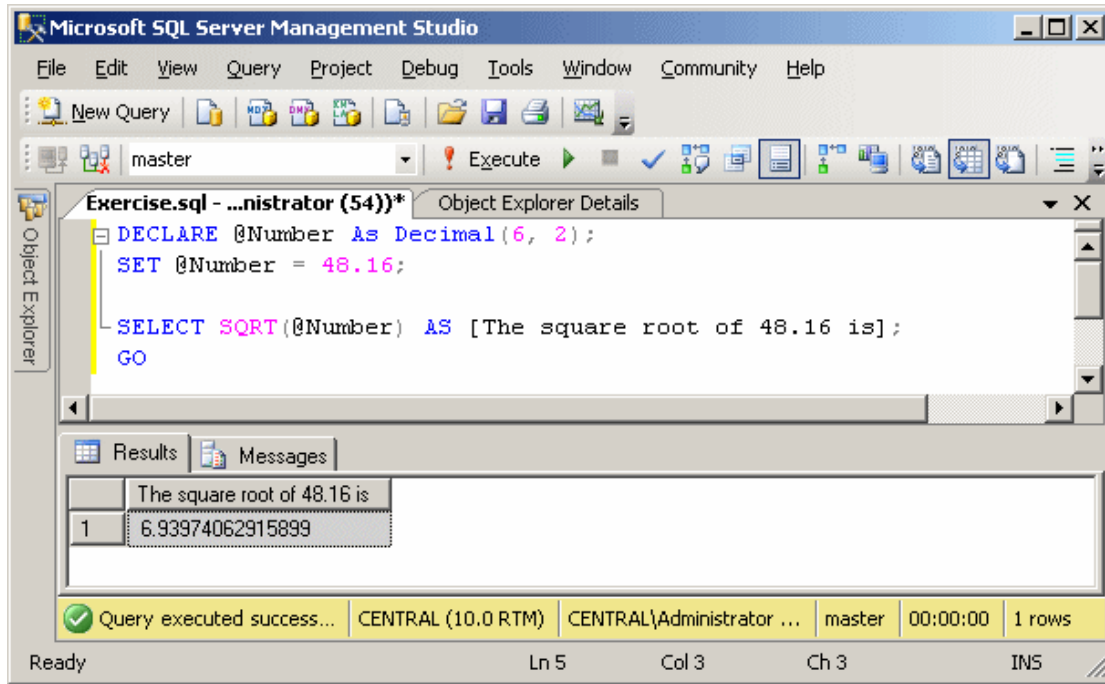
```

DECLARE @Number As Decimal(6, 2);
SET @Number = 48.16;

SELECT SQRT(@Number) AS [The square root of 48.16 is];
GO

```

وهذا من شأنه أن ينتج:



وإذا كان العدد سالبا، فإنك ستحصل على خطأ، وفي هذه الحالة يستحسن فحص القيمة لمعرفة إشارة العدد، مثال:

```

DECLARE @Number As Decimal(6, 2);
SET @Number = 258.4062;

IF SIGN(@Number) > 0
    PRINT 'The square root of 258.4062 is ' +
        CONVERT(varchar(12), SQRT(@Number));
ELSE
    PRINT 'You must provide a positive number';
GO

```

بعد تنفيذ الاستعلام نحصل على:

The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL query in a text editor. The query is as follows:

```

DECLARE @Number As Decimal(6, 2);
SET @Number = 258.4062;

IF SIGN(@Number) > 0
    PRINT 'The square root of 258.4062 is ' +
        CONVERT(varchar(12), SQRT(@Number));
ELSE
    PRINT 'You must provide a positive number';
GO

```

Below the query editor, the Messages pane shows the output of the query execution:

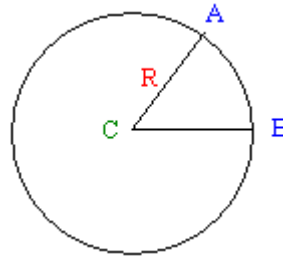
```

The square root of 258.4062 is 16.0751

```

The status bar at the bottom indicates that the query was executed successfully, with a status of 'Ready', line 9, column 3, character 3, and 0 rows affected.

دوال القياسات الهندسية:



الدائرة هي مجموعة من النقاط المتقابلة بنفس البعد عن نقطة مميزة تمثل المركز C، تسمى المسافة من المركز C إلى أحد هذه النقاط بنصف القطر R، الخط الذي يربط بين جميع النقاط المتساوية إلى المركز يسمى محيط الدائرة، قطر الدائرة هي المسافة بين نقطتين متقابلتين من محيط الدائرة.

محيط الدائرة مقسم إلى 360 جزء متساو، يسمى كل منها الدرجة °، تقاس المسافة بين نقطتين من محيط الدائرة بالدرجة، ويسمى الخط الذي يربط نقطتين من المحيط بالقوس، والزاوية θ هي نسبة المسافة بين النقطتين A و B من الدائرة إلى نصف قطرها R، يمكن كتابتها على النحو التالي:

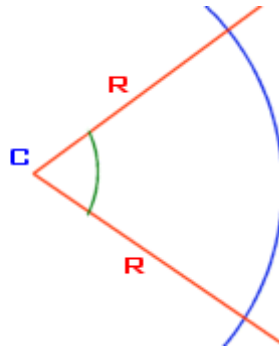
$$\theta = \frac{AB}{R}$$

العدد Pi:

يستخدم الحرف π في العديد من العمليات الرياضية، وقيمته التقريبية هي 3.1415926535897932، للحصول على قيمة مقربة للعدد π من خلال Transact-SQL تستخدم الدالة:

`PI()`

الراديان:



الزاوية هي نسبة قوس دائرة إلى نصف قطرها، ويفضل حساب قياس الزوايا بالراديان، فإذا كنت تعرف قيمة الزاوية بالدرجة وتريد الحصول قياسها بالراديان استخدم الدالة **RADIANS()**، وفق الصيغة:

`RADIANS(Expression)`

تأخذ هذه الدالة قيمة الزاوية بالدرجة وتعيد قيمتها بالراديان.

القياس بالدرجات:

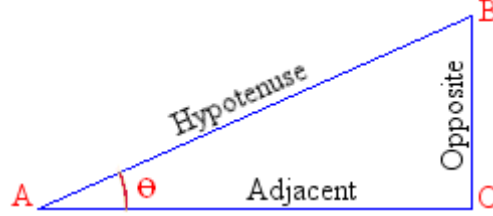
وإذا كنت تعرف قيمة الزاوية بالراديان وتريد الحصول عليها بالدرجات، يمكنك استخدام درجات الدالة **DEGREES()**، صيغتها:

`DEGREES(Expression)`

تأخذ الدالة متغير عددي يمثل قياس الزاوية بالراديان وتعيد قيمته بالدرجة الزاوية.

الدوال المثلثية:

جيب تمام الزاوية (Cosine):



ليكن AB طول وتر (hypotenuse) المثلث ABC، و AC هو الضلع المجاور (adjacent) للزاوية θ ، و BC هو الضلع المقابل لها (Opposite)، تسمى النسبة AC/AB بجيب تمام (ويكتب تجب Cos) الزاوية θ وقيمتها محصورة بين -1 و 1، وتحسب بالعلاقة:

$$\cos \theta = \frac{AC}{AB} = \frac{\text{Adjacent}}{\text{Hypotenuse}}$$

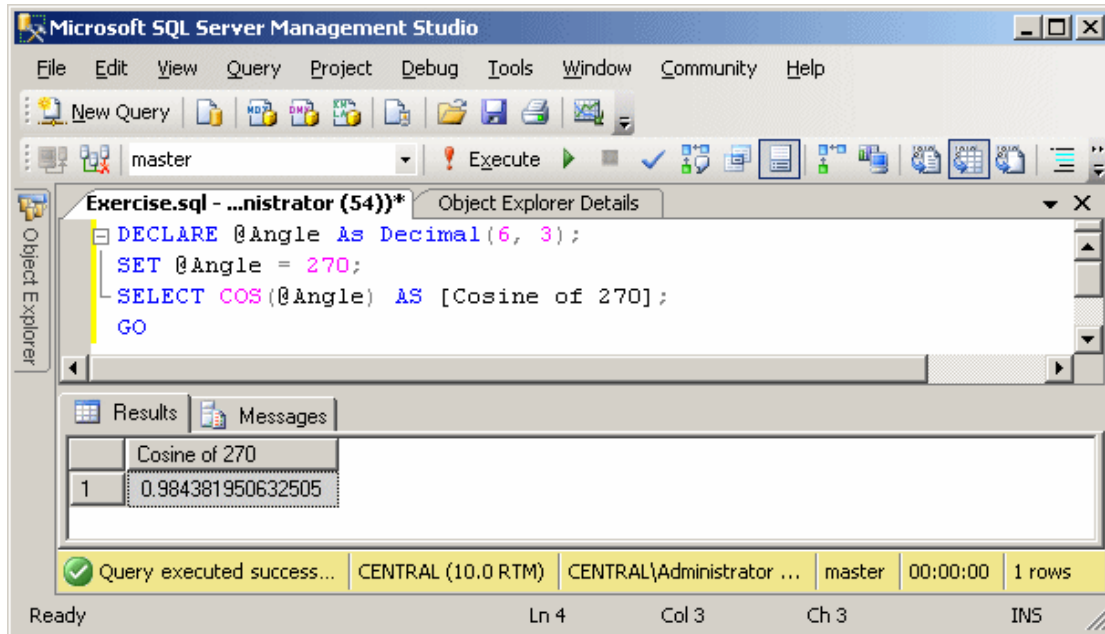
للحصول على جيب التمام لزاوية، يمكنك استدعاء الدالة COS()، صيغتها:

COS(Expression)

تستقبل الدالة COS قيمة الزاوية بالدرجة، وتعيد قيمة جيب التمام لها، مثال:

```
DECLARE @Angle As Decimal(6, 3);
SET @Angle = 270;
SELECT COS(@Angle) AS [Cosine of 270];
GO
```

نتائج الاستعلام:



جيب الزاوية (Sine):

يمثل جيب الزاوية نسبة طول الضلع المقابل إلى طول الضلع المجاور من مثلث قائم الزاوية، وانطلاقاً من المثال السابق يمثل جيب الزاوية النسبة CB/AB ، ويحسب باستعمال الدالة $SIN()$ ، وفق الصيغة:

$SIN(Expression)$

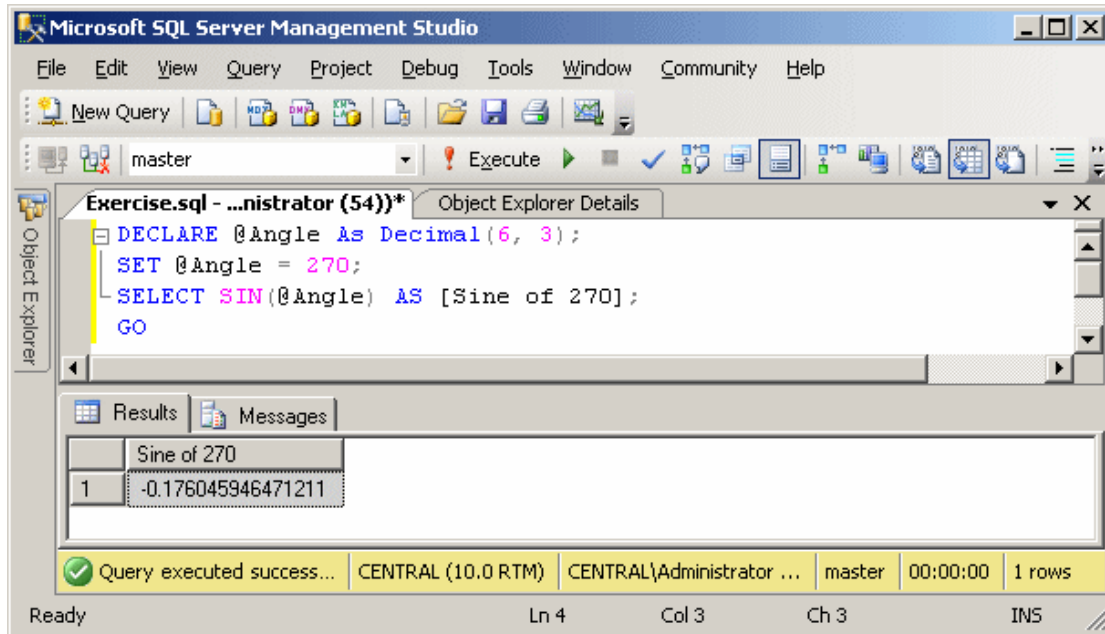
تستخدم الدالة SIN معلم واحد يمثل قيمة الزاوية، لتعيد قيمة جيبها بين 1 و-1، مثال:

```

DECLARE @Angle As Decimal(6, 3);
SET @Angle = 270;
SELECT SIN(@Angle) AS [Sine of 270];
GO

```

النتائج:



ظل الزاوية (Tangent):

تمثل قيمة ظل زاوية نسبة طول الضلع المقابل على طول الضلع المجاور، وتمثل انطلقاً من المثال السابق النسبة: BC/AC ، لحساب ظل زاوية تستخدم الدالة $TAN()$ ، وفق الصيغة:

$TAN(Expression)$

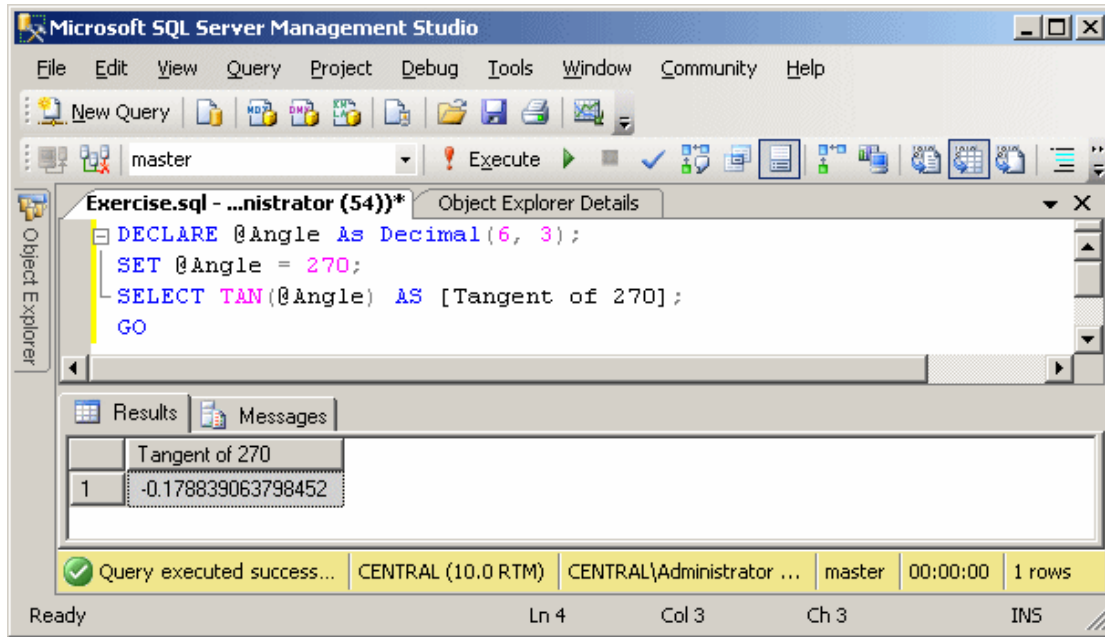
تأخذ هذه الدالة قيمة وحيدة تمثل الزاوية وتعيد قيمة ظلها، مثال:

```

DECLARE @Angle As Decimal(6, 3);
SET @Angle = 270;
SELECT TAN(@Angle) AS [Tangent of 270];
GO

```

والنتيجة:



دوال التاريخ والتوقيت:

التاريخ والتوقيت من أهم أنواع البيانات استعمالاً في تطبيقات قواعد البيانات، لأنها تتعلق بالمبيعات، وجدول الأعمال، والضرائب، والعمل الإضافي، وغير ذلك من العمليات التي تستخدمها عند تطوير برنامجك. يوفر Transact-SQL مستوى عالٍ من الدعم لقيم التاريخ والتوقيت والدوال المتعلقة بهما.

قبل أن تستخدم التاريخ أو التوقيت في الحساب عليك التأكد أولاً من استخدام النوع المناسب DateTime أو SmallDateTime، ولا تنسى تهيئة المتغيرات بالقيم المناسبة قبل استخدامها، حيث يمكنك الحصول على التاريخ أو الوقت من تطبيق آخر أو من إدخال قام به المستخدم.

التاريخ والوقت الحالي للنظام:

من الطرق التي يمكن أن تساعد بها المستخدم في إدخال التاريخ والوقت الحالي، الدالة GETDATE()، وصيغتها كما يلي:

GETDATE ()

تعيد هذه الدالة التاريخ والوقت الحالي من نظام التشغيل.

إضافة التاريخ / الوقت:

من أهم العمليات التي قد ترغب في القيام بها على قيمة تاريخية هي إضافة قيمة تاريخية لها، للقيام بذلك استخدم الدالة DATEADD، وفق الصيغة:

```
DATEADD(.TypeOfValue, ValueToAdd, DateOrTimeReferenced)
```

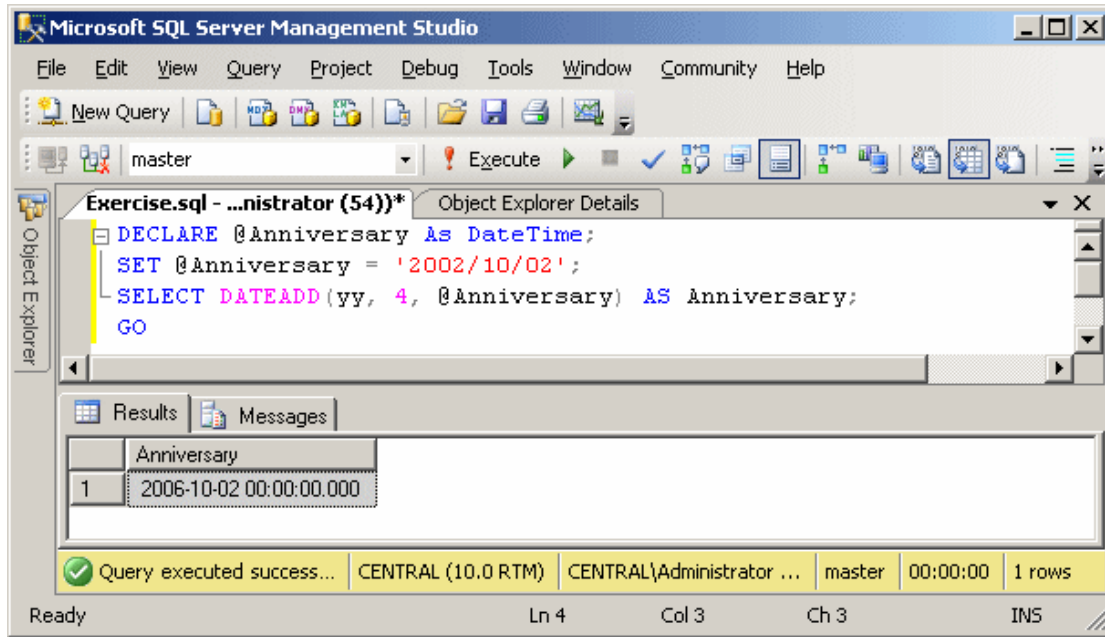
المعلم الثالث *DateOrTimeReferenced* لهذه الدالة هي قيمة التاريخ أو الوقت الذي تأخذه الدالة، ويمكن أن يكون قيمة ثابتة بالشكل 'السنة/الشهر/اليوم' في متغير التاريخ، أو 'ساعة: دقيقة ص/م' لمتغير الوقت.

المعلم الثاني *ValueToAdd* الذي تتخذه الدالة يمثل القيمة التي تضاف إلى التاريخ أو الوقت، وينبغي أن يكون عددا ثابتا صحيحا كان أم عشريا، في المعلم الثالث *TypeOfValue* تحدد طبيعة القيمة المضافة.

لاستخدام الدالة يجب أولا تحديد نوع القيمة التي تريد إضافتها، وهي تستخدم على النحو التالي:

- إذا كنت ترغب في إضافة عدد من السنين على التاريخ، حدد قيمة *TypeOfValue* بالسنوات مثل yy أو yyyy أو أكتب **Year**، مثال:

```
DECLARE @Anniversary As DateTime;  
SET @Anniversary = '2002/10/02';  
SELECT DATEADD(yy, 4, @Anniversary) AS Anniversary;  
GO
```

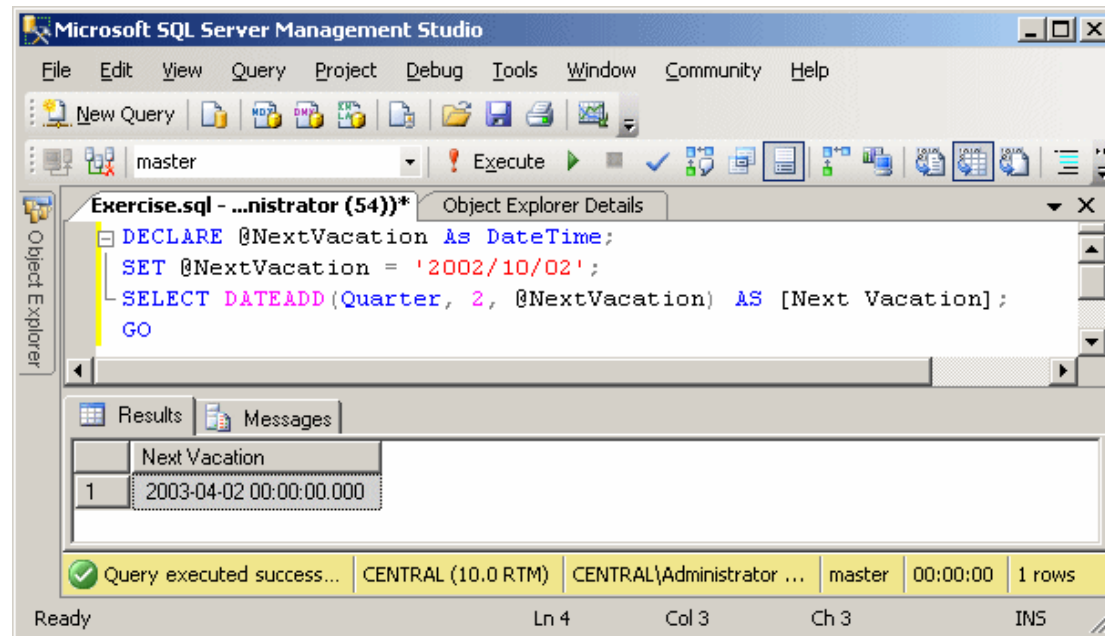


- إذا أردت إضافة عدد من فصول السنة (ربع السنة) لقيمة التاريخ، حدد قيمة TypeOfValue بالرمز **q** أو **qq** أو أكتب **Quarter**، مثال:

```

DECLARE @NextVacation As DateTime;
SET @NextVacation = '2002/10/02';
SELECT DATEADD(Quarter, 2, @NextVacation) AS [Next Vacation];
GO

```



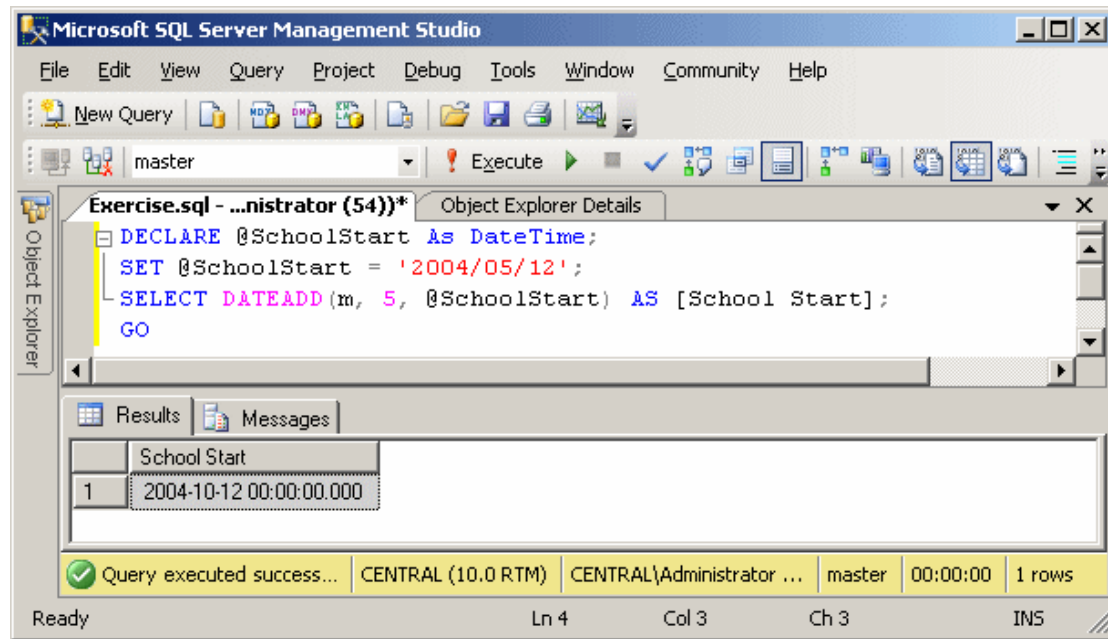
- إذا أردت إضافة عدد من الأشهر إلى التاريخ، حدد قيمة TypeOfValue بالرمز **m** أو **mm** أو أكتب **Month**، هذا المثال يضيف 5 أشهر إلى التاريخ:

```

DECLARE @SchoolStart As DateTime;
SET @SchoolStart = '2004/05/12';

```

```
SELECT DATEADD(m, 5, @SchoolStart) AS [School Start];
GO
```



بنفس الطريقة، يمكنك إضافة قيم أخرى على النحو التالي:

النتيجة	الاختصار	نوع القيمة
عدد السنوات التي ستضاف إلى قيمة التاريخ	yy	Year
	yyyy	
عدد أرباع السنة التي ستضاف إلى قيمة التاريخ	q	quarter
	qq	
عدد الأشهر التي ستضاف إلى قيمة التاريخ	m	Month
	mm	
عدد الأيام من السنة ستضاف إلى قيمة التاريخ	y	dayofyear
	dy	
عدد الأيام التي ستضاف إلى قيمة التاريخ	d	Day
	dd	
عدد الأسابيع التي ستضاف إلى قيمة التاريخ	wk	Week
	ww	
عدد الساعات التي ستضاف إلى قيمة الوقت	hh	Hour
عدد الدقائق التي ستضاف إلى قيمة الوقت	n	minute
	mi	
عدد الثواني التي ستضاف إلى قيمة الوقت	s	second
	ss	
عدد من أجزاء ميلي-الثانية التي ستضاف إلى قيمة الوقت	ms	millisecond

الفرق بين قيمتين من التاريخ / التوقيت:

من العمليات الهامة في تصميم قواعد البيانات حساب الفرق بين قيمتين من التاريخ أو التوقيت، لدعم هذه العملية يوفر Transact-SQL الدالة DATEDIFF، وصيغتها:

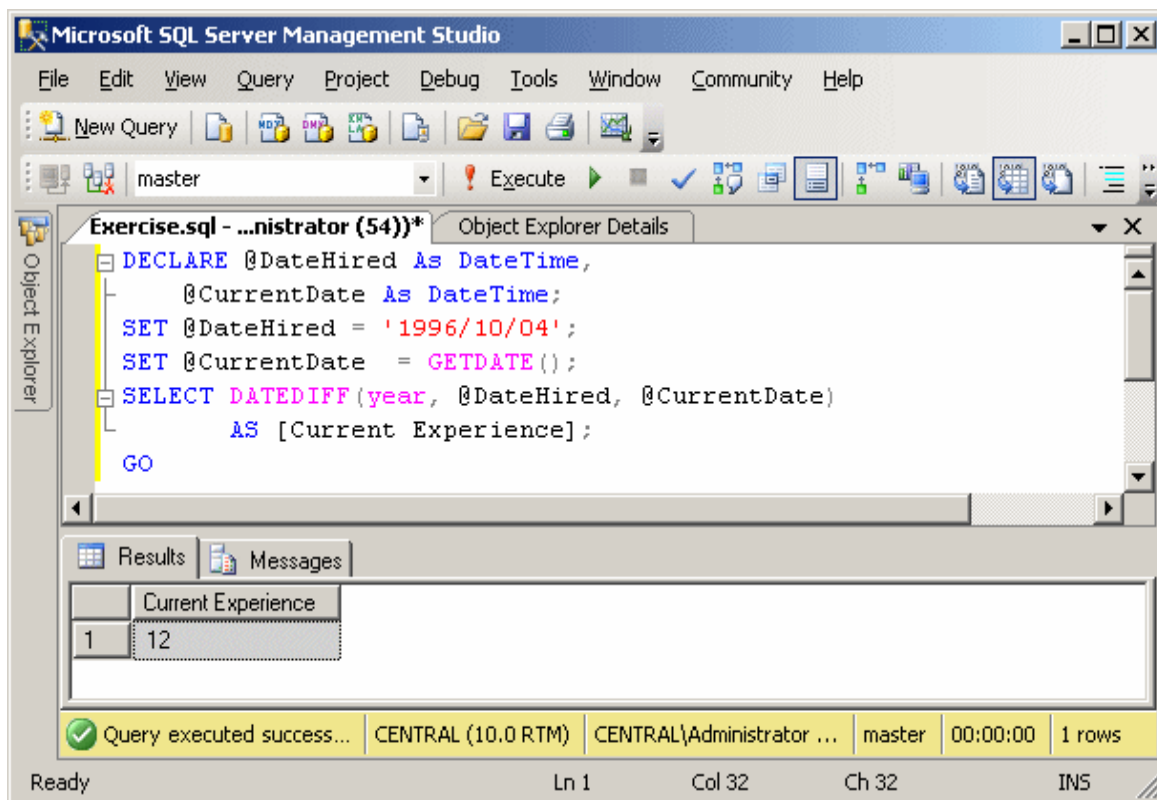
`DATEDIFF(.TypeOfValue, StartDate, EndDate)`

تتضمن هذه الدالة على ثلاث معالم، المعلم الثاني يمثل القيمة الأولى من التاريخ أو التوقيت والمعلم الثالث القيمة الثانية للتاريخ أو الوقت، ويستخدم المعلم الأول لتحديد نوع القيمة التي تريد من الدالة إرجاعها، يأخذ هذا المعلم القيم نفسها التي تستخدم في الدالة .DATEADD

هذا المثال يحسب عدد السنوات التي قضاها عامل في الشركة:

```
DECLARE @DateHired As DateTime,
        @CurrentDate As DateTime;
SET @DateHired = '1996/10/04';
SET @CurrentDate = GETDATE();
SELECT DATEDIFF(year, @DateHired, @CurrentDate)
       AS [Current Experience];
GO
```

وهذا ينتج:



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
DECLARE @DateHired As DateTime,
        @CurrentDate As DateTime;
SET @DateHired = '1996/10/04';
SET @CurrentDate = GETDATE();
SELECT DATEDIFF(year, @DateHired, @CurrentDate)
       AS [Current Experience];
GO
```

The Results pane shows a single row with the value 12 under the column header 'Current Experience'.

Current Experience
12

The status bar at the bottom indicates: Query executed success... CENTRAL (10.0 RTM) CENTRAL\Administrator ... master 00:00:00 1 rows

تطبيق عملي: نهاية الدرس

- 1- أغلق نافذة محرر الاستعلامات دون حفظ الملف.
- 2- من متصفح الكائنات اختر Databases، انقر باليمين على Exercise1 واختر Delete.
- 3- في مربع الحوار، انقر على OK.

ملخص الدرس:

تمارين متنوعة:

- 1- أكتب الاستعلام الذي ينشئ دالة تحسب عدد ساعات العمل في الأسبوع، ثم أكتب دالة تعيد قيمة الساعات الإضافية بحيث إذا كان الموظف يعمل أقل من 40 ساعة فليس هناك وقت إضافي، وإذا كان الموظف يعمل أكثر من 40 ساعة، فالعدد الزائد عن 40 ساعة يعتبر ساعات العمل الإضافية.
- 2- أكتب الدالة التي تعرض انطلاقا من قيمة تاريخ اسم هذا الأسبوع من هذا التاريخ.
- 3- أكتب الدالة التي تحسب الفرق بين تاريخين وتعيد النتيجة بعدد من الأيام.
- 4- أكتب الدالة التي تضيف عدد صحيحا إلى قيمة تاريخ.

الجدول في قاعدة البيانات

الجدول:

مبدئيا الجدول هو قائمة من البنود أو مجموعة من القوائم، لإدارة محتويات الجدول ينبغي أن تكون منظمة بدقة، ولتنظيم هذه المعلومات يتم تقسيمها إلى أبواب، مثال:

Name	Age	Gender	Relationship
Judie	18	Female	Sister
Ernest	24	Male	Cousin
Bill	52	Unknown	Uncle
David	36	Male	Brother
Hermine	12	Unknown	Niece

محتويات هذا الجدول هي لائحة مرتبة من المعلومات المخزنة فيه، وتدعى هذه المعلومات بالبيانات.

تطبيق عملي: إنشاء قاعدة البيانات

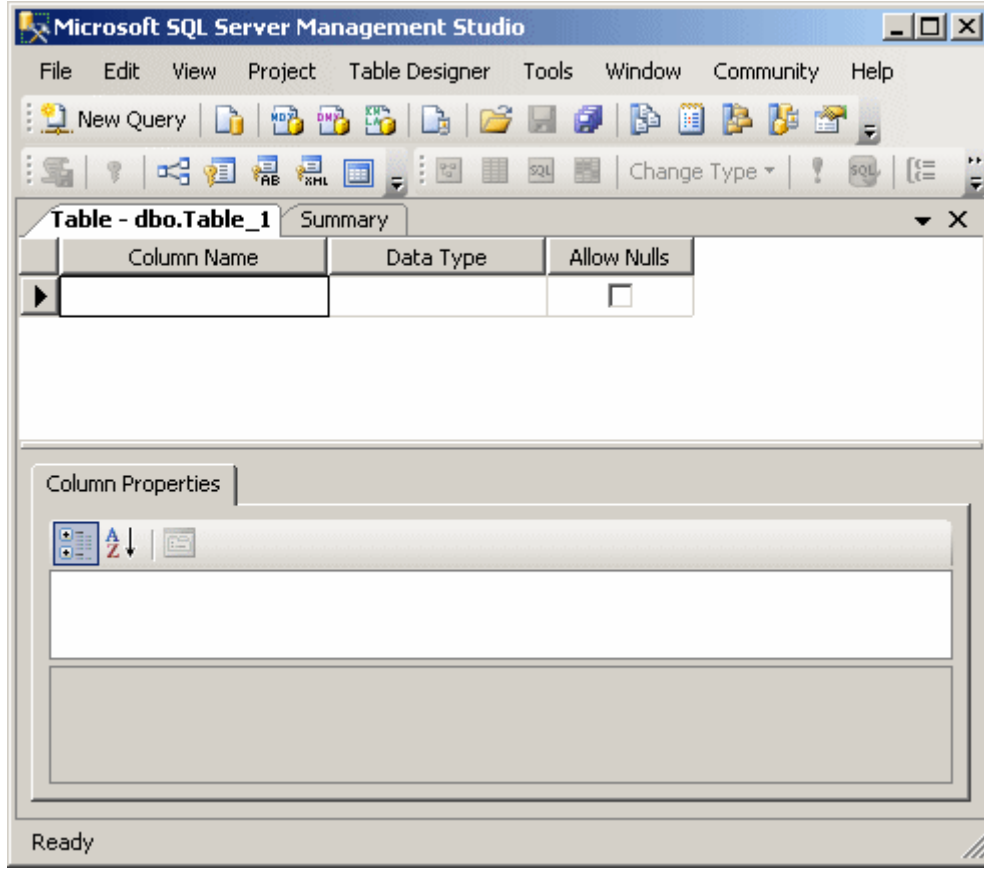
- 1- قم بتشغيل Microsoft SQL Server Management، ثم اتصل بالخادم.
- 2- من متصفح الكائنات، انقر بالزر الأيمن على Databases ثم على New Database...
- 3- في مربع الحوار لإنشاء قاعدة بيانات جديدة، حدد اسم قاعدة البيانات BCR.
- 4- من العمود (Initial Size (MB)، حجم الأولي PRIMARY filegroup بقيمة 10، ثم انقر OK.

إنشاء الجدول أثناء التصميم:

تسمى بيانات الجدول المرتبة في فئات بالأعمدة أو الحقول، وتسمى الترتيبات الأفقية بالصفوف أو السجلات، يحمل كل عمود فئة من البيانات مشتركة بين جميع السجلات، ويجب أن يكون في الجدول على الأقل عمود واحد، أي أنه لا يمكن إنشاء جدول دون إنشاء عمود واحد على الأقل.

تطبيق عملي: إنشاء جدول

- 1- من متصفح الكائنات، قم بتوسيع البند الخاص بقاعدة البيانات BCR (انقر على +).
- 2- تحت BCR، انقر بالزر الأيمن واختر Tables وانقر على New Table...



- 3- وحيث يظهر المؤشر في الخانة الفارغة في تحت اسم العمود الفارغ، اكتب name واضغط Enter.

أسماء الجداول:

لاستكمال إنشاء الجدول ينبغي حفظه، ويمكن أن يكون اسم الجدول مكون من أرقام فقط، مثل 148، مبتدئاً برقم، حرف، أو الرمز _، ويمكن إضافة الحروف والفراغات.

لتجنب الخلط هنا سنتبع لتسمية الجداول ما يلي:

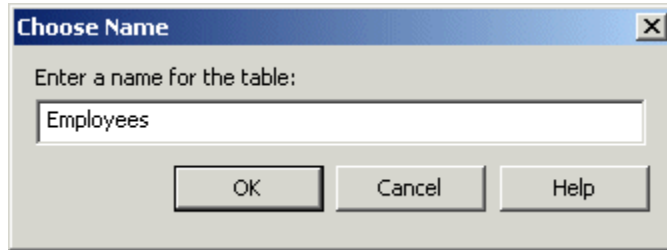
بداية اسم الجدول بحرف ثم يليه مجموعة من الحروف والأرقام والرمز _، أمثلة: _n24، act_52_t، مع اجتناب الرموز الخاصة مثل !، @، #، \$، %، ^، &، *

ويستحسن إذا كان الاسم مركب من عدة كلمات، أن تبدأ كل كلمة بالأحرف الكبيرة،


أمثلة: Staff Members، Video Titles

تطبيق عملي: تسمية الجدول

1- لحفظ الجدول، من شريط الأدوات الرئيسي انقر على زر الحفظ 



2- في مربع الحوار لاختيار الاسم، أدخل Employees

3- بعد حفظ الجدول قم بالنقر على زر الإغلاق 

إنشاء الجدول باستخدام SQL:

لإنشاء الجدول في محرر SQL، مبدئياً يستخدم الاستعلام الآتي:

```
CREATE TABLE TableName;
```

لإنشاء الجدول تستخدم العبارة CREATE TABLE، حيث يمثل العامل TableName اسم الجدول الجديد، ويجب عند إنشاء الجدول تحديد حقل واحد من البيانات على الأقل.

باستخدام عينة من الاستعلام:

لمساعدتك في إنشاء الجدول يمكن باستخدام SQL Server Management Studio تخريج عينة SQL، ومن ذلك يمكنك تعديله أو تخصيصه بما يناسبك، للقيام بذلك افتح نافذة استعلام، ثم لعرض قوالب SQL، من القائمة الرئيسية، انقر على View <-- Templates Explorer، قم بتوسيع البند Table، ثم اسحب وأفلت البند Create Table في نافذة استعلام جديدة، ليتم توليد شيفرة SQL.

صيانة الجدول:

تتمثل صيانة الجداول في إعادة النظر في هيكلها لتغيير تركيبتها، كتغيير الاسم أو حذف أو إضافة عناصر أخرى...

خصائص الجدول:

كغيرها من الأغراض المنشأة، يملك الجدول بعض الخصائص التي تتناسبه، للنظر إلى هذه الخصائص انقر بالزر الأيمن على الجدول ثم انقر على Properties.

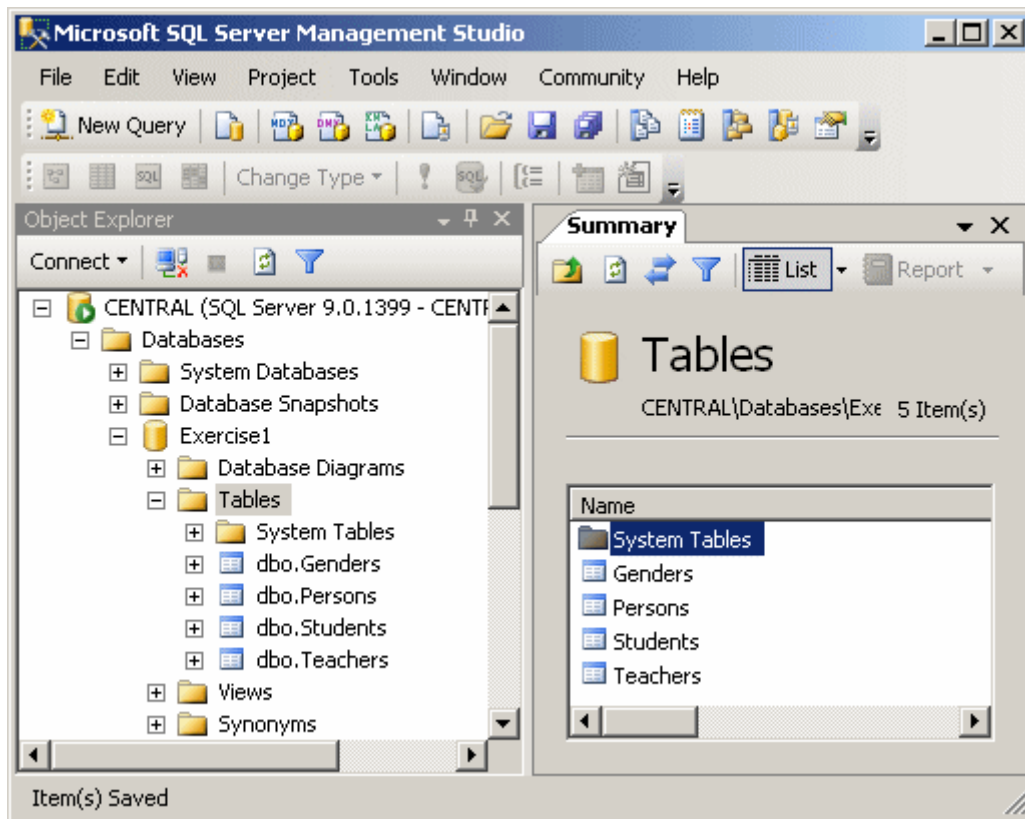
فتح الجدول:

لفتح أي جدول والقيام بتعديلات عليه، يوفر SQL Server Management Studio عدة طرق:

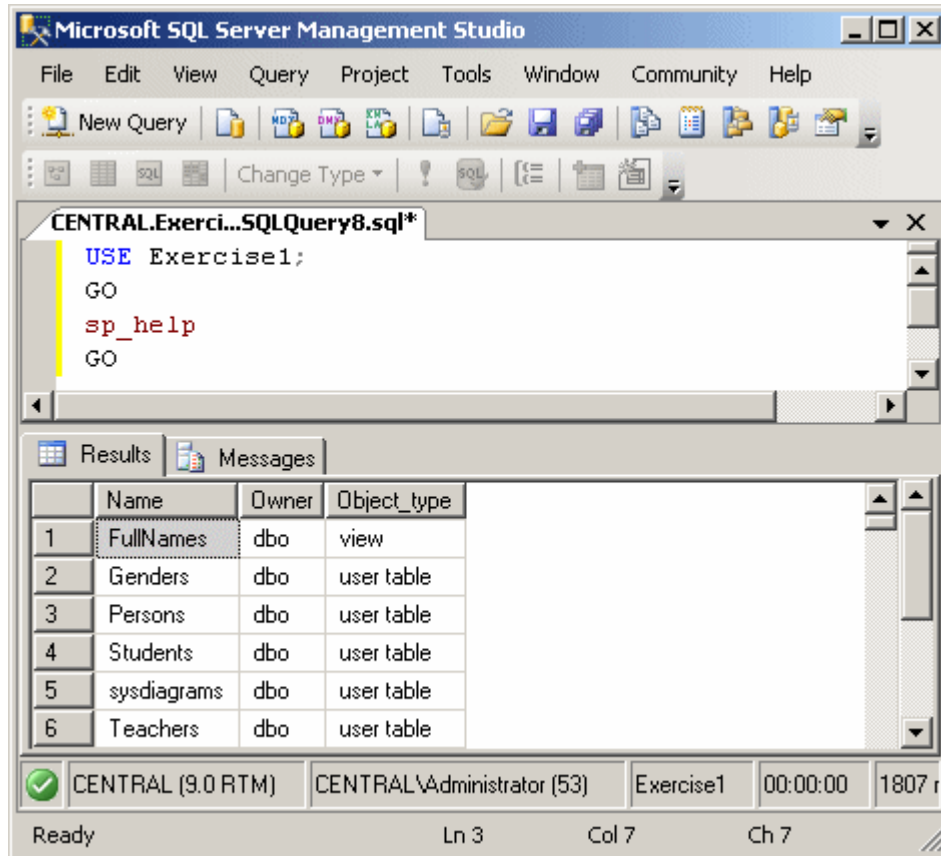
- للإطلاع على هيكله الجدول، لتغيير أعمدته مثلاً، قم بتوسيع قاعدة بياناتك، ثم انقر بالزر الأيمن على الجدول الذي ترغب بالتعديل عليه واختر Modify، وسيفتح الجدول في وضع التصميم لترى نفس الواجهة التي استخدمتها لإنشاء الجدول.
- إذا أردت استعراض استعلام إنشاء الجدول، من متصفح الكائنات انقر بالزر الأيمن على الجدول ثم اختر Script Table AS، ثم CREATE To، وانقر على New Query Editor Window
- لفتح الجدول وعرض محتوياته، أو إدخال بيانات إليه، انقر عليه بالزر الأيمن ثم اختر Open Table

مراجعة الجداول:

لمشاهدة جداول قاعدة بيانات في متصفح الكائنات، يمكنك النقر على البند Tables:



لمشاهدة قائمة جداول قاعدة بيانات باستخدام SQL، في نافذة محرر الاستعلام حدد قاعدة البيانات (باستخدام العبارة USE)، ونفذ `sp_help` (وهي من الإجراءات المخزنة)، مثال:



تغيير اسم الجدول:

إذا رأيت أن اسم الجدول غير مناسب يمكنك تغييره، لتغيير اسم جدول في SQL Server من متصفح الكائنات انقر بالزر الأيمن على الجدول وانقر على Rename، أدخل الاسم الجديد واضغط Enter.

لتغيير اسم الجدول باستخدام SQL قم بتنفيذ الإجراء المخزن **sp_rename**، يليه الاسم الحالي للجدول، بعده فاصلة ثم الاسم الجديد للجدول، بالصيغة التالية:

```
sp_rename ExistingTableName, TableNewName;
```

وينبغي إدراج أسماء الجداول بين علامتي تنصيص، مثال:

```
sp_rename 'StaffMembers', 'Employees';
GO
```

في هذه الحالة، سيقوم المترجم بالبحث عن اسم الجدول StaffMembers في قاعدة البيانات الحالية¹، فإن وجدها سيغير اسمها لتصبح Employees، وإذا كان الجدول غير موجود فإنك ستحصل على خطأ.

حذف الجدول:

إذا كان لديك جدول غير مرغوب فيه ضمن قاعدة البيانات، يمكنك حذفه. لحذف جدول باستخدام SQL Server Management Studio، انقر بالزر الأيمن في متصفح الكائنات على اسم الجدول من قاعدة البيانات واختر من القائمة Delete، ستتلقى تحذيراً يعطيك فرصة للتراجع عن الحذف، أكد اختيارك بالنقر على OK.

لحذف جدول باستخدام SQL ، استخدم الصيغة التالية:

```
DROP TABLE TableName
```

العبارة DROP TABLE عبارة محجوزة تستخدم لحذف الجدول، ويلبها مباشرة اسم الجدول غير المرغوب فيه، وعند تنفيذ الاستعلام لن تتلقى أي تحذير قبل حذف الجدول.

يمكنك أيضاً استخدام قالب SQL الذي يوفره SQL Server Management Studio، للقيام بذلك أولاً قم بعرض نافذة استعلام جديدة، افتح Templates Explorer واختر من البند Table ثم حدد منه Drop Table، قم بسحب وإفلات هذا البند إلى نافذ الاستعلام، وسيتم توليد SQL الخاص بحذف الجدول، عدل على المتغيرات ونفذ الاستعلام.

تحديد الجدول:

سيأتي في هذه الدروس عبارات كثيرة تحتوي على أسماء الجداول، في هذه التعبيرات سنحتاج لتحديد الجدول الذي نريد استخدامه، وهناك ثلاثة طرق رئيسية يمكنك القيام بتأشير أو تحديد جدول معين:

- ببساطة كتابة اسمه مباشرة، مثل Students

¹ للتعرف على قاعدة البيانات الحالية، انظر إلى شريط الأدوات الرئيسي ستشاهد قائمة اختيارات (ComboBox) تحمل أسماء جميع قواعد البيانات المعرفة، والاسم المحدد هو الذي يتم تطبيق الاستعلامات عليه ما لم يتم تحديد قاعدة البيانات التي يتم التعامل معها باستخدام العبارة USE.

- يمكنك كتابة ¹dbo متبوعة بنقطة ثم اسم الجدول، مثلا: dbo.Students
- يمكنك كتابة اسم قاعدة البيانات التي ينتمي إليها الجدول، تليها نقطة، ثم dbo ثم نقطة وبعدها اسم الجدول، مثال RedOakHighSchool.dbo.Students

الأعمدة في الجدول:

رأينا أن في البداية أن بيانات الجدول تنظم في قائمة من الفئات تدعى بالأعمدة، لتنظيم البيانات في عمود ما، يحتاج الجدول إلى تفاصيل عن كل عمود، تتمثل أساسا في اسم العمود ونوع البيانات التي يمكن أو يجب أن يحتويها.

اسم العمود:

يمكن تسمية عمود من البيانات باستخدام الحروف اللاتينية أو الأرقام و الرمز _ أو توليفة من كل ذلك، وفيما سيأتي سنتبع نفس طريقة تسمية الجداول.

تطبيق عملي: تحديد أسماء الأعمدة

- 1- قم بإنشاء قاعدة البيانات بالنقر على Databases بالزر الأيمن واختر New Database... اكتب الاسم Employees ثم وافق بالنقر على OK.
- 2- انقر على قاعدة البيانات Employees ثم انقر بالزر الأيمن على Tables واختر New Table... .Table...
- 3- اكتب اسم العمود الأول (الحقل) FirstName واضغط على Enter.

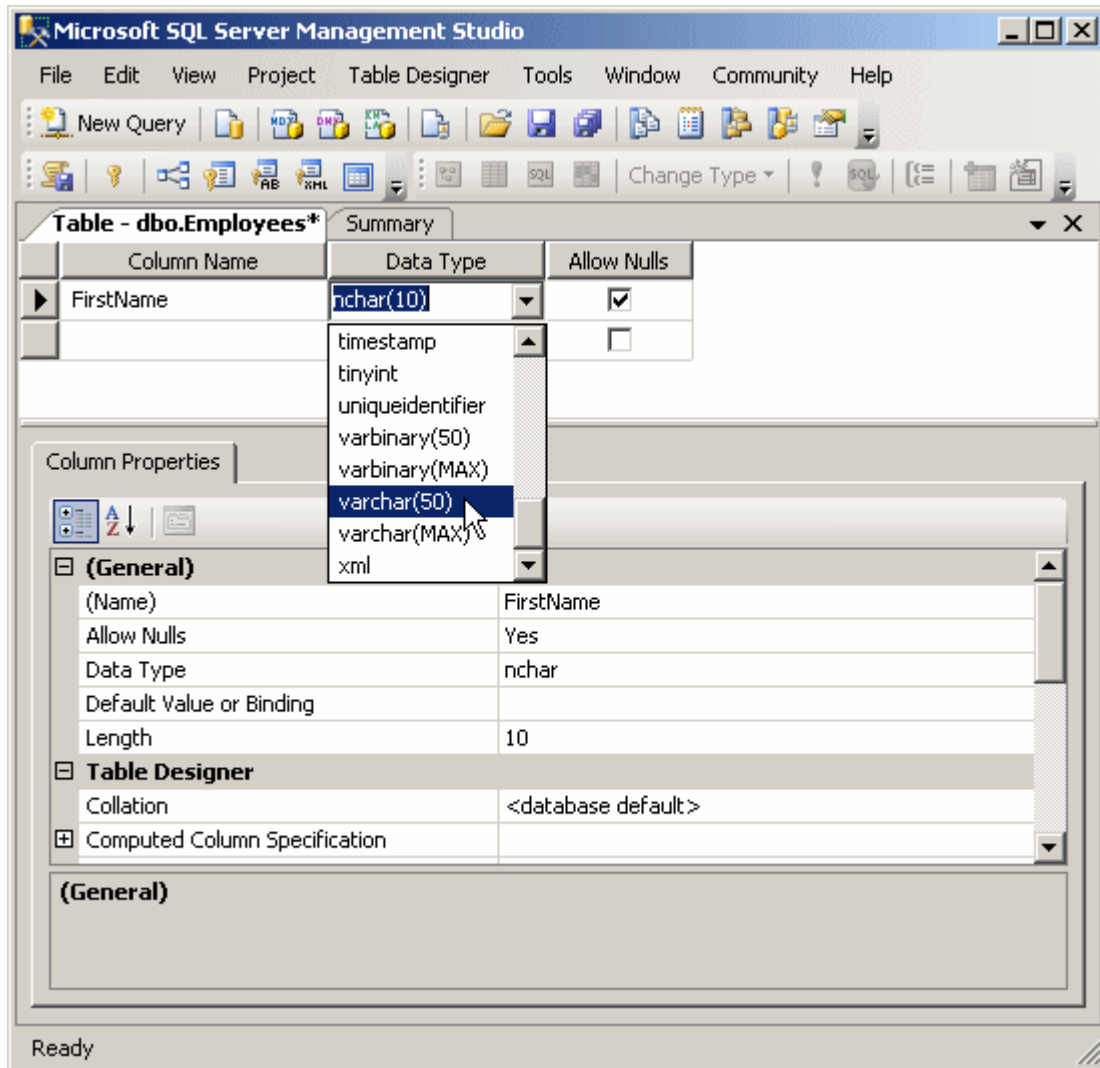
أنواع البيانات:

بعد تحديد اسم الحقل يبقى تحديد نوع من البيانات التي سيجوئها هذا الحقل حسب ما رأيناه في درس أنواع البيانات من هذه السلسلة، وهنا يجب أن تحدد نوع البيانات المناسب لكل حقل.

تطبيق عملي: تحديد أنواع البيانات

¹ dbo هو اسم المستخدم المالك (Owner) لقاعدة البيانات، يقوم Microsoft SQL Server بإنشائه افتراضيا.

- 1- انقر على السهم لاختيار أحد الأنواع المتوفرة في القائمة نوع البيانات.
- 2- انتقل لأسفل القائمة وحدد varchar.



- 3- انقر تحت الحقل FirstName واكتب MI
- 4- اضغط على مفتاح السهم للأسفل LastName ثم اضغط مفتاح السهم للأسفل ثانية.
- 5- اكتب DateHired واضغط المفتاح Tab واكتب الحرف d.
- 6- لاحظ أن اختيار النوع datetime.
- 7- أكمل الجدول على النحو التالي:

Column Name	Data Type	Allow Nulls
FirstName	varchar(50)	<input checked="" type="checkbox"/>
MI	char(10)	<input checked="" type="checkbox"/>
LastName	varchar(50)	<input checked="" type="checkbox"/>
DateHired	datetime	<input checked="" type="checkbox"/>
EmployeeNumber	char(10)	<input checked="" type="checkbox"/>
Address	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	char(10)	<input checked="" type="checkbox"/>
ZIPCode	varchar(50)	<input checked="" type="checkbox"/>
Country	varchar(50)	<input checked="" type="checkbox"/>
WorkPhone	varchar(50)	<input checked="" type="checkbox"/>
Extension	char(10)	<input checked="" type="checkbox"/>
MaritalStatus	bit	<input checked="" type="checkbox"/>
HourlySalary	money	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

8- قم بحفظ الجدول.

طول البيانات:

تتعامل قاعدة البيانات مع مختلف أنواع البيانات، بعضها مناسب والآخر غير ملائم لبعض المجالات، وهذا يعني أن عليك مراعاة تلائم البيانات بقدر ما تستطيع وذلك عن طريق التحكم في كمية المعلومات التي يمكن تخزينها في حقل معين. وحيث يمكن لنفس النوع من البيانات الخاصة أن يقوم برقابة داخلية آلية أثناء عملية إدخال البيانات، من أهم أنواع هذه الرقابة حجم أو طول البيانات، حيث يمكن للأعمدة التي تحمل نفس نوع البيانات أن تكون لها أطوال مختلفة.

الحقول المنطقية: لقد رأينا أن الحقول التي تحمل قيمة منطقية (Bit)، مثل: نعم أو لا، True أو False، On أو Off، 1 أو 0 ... إلى غير ذلك، لذلك فطول هذا الحقل يساوي القيمة 1.

الأعداد الصحيحة (Integers): طول عدد Integer هو حجم المساحة التي يحجزها من الذاكرة، وفي النوع int ستكون 4 bytes.

الأعداد العشرية: في الأعداد العشرية والتي تحمل قيم عائمة (Floating-Point) يحدد الطول بالحجم الذي يحجزه الحقل لتخزين القيم.

السلاسل النصية: طول سلسلة نصية لحقل يحدد بالحد الأقصى لعدد الحروف التي يمكن أن يخزنها.

سوف تحتاج أحيانا إلى تغيير حجم نوع بيانات في حقل معين، فمثلا على الرغم من استخدام نفس النوع varchar للاسم والعنوان في حقل ما، فلن يكون بالضرورة لهما نفس الطول، وعموما يمكن قبول اقتراح SQL Server لحجم حقل في قاعدة البيانات.

هناك طريقتان لتغيير طول سلسلة نصية لحقل ما:

- أكتب بين قوسين قيمة طول حجم البيانات، مثلا: VARCHAR(100).
- أو اكتب الطول المناسب في القسم الأسفل من واجهة تصميم الجدول من Length واضغط Enter.

تطبيق عملي: تحديد أنواع البيانات

- 1- في القسم العلوي انقر على EmployeeNumber لتحديدها.
- 2- في القسم السفلي، انقر على Length واكتب القيمة 6.
- 3- ثم حدد الحقل Address واضغط المفتاح Tab.
- 4- لنوع البيانات اكتب VARCHAR (100).
- 5- بنفس الطريقة، أكمل الجدول على النحو التالي:

Column Name	Data Type	Allow Nulls
FirstName	varchar(20)	<input checked="" type="checkbox"/>
MI	char(1)	<input checked="" type="checkbox"/>
LastName	varchar(20)	<input checked="" type="checkbox"/>
DateHired	datetime	<input checked="" type="checkbox"/>
EmployeeNumber	char(6)	<input checked="" type="checkbox"/>
Address	varchar(100)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	char(2)	<input checked="" type="checkbox"/>
ZIPCode	varchar(10)	<input checked="" type="checkbox"/>
Country	varchar(50)	<input checked="" type="checkbox"/>
WorkPhone	varchar(16)	<input checked="" type="checkbox"/>
Extension	char(3)	<input checked="" type="checkbox"/>
MaritalStatus	bit	<input checked="" type="checkbox"/>
HourlySalary	money	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

6- قم بحفظ الجدول.

إنشاء الحقول برنامجياً:

رأينا أن الصيغة العامة لإنشاء الجدول بهذا الشكل:

```
CREATE TABLE TableName
```

بعد تحديد اسم الجدول يتم تحديد قائمة الحقول التي يحتويها، تبدأ القائمة بفتح قوس ")" وتنتهي بقوس مغلق "(", ويفصل بين أسماء الحقول بفاصلة على النحو التالي:

```
CREATE TABLE Country(Column1, Column2, Column3)
```

يفضل كتابة كل حقل في سطر منفرد لتسهيل القراءة.

عند إنشاء أي حقل يجب تحديد اسمه ونوعه، على النحو التالي:

```
ColumnName DataType Options
```

بعد كتابة اسم الحقل حدد نوع البيانات المناسب له، وحدد طول البيانات المدخلة إذا استلزم نوع البيانات ذلك، وهذا ينطبق على جميع الأنواع النصية (char، text، varchar،

الخ). في حالة إنشاء نوع بيانات نصي باستخدام SQL لإنشاء الحقول (برمجيا)، لن تشاهد قيمة افتراضية لطول البيانات، ولا يمكنك الاعتماد على القيمة الافتراضية التي يقترحها SQL، ولذلك عند استخدام SQL لإنشاء الحقول عليك أن تحدد بنفسك طول البيانات النصية.

يمكنك أيضا استخدام قوالب SQL الجاهزة التي يوفرها SQL Server Management Studio لإنشاء الجدول، هذا يسمح لك بالحصول على مزيد من السيطرة على SQL، للقيام بذلك افتح نافذة استعلام جديدة وقم بفتح البند Table من Templates Explorer ثم اسحب وافلت أحد هذه العناصر Create Table, Add Column, Drop Column.

قم بالتعديل على SQL واحذف الأجزاء غير المرغوب فيها، هنا مثال:

```
--
=====
=====
-- Add column template
--
-- This template creates a table, then it adds a new column to the
table.
--
=====
=====
USE <database, sysname, AdventureWorks>
GO

CREATE TABLE <schema_name, sysname, dbo>.<table_name, sysname,
sample_table>
(
    column1 int,
    column2 char(10)
)
GO
```

تطبيق عملي: إنشاء الجدول

1- من متصفح الكائنات، حدد قاعدة البيانات BCR وافتح نافذة استعلام جديدة.

2- أكتب في محرر SQL، اكتب ما يلي:

```
CREATE TABLE Customers (
DrvLicNbr VarChar(50),
DateIssued DateTime,
DateExpired DateTime,
FullName varchar(120),
Address VARCHAR(120),
City varchar(50),
State varchar(100),
PostalCode varchar(20),
HomePhone varchar(20),
OrganDonor bit)
```

3- اضغط F5 لتنفيذ الاستعلام.

4- إغلق نافذة الاستعلام واحفظه باسم Customers.

تحديد الحقول:

سنكتب الكثير من الاستعلامات التي تشمل أسماء الحقول، في مثل هذه الاستعلامات يتم التعرف على الحقل من خلال اسمه، وهناك عدة طرق للقيام بذلك:

- كتابة اسم الجدول الذي ينتمي إليه الحقل متبوعاً بنقطة ثم اسم الحقل، مثال:

Employee.LastName.

- كتابة dbo متبوعاً بنقطة يليها اسم الجدول الذي ينتمي الحقل ثم نقطة، ثم اسم الحقل،

مثال: dbo.Employee.LastName.

- كتابة اسم قاعدة البيانات التي تحوي الجدول متبوعاً بنقطة، ثم dbo ثم نقطة يليها اسم

الجدول الذي ينتمي إليه الحقل متبوعاً بنقطة، ثم اسم الحقل، مثال:

RedOakHighSchool.dbo.Employee.LastName.

استخدام اسم مستعار للجدول:

يمكنك إنشاء اسم مستعار للجدول، لاستخدامه في كتابة الاستعلام وتحديد الحقول، للقيام بذلك اكتب الاسم المستعار (حرفاً أو كلمة) الذي سيمثل الجدول، متبوعاً بنقطة ثم اسم أحد حقوله، مثلاً نختار اسم مستعار للجدول Employee ونسميه empl، ثم نكتب لتحديد الحقل LastName ما يلي: empl.LastName.

وفي نهاية الاستعلام، يجب أن نكتب اسم الجدول متبوعاً بالكلمة AS أو فراغ، ثم الاسم

المستعار، مثال: Employee empl أو Employee AS empl، ويكون الاستعلام بالشكل:

```
SELECT empl.LastName FROM Employee empl
```

أو:

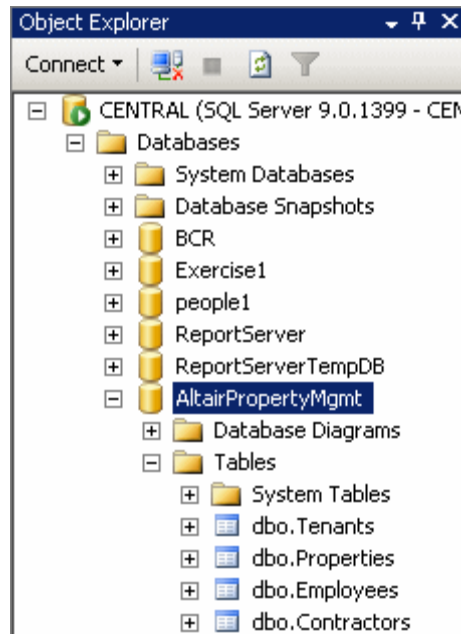
```
SELECT e.LastName FROM Employee AS e
```

صيانة الحقول:

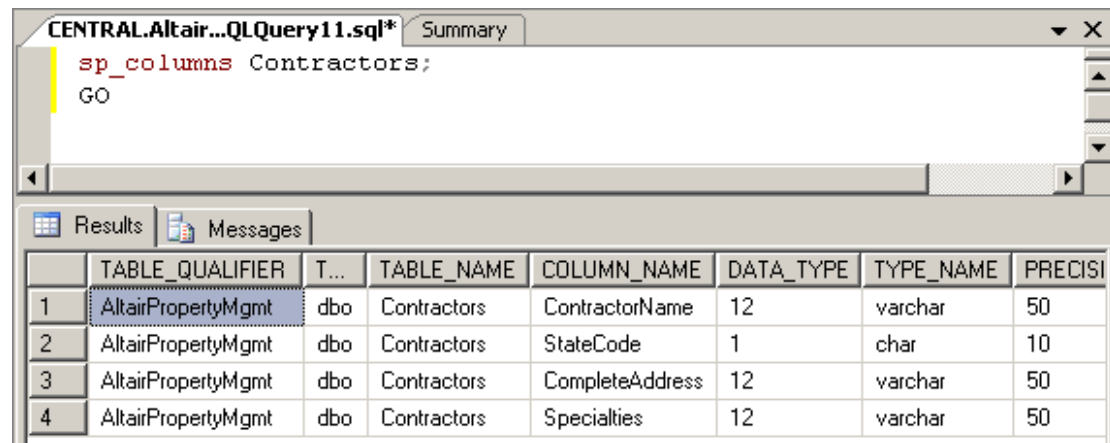
تتمثل صيانة الحقول في مراجعتها من أجل تعديلها أو تغيير تركيبتها، ويشمل ذلك إعادة النظر في هيكل أعمدة الجدول، إعادة تسمية عمود، حذف عمود، وتغيير نوع البيانات، قابلية عدم إدخال قيم ... إلخ.

استعراض الحقول:

لاستعراض هيكل جدول باستخدام SQL Server Management Studio ، قم بتوسيع بند الجدول من متصفح الكائنات:



للإطلاع على حقول أي جدول باستخدام SQL استخدم الإجراء المخزن sp_columns بكتابة اسم الإجراء متبوعاً باسم الجدول، مثال:

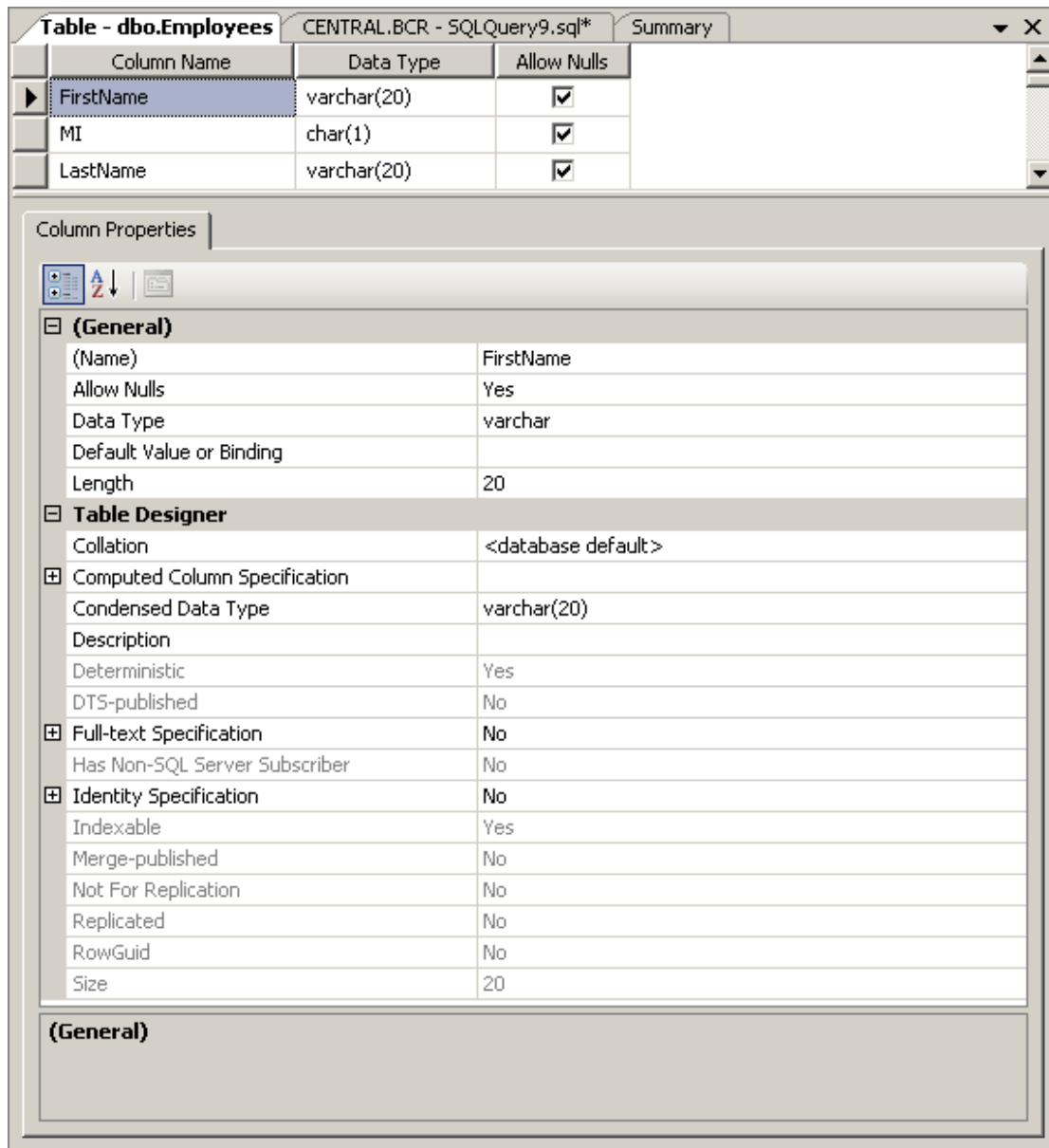


يعرض هذا الاستعلام أسماء حقول الجدول في القائمة COLUMN_NAME وغيرها من الخصائص في القوائم الأخرى.

خصائص الحقول:

رأينا فيما سبق ثلاث خصائص أساسية وإجبارية لكل حقل: اسم ونوع البيانات، وطولها إذا اقتضى النوع، بالإضافة إلى هذه الخصائص، هناك خصائص أخرى يمكن استخدامها لمزيد من السيطرة على سلوك حقل معين.

يمكنك التحكم في خصائص أي حقل من الجدول باستخدام جدول الخصائص Column Properties في الجزء السفلي من واجهة تصميم الجدول. تختلف هذه الخصائص في بعض الأحيان حسب نوع البيانات الحقل، لذلك يجب أولاً تحديد الحقل في الجزء العلوي من واجهة التصميم ثم تعديل خصائصه، يمكنك الضغط على الزر F6 للانتقال مباشرة إلى الجزء السفلي حيث خصائص الحقل، وسنعرض بعض خصائص الحقول:



:Description

وصف الحقل من الخانة Description في التبويب Table Designer عام لجميع أنواع الحقول، يستخدم لوصف الحقول، ويمكنك كتابة أي شيء في هذه الخاصية.

:Collation

بسبب إمكانية استخدام آليات مختلفة لإدخال الحروف، ويمكن لهذا أن يؤثر في خوارزميات الاستعلامات التي تتم على البيانات، يمكنك أن تجعل من قاعدة البيانات تطبق آلية

لغة معينة للحقل عن طريق تغيير قيمة الخاصية Collation. وإلا سيستخدم الجدول قيمة افتراضية.

لتحديد قيمة لخاصية Collation لأي حقل عند إنشائه باستخدام SQL، أكتب **COLLATE** متبوعة بالقيمة المرغوب فيها، مثال:

```
CREATE TABLE Customers(  
    FullName varchar(50) COLLATE SQL_Latin1_General_CP1_CI_AS  
);
```

تعديل الحقول:

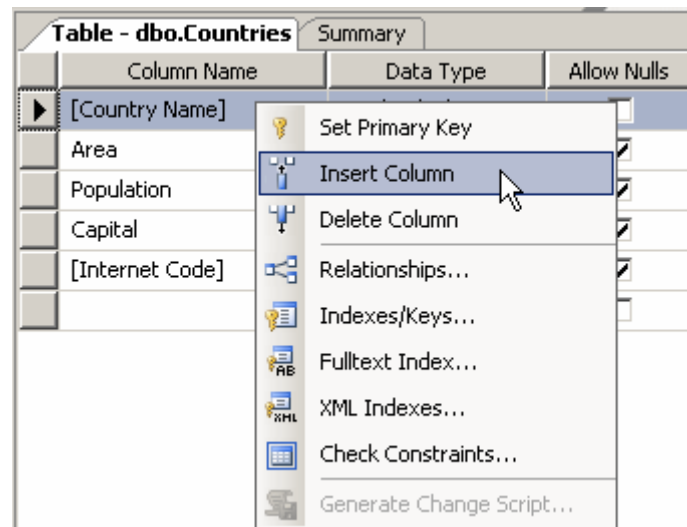
للتعديل على أحد الحقول استخدم عبارة **ALTER TABLE** لإجراء تغيير في الجدول الذي يحتويه، الصيغة العامة لذلك:

```
ALTER TABLE TableName
```

العبارة ALTER TABLE إجبارية و تمثل القيمة TableName اسم الجدول.

إضافة حقل جديد:

يمكن إضافة حقل حتى بعد إنشاء الجدول، لإضافة حقل جديد في SQL Server Management Studio، انقر بالزر الأيمن على الجدول ثم اختر Design Table، انقر بالزر الأيمن واختر Insert Column لإدراج حقل جديد:



باستخدام SQL نستخدم الصيغة التالية لإضافة حقل جديد:

```
ALTER TABLE TableName
```


`ADD ColumnName Properties`

حيث يمثل ColumnName اسم الحقل، وعلى يمينه حدد خصائص الحقل واستخدام لذلك كل الخيارات التي استعرضناها للحقول، مثال:

```
ALTER TABLE StaffMembers
ADD Address varchar(100) NULL
GO
```

بتنفيذ هذا الاستعلام سيتم إضافة حقل جديد إلى الجدول StaffMembers من قاعدة البيانات الحالية، من النوع varchar طوله 100 حرف متغير، مع إمكانية عدم إدخال قيم.

بطريقة أخرى يمكنك استخدام Templates Explorer لإضافة حقل جديد إلى الجدول، اسحب البند Add Column من البند الرئيسي Table وأقلته في نافذة استعلام جديدة أو انقر عليه مرتين، قم بحذف الأجزاء غير المرغوب فيها واترك الجزء الخاص بإضافة حقل، وأخيرا عدل على الاستعلام من خلال الزر المخصص لذلك ثم نفذه، مثال:

```
=====
-- Add column template
--
-- This template creates a table, then it adds a new column to the
table.
=====
USE <database, sysname, AdventureWorks>
GO

-- Add a new column to the table
ALTER TABLE <schema_name, sysname, dbo>.<table_name, sysname,
sample_table>
    ADD <new_column_name, sysname, column3>
        <new_column_datatype,, datetime>
        <new_column_nullability,, NULL>
GO
```

تغيير اسم الحقل:

لتغيير اسم أي حقل، قم بتوسيع بند الجدول من متصفح الكائنات ثم بند Columns وانقر على الحقل الذي ترغب في إعادة تسميته بالزر الأيمن واختر Rename أو اضغط مباشرة على المفتاح F2، بطريقة أخرى يمكنك النقر بالزر الأيمن على الجدول واختر Modify، ثم يمكنك تعديل أي من خصائص الحقول.

باستخدام SQL لتغيير اسم الحقل، استخدم الإجراء المخزن sp_rename وفق الصيغة التالية:

```
sp_rename 'TableName.ColumnName', 'NewColumnName', 'COLUMN'
```

يمثل TableName اسم الجدول الذي ينتمي إليه الحقل، أما ColumnName فهو اسم الحقل، و NewColumnName هو الاسم الجديد للحقل. مثال:

```
sp_rename 'StaffMembers.FullName', 'EmployeeName', 'COLUMN'  
GO
```

عند تنفيذ هذا الاستعلام سيبحث المترجم عن الحقل FullName في جدول StaffMembers من قاعدة البيانات الحالية، فإذا وجدته سيغير اسمه إلى EmployeeName.

حذف حقل:

لحذف أحد الحقول باستخدام SQL Server Management Studio قم بتوسيع بند قاعدة البيانات من متصفح الكائنات ثم الجدول الذي يحوي الحقل المراد حذفه ثم الحقول، انقر بالزر الأيمن على الحقل غير المرغوب فيه وانقر على Delete، ستواجه مربع الحوار الذي يطالبك بتأكيد رغبتك، انقر على OK للحذف.

لحذف حقل باستخدام SQL، أكتب استعلام وفق الصيغة التالية:

```
ALTER TABLE TableName  
DROP COLUMN ColumnName
```

حيث TableName هو اسم الجدول الذي يحوي الحقل المراد حذفه، و ColumnName اسم الحقل الذي تريد حذفه، مثال:

```
ALTER TABLE StaffMembers  
DROP COLUMN CurrentResidence;  
GO
```

يمكن بنفس الطريقة استخدام Templates Explorer لعرض قالب SQL الخاص بحذف حقل، اسحب وأفلت البند Drop Column الموجود في التبويب Table، وقبل تنفيذه قم بحذف الجزء غير المرغوب فيه وتعديل أسماء المتغيرات بما يناسب، مثال:

```
=====
-- Drop column template
--
-- This template creates a table, then it
-- drops one of the columns of the table.
=====
USE <database, sysname, AdventureWorks>
GO

-- Drop a column from the table
ALTER TABLE <schema_name, sysname, dbo>.<table_name, sysname,
sample_table>
    DROP COLUMN <new_column_name, sysname, column3>
```

تطبيق عملي: نهاية الدرس

- أغلق نافذة الاستعلام دون حفظ الملف.
- في متصفح الكائنات، انقر بالزر الأيمن على قاعدة البيانات BCR واختر Delete.

ملخص الدرس:

استعرضنا في هذا الدرس:

- ✓ التعامل مع الجداول
- ✓ التعامل مع الحقول.

تمرين: شركة مرافق عامة

- أنشئ قاعدة بيانات UtilityCompany1.
- قم بإنشاء جدول Employees بالحقول التالية: EmployeeNumber، FirstName، LastName، Title مع تحديد نوع البيانات المناسب.
- قم بإنشاء جدول باسم Customers مزودا بالحقول التالية: AccountNumber، DateAccountCreated، CustomerName، Address، City، State، EmailAddress.
- قم ببعض العمليات التي استعرضناها في هذا الدرس.

السجلات في قاعدة البيانات

ذكرنا سابقا أن غرض الجدول هو تخزين البيانات، ولأن الجدول هو أهم عنصر من قاعدة البيانات فينبغي أن ترتب البيانات فيه بشكل دقيق، وللتحكم أكثر في البيانات المخزنة في الجدول تم تخزينها في شكل أعمدة (نسميها الحقول) وأسطر (نسميها السجلات)، حيث تتقاطع الحقول والسجلات في خانات البيانات، من أجل تحرير وإدارة البيانات في الجداول يمكن استخدام برنامج SQL Server Management Studio أو أي برنامج خارجي مخصص لذلك¹.

لاستعراض الجداول من قاعدة بيانات يعرض SQL Server Management Studio قائمة ذات بنود تمثل قواعد البيانات المثبتة على الخادم، وبنود فرعية تمثل جداول قواعد البيانات.

تصفح البيانات في SQL Server Management Studio:

نقصد بتصفح البيانات عرضها ومراجعتها، ولأن تخزين البيانات يتم في الجداول فإن مراجعة البيانات يتم بتفحص محتويات الجداول. يمكنك التنقل عبر بين مختلف بيانات الجدول من خلال أزرار الأسهم في لوح المفاتيح أو بالنقر مباشرة على الموضع المراد.

إدخال البيانات:

رأينا أن الأعمدة (الحقول) تخزن البيانات بشكل متجانس، بحيث لكل حقل نوع من البيانات محدد مسبقا، إدخال البيانات يعني إضافة قيمة مناسبة لكل حقل من الجدول في نفس السطر، ليتم بعد ذلك إنشاء سطر جديد لإدخال بيانات أخرى، يدعى هذا السطر بالسجل.

في SQL Server Management Studio يمكن إدخال البيانات بعدة طرق:

- من خلال متصفح الكائنات (Object Explorer)
- من خلال تحرير شيفرة SQL من نافذة الاستعلامات
- عن طريق استيراد البيانات من قاعدة بيانات أخرى

¹ هناك أيضا برنامج EMS SQL Server Manager بإصدار Lite Free Edition مجاني وله إصدار آخر تجاري.

- أو عبر أحد برامج التطوير مثل Microsoft Access, Microsoft Visual Basic, Borland C++ Builder, Microsoft Visual C++, Borland/CodeGear Delphi, Microsoft Visual Basic, C#, Visual C#, J#, وغير ذلك...

إدخال البيانات عن طريق متصفح الكائنات:

من أسهل وأسرع الطرق لإدخال البيانات في SQL Server Management Studio استعمال متصفح الكائنات، بمجرد فتح قاعدة بيانات لاستعراض الجداول يمكنك فتح أي جدول وإدخال البيانات في سطر جديد، أو تعديل البيانات التي يخزنها الجدول.

قد يرفض SQL Server Management Studio ويمنعك من إدخال البيانات في بعض الحالات إما لعدم تطابق نوع البيانات المدخلة مع نوع بيانات الحقل التي يدعمها، أو لوجود قيود أو شروط لإدخال البيانات في هذا الحقل.

تطبيق عملي: إدخال البيانات

1- افتح SQL Server Management Studio واتصل بالخادم

2- انقر بالزر الأيمن على اسم الخادم واختر New Query

3- في نافذة الاستعلام اكتب شيفرة SQL التالية:

```
CREATE DATABASE WorldStatistics;
GO
USE WorldStatistics;
GO
CREATE TABLE Countries
(
    [Country Name] VARCHAR(80),
    Area INT,
    Population BIGINT,
    Capital VARCHAR(50),
    [Internet Code] char(2)
);
GO
```

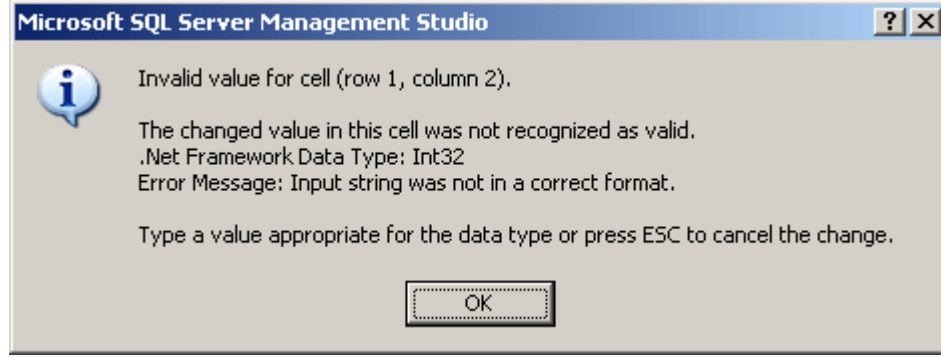
4- F5 لتنفيذ الاستعلام

5- أغلق نافذة الاستعلام واحفظه باسم Countries

6- انقر بالزر الأيمن على Databases واختر Refresh

7- قم بتوسيع البند Databases ثم WorldStatistics ثم Tables

- 8- إذا لم يظهر الجدول Countries انقر بالزر الأيمن على Tables واختر Refresh
- 9- انقر بالزر الأيمن على الجدول Countries واختر Open Table.
- 10- أدخل في الحقل CountryName القيمة Cote d'Ivoire ثم اضغط على Enter واكتب 322460 ثم اضغط على Tab واكتب القيمة 16,393,221 لتشاهد عندئذ الرسالة:



- 11- اضغط على OK وأدخل القيمة 16393221 People واضغط على Tab، وحينها ستشاهد رسالة أخرى مفادها أن الحقل معدّ لاستقبال قيمة عدد طبيعي لا سلسلة نصية.
- 12- اضغط على OK وامسح الكلمة People
- 13- تحت الحقل Internet Code أدخل القيمة ci واضغط على Enter
- 14- اكتب تحت الحقل Capital الكلمة Yamoussoukro واضغط على Enter مرتين.
- 15- أكمل إدخال القيم وفق الجدول التالي:

Country Name	Area	Population	Capital	Internet Code
Cote d'Ivoire	322460	16393221	Yamoussoukro	ci
Panama	78200	3191319	Panama	pa
Australia	7686850	20264082	Canberra	au
Canada	9984670	33098932	Ottawa	ca
Iran	1648000	68688433	Tehran	ir

16- أغلق الجدول.

إدخال البيانات عن طريق SQL:

لإدخال بيانات عن طريق SQL:

- من خلال متصفح الكائنات انقر على اسم الجدول بالزر الأيمن واختر Script Table
As -> INSERT To -> New Query Editor Window
- لإدخال البيانات أكتب استعمال إضافة بيانات في الجدول.

باستعمال SQL يتم إضافة بيانات بواسطة الكلمة المفتاحية INSERT و VALUES وفق الصيغة التالية:

```
INSERT TableName VALUES(Column1, Column2, Column_n);
```

أو بشكل محدد أكثر وفق الصيغة التالية:

```
INSERT INTO TableName VALUES(Column1, Column2, Column_n)
```

يمثل TableName اسم الجدول الموجود في قاعدة البيانات الحالية (التي يتم العمل عليها) التي سيتم إدخال البيانات إليها، إذا لم يعثر مترجم SQL على اسم الجدول فسيعيد رسالة خطأ. العبارة VALUES تعلن عن أننا نود إدخال قيمة في الحقول التي سنسردها بين قوسين.

لأجل نوع البيانات BIT أدخل القيمة 0 أو 1.

لأجل نوع البيانات الأعداد الطبيعية (int, bigint, smallint) أدخل قيمة رقمية دون فاصلة.

لأجل نوع البيانات الأعداد العشرية (float, real, decimal, numeric) يمكنك إدراج الفاصلة (الفاصلة في Microsoft SQL Server هي النقطة ".").

لأجل نوع البيانات Date أدخل قيمة تاريخية صحيحة.

لأجل نوع البيانات السلاسل النصية أدخل سلسلة نصية بين شاليتين ("").

تطابق القيم مع الحقول:

عند إدخال القيم في الجدول باستخدام SQL ينبغي أن تتطابق القيم المدخلة لكل حقل مع نوع بيانات الحقل في كل موضع، وإذا أردت إدخال قيمة معدومة في حقل عددي يمكن كتابة الرقم 0، ولإدخال قيمة فارغة في حقل بيانات نصية يمكن إدخال سلسلة فارغة بكتابة شاليتين متجاورتين " للإشارة أننا نريد إدخال سلسلة نصية فارغة.

تطبيق عملي: إدخال البيانات مع تطابق نوع بيانات الحقول

1- اضغط من لوح المفاتيح على Ctrl+N لكتابة استعلام جديد

2- في نافذة الاستعلام اكتب ما يلي:

```
USE WorldStatistics;
GO
INSERT INTO Countries
VALUES('Angola', 1246700, 12127071, 'Luanda','ao');
GO
```

3- اضغط على F5 لتنفيذ الاستعلام

4- امسح الاستعلام واكتب الاستعلام التالي لإدخال عدة سجلات في الجدول:

```
USE WorldStatistics;
GO
INSERT INTO Countries
VALUES('Mexico', 1972550, 107449525, 'Mexico City','mx');
GO
INSERT INTO Countries
VALUES('South Africa', 1219912, 44187637, 'Pretoria','za');
GO
INSERT INTO Countries
VALUES('Iraq', 0, 0, 'Baghdad','iq');
GO
INSERT INTO Countries
VALUES('United States', 9826630, 0, '', '');
GO
INSERT INTO Countries
VALUES('Saudi Arabia', 2149690, 0, 'Riyadh', '');
GO
```

5- اضغط على F5 لتنفيذ الاستعلام.

إدخال البيانات عشوائيا:

في الطريقة السابقة التي أدخلنا بها البيانات ينبغي علينا معرفة موضع كل حقل. يوفر SQL طريقة أخرى لإدخال البيانات بحيث تتطلب هذه الطريقة معرفة اسم كل حقل بدلا عن موضعه، وبذلك يمكنك إدخال البيانات وفق ترتيب من اختيارك.

في هذه الطريقة لست مجبرا على إدخال قيمة NULL للخانات الخالية، بل يكفي تحديد الحقول التي ستأخذ قيمة غير خالية وباقي الحقول تأخذ قيمة خالية.

تطبيق عملي: إدخال البيانات عشوائيا

1- افتح نافذة استعلام جديدة واكتب الاستعلام التالي:

```
USE WorldStatistics;
```



```

GO
INSERT Countries([Country Name],Capital,[Internet
Code],Population,Area)
VALUES('China', 'Beijing', 'cn', 1313973713, 9596960)
GO

```

2- قم بتنفيذ الاستعلام (F5)

3- لإدخال بيانات أخرى أكتب الاستعلام التالي:

```

USE WorldStatistics;
GO
INSERT Countries(Capital, [Internet Code], [Country Name])
VALUES('Nouakchott', 'mr', 'Mauritania')
GO
INSERT Countries([Internet Code], Population, [Country Name])
VALUES('ro', 22303552, 'Romania')
GO
INSERT Countries(Area, [Country Name], Population)
VALUES(21040, 'El Salvador', 6822378)
GO
INSERT Countries(Capital, [Country Name])
VALUES('Phnom Penh', 'Cambodia')
GO

```

4- F5 لتنفيذ الاستعلام، ثم أغلق نافذة الاستعلامات دون حفظ الملف.

القيمة الخالية لحقل (NULL):

عند إدخال البيانات عادة من قبل مستخدم البرنامج تبقى بعض الحقول دون قيمة مدخلة، مما يعني أنها لا تحمل أية قيمة، وقد يقع هذا مع أي نوع من البيانات، لأجل ذلك ينبغي التفكير في هذه الحالة مسبقاً لأجل التعامل معها بالشكل الصحيح.

يجب التنبيه أن الخانات التي تحمل القيمة NULL، لا يعني أنها تحمل القيمة العددية 0 ولا القيمة سلسلة نصية فارغة"، القيمة الخالية تعني أن مستخدم البرنامج لم يدخل أية قيمة أو قام بحذف القيمة المدخلة أو منعه أحد قيود إدخال البيانات.

من أجل منع أو إتاحة إدخال قيم في أي حقل من الحقول يوفر SQL Server أحد الخيارين NULL أو NOT NULL عند إنشاء الحقل، باستخدام SQL Server Management Studio عند إنشاء الجدول يمكن تفعيل أو عدم تفعيل خاصية Allow Nulls التي تسمح بعدم إدخال قيم في الحقول، وبذلك يصبح مستخدم البرنامج مجبراً على إدخال قيمة عندما يحاول الانتقال إلى السجل الموالي.

منع أو إتاحة قيم خالية باستخدام SQL:

لتحديد خيار منع عدم إدخال قيم (أو إجبار المستخدم على إدخال قيم) لأي حقل يمكن كتابة العبارة NOT NULL مباشرة بعد اسم الحقل، وإذا كانت قيمة الحقل غير ضرورية بحيث يمكن للمستخدم عدم إدخال أي قيمة تستخدم عبارة NULL مباشرة بعد كتابة اسم الحقل، مثال:

```
CREATE TABLE Persons
(
    FirstName varchar(20) NULL,
    LastName varchar(20) NOT NULL,
    Gender smallint
);
GO
```

ملاحظة هامة:

إذا تم إنشاء الجدول سابقا وأدخلت إليه بيانات وتركت أحد قيم الحقول خالية، فلا يمكن حينها تغيير خاصية قبول القيم الخالية (Allow Nulls).

تطبيق عملي: تغيير خاصية قبول القيم الخالية

1- في متصفح الكائنات افتح من قاعدة البيانات WorldStatistics الجدول Countries

2- عطل خاصية قبول القيم الخالية Allow Nulls كما يلي:

Table - dbo.Countries		Summary
Column Name	Data Type	Allow Nulls
[Country Name]	varchar(80)	<input type="checkbox"/>
Area	int	<input checked="" type="checkbox"/>
Population	bigint	<input checked="" type="checkbox"/>
Capital	varchar(50)	<input checked="" type="checkbox"/>
[Internet Code]	char(2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

3- قم بغلق الجدول مع حفظ التعديلات.

القيمة الافتراضية للحقل:

تحتوي بعض السجلات أحيانا على نفس القيمة في حقل معين في الجدول بكامله، وحتى لا يكون مستخدم البرنامج مضطرا لإدخال نفس القيمة لكل السجلات يمكن تحديد قيمة

افتراضية لهذا الحقل، بحيث كلما قام المستخدم بإضافة سجل جديد تظهر القيمة الافتراضية في الحقل تلقائياً.

تحديد قيمة افتراضية:

يمكن تحديد القيمة الافتراضية للحقل عند إنشائه، لإنشاء قيمة افتراضية لأي حقل باستخدام SQL Server Management Studio قم بتحديد الحقل من الجهة العليا لواجهة تصميم الجدول ثم حدد من الجانب السفلي للواجهة خاصية Default Value حسب نوع بيانات الحقل:

- سلسلة نصية (char, varchar, text): اكتب القيمة الافتراضية بين شالنتين ("").
- قيمة عددية: اكتب قيمة عددية متوافقة مع نوع البيانات المحدد (Tinyint أو int).
- تاريخ وتوقيت: التاريخ من الشكل MM/DD/YYYY أو YYYY/MM/DD مباشرة أو بين شالنتين، والتوقيت حسب إعدادات النظام في Regional Settings.
- قيمة منطقية bit: أدخل القيمة الافتراضية 0 لأجل القيمة False، أو أي عدد آخر للقيمة True.

تحديد قيمة افتراضية برمجياً:

لتحديد قيمة افتراضية لأي حقل برمجياً من خلال SQL تستخدم الكلمة المفتاحية DEFAULT بعد كتابة نوع بيانات الحقل وقبل رمز الفاصلة، متبوعة برمز المساواة (=) ثم القيمة الافتراضية للحقل، مثال:

```
CREATE TABLE Employees
(
    FullName VARCHAR(50),
    Address VARCHAR(80),
    City VARCHAR(40),
    State VARCHAR(40) DEFAULT = 'NSW',
    PostalCode VARCHAR(4) DEFAULT = '2000',
    Country VARCHAR(20) DEFAULT = 'Australia'
);
GO
```

يمكن لمدخل البيانات أن يغير قيمة الحقل من القيمة الافتراضية إلى قيمة أخرى كما يمكنه حذفها أيضاً.

الحقول التعريفية Identity Column:

يوفر SQL إمكانية إنشاء حقول تعريفية (Identity Column) تأخذ قيمة وحيدة في كل جدول، يسمح هذا النوع من الحقول بتمييز البيانات عن بعضها البعض، وبذلك يمكن تفادي وقوع أي حذف أو تعديلات على السجل أو السجلات غير المرغوب فيها.

تأخذ الحقول التعريفية تلقائياً قيم عددية صحيحة (Integer) مباشرة بعد إنشاء سجل جديد، ولا يمكن لمستخدم البرنامج (مدخل البيانات) أي يحدد لها أي قيمة أخرى، ولا يمكن إنشاء حقول تعريفية إلا عند إنشاء الجدول، كما لا يمكن إنشاء أكثر من حقل تعريفي في نفس الجدول.

إنشاء حقل تعريفي:

لإنشاء حقل تعريفي باستخدام SQL Server Management Studio بعد تحديد اسم الحقل ونوعه (int في الغالب) في الجزء الأعلى من نافذة تصميم الجدول، إضبط من الجزء السفلي خاصية Is Identity على القيمة Yes من المجموعة Identity Specification.

وتحتها مباشرة خاصية Identity Seed property التي تمثل القيمة الابتدائية للرقم التعريفي مضبوطة على القيمة 1 افتراضياً، وتحتها أيضاً خاصية Identity Increment التي تمكنك من تحديد قيمة تزايد الرقم التعريفي وقيمتها الافتراضية 1.

تطبيق عملي: إنشاء حقل تعريفي

1- من متصفح الكائنات حدد قاعدة البيانات WorldStatistics ثم انقر بالزر الأيمن واختر

...New Table

2- اكتب اسم الحقل الأول ContinentID واضغط على Tab ثم اكتب int كنوع الحقل ثم

اضغط على F6 للانتقال للجزء الأسفل

3- في الجزء الأسفل انتقل إلى التبويب Identity Specification وانقر مرتين على قيمة

الخاصية Is Identity لضبطها على القيمة Yes

4- أكمل حقول الجدول على الشكل التالي:

اسم الحقل	نوع البيانات	إتاحة قيمة خالية
ContinentID	int	Unchecked
Continent	varchar(80)	
Area	bigint	
Population	bigint	

5- أغلق الجدول واحفظه باسم Continents.

إنشاء حقل تعريفي باستخدام SQL:

لإنشاء حقل تعريفي برمجياً تكفي الإشارة إلى ذلك بعد الاسم ونوع البيانات باستخدام الكلمة المحجوزة **IDENTITY**، يليها قوسين وبينهما تكتب القيمة الابتدائية للحقل التعريفي متبوعة بفاصلة ثم قيمة الزيادة، هنا مثال:

```
CREATE TABLE StoreItems(
ItemID int IDENTITY(1, 1) NOT NULL,
Category varchar(50),
[Item Name] varchar(100) NOT NULL,
Size varchar(20),
[Unit Price] money);
GO
```

دوال إدخال البيانات:

أثناء إدخال البيانات باستخدام SQL يمكنك استدعاء أي دالة تقوم بإرجاع قيمة قابلة للاستعمال كقيمة مدخلة إلى الحقل، يمكنك إنشاء الدوال الخاصة بك واستخدامها، كما يمكنك استخدام أحد الدوال المضمنة في SQL Server.

استخدام الدوال المضمنة:

من أجل إشراك أحد الدوال خلال إدخال البيانات ينبغي أن تكون الدالة متاحة للاستخدام ومناسبة لنوع البيانات، ويمكن أيضاً استخدام أحد الدوال المضمنة في SQL Server. وقبل إنشاء دالة ألق نظرة ببحث صغير في الانترنت فربما تكون متداولة بين المبرمجين.

استخدام الدوال المدمجة قد يوفر عليك العناء في كثير من الحالات. مثلاً، لديك قاعدة بيانات باسم AutoRepairShop وفيها جدول يخزن بيانات صيانة طلبات الزبائن:

```
CREATE TABLE RepairOrders
(
RepairID int Identity(1,1) NOT NULL,
CustomerName varchar(50),
```

```

    CustomerPhone varchar(20),
    RepairDate DateTime
);
GO

```

عند إدخال البيانات في هذا الجدول يقوم المستخدم بإدخال اسم العميل ورقم هاتفه، وهنا يمكنك مساعدة المستخدم برمجيا بإدخال التاريخ الحالي تلقائيا، للقيام بذلك يمكنك استدعاء الدالة المضمنة GETDATE، أمثلة:

```

INSERT INTO RepairOrders(CustomerName, CustomerPhone, RepairDate)
VALUES('Annette Berceau', '301-988-4615', GETDATE());
GO
INSERT INTO RepairOrders(CustomerPhone, CustomerName, RepairDate)
VALUES('(240) 601-3795', 'Paulino Santiago', GETDATE());
GO
INSERT INTO RepairOrders(CustomerName, RepairDate, CustomerPhone)
VALUES('Alicia Katts', GETDATE(), '(301) 527-3095');
GO
INSERT INTO RepairOrders(RepairDate, CustomerPhone, CustomerName)
VALUES(GETDATE(), '703-927-4002', 'Bertrand Nguyen');
GO

```

يمكنك استعمال دوال ضمن عمليات حسابية، ثم استخدام النتيجة كقيمة مدخلة، أيضا يمكنك أيضا استدعاء الدالة التي تأخذ معلما أو أكثر؛ وتأكد من احترام قواعد تمرير القيم عند استدعاء الدوال، إذا لم تكن أي من الدوال المدمجة تفي بالغرض، يمكنك إنشاء دالة خاصة بك باستخدام التقنيات التي تطرقنا إليها في الدرس الخاص بالدوال.

استخدام عبارات حسابية لإدخال البيانات (Using Formula):

هناك طرق مختلفة يمكنك من خلالها مساعدة المستخدم عند إدخال البيانات، بالإضافة إلى استخدام الدوال يمكنك إنشاء عبارات حسابية عند إنشاء الجدول، باستخدام مصمم الجداول أو عن طريق SQL.

إنشاء عبارات حسابية أثناء تصميم الجدول:

لإنشاء تعبير حسابي عند إنشاء جدول باستخدام SQL Server Management Studio، في القسم العلوي من واجهة تصميم الجداول حدد اسم الحقل، ثم حدد من أسفل الواجهة الخاصية (Formula) من التبويب Computed Column Specification، ثم أدخل التعبير المناسب. مثال:

Column Name	Data Type	Allow Nulls
SquareID	int	<input type="checkbox"/>
Side	decimal(8, 3)	<input checked="" type="checkbox"/>
Perimeter		<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
(General)	
(Name)	Perimeter
Allow Nulls	Yes
Data Type	
Default Value or Binding	
Precision	18
Scale	0
Table Designer	
Collation	<database default>
Computed Column Specification	
(Formula)	Side * 4
Is Persisted	No
Condensed Data Type	

إنشاء عبارات حسابية باستخدام SQL:

لإنشاء تعبير حسابي باستخدام SQL أكتب في الاستعلام الذي تستخدمه لإنشاء الجدول، بعد كتابة اسم الحقل الكلمة AS ويليهما التعبير المناسب، مثال:

```
CREATE TABLE Circle
(
    CircleID int identity(1,1) NOT NULL,
    Radius decimal(8, 3) NOT NULL,
    Area AS Radius *Radius * PI()
);
GO
```

استخدام تعبير حسابي أثناء إدخال البيانات:

عند إدخال البيانات ينبغي إدخال قيمة للعمود الذي يحتوي على التعبير الحسابي؛ حينها يقوم مترجم SQL بتوليد حاصل العبارة تلقائياً، مثال إدخال البيانات لحقل حساب مساحة الدائرة للجدول أعلاه:

```
INSERT INTO Circle(Radius) VALUES(46.82);
GO
INSERT INTO Circle(Radius) VALUES(8.15);
GO
```

```
INSERT INTO Circle(Radius) VALUES(122.57);  
GO
```

قيود إدخال البيانات (Constraints):

عند إدخال البيانات في بعض الأعمدة وبعد تحديد نوع البيانات، قد ترغب في تحديد مجموعة من القيم التي يسمح (أو لا يسمح) بإدخالها، للتعرف على موافقة القيمة المدخلة للشروط المحددة، يوفر SQL ما يعرف بـ **قيود الفحص (Constraints)**.

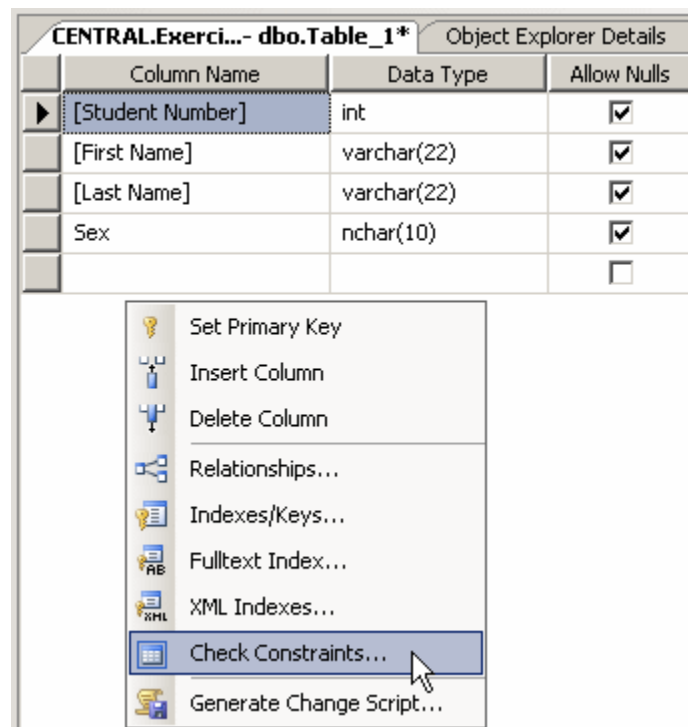
قيود الفحص (أو قيود إدخال البيانات) هي عمليات تعيد قيمة منطقية يقوم بها مترجم SQL، يقوم المترجم بفحص عملية الإدخال فيما إذا كانت موافقة للشروط (القيود) المحددة في الحقل، وبعد كل عملية فحص يعيد المترجم أحد القيمتين True (وبالتالي قبول القيمة وإدخالها في الحقل) أو False (رفض القيمة وعدم إدخالها في الحقل).

إنشاء قيود الفحص يتم عند إنشاء الجدول.

إنشاء قيود إدخال البيانات:

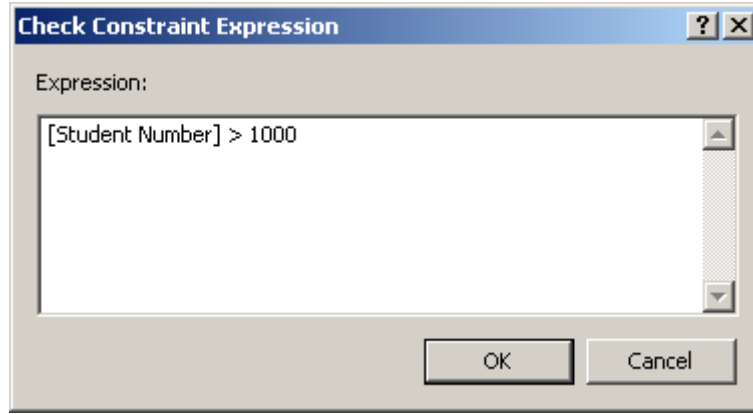
لإنشاء قيد فحص البيانات، انقر بالزر الأيمن عند إنشاء الجدول واختر Check

.Constraints



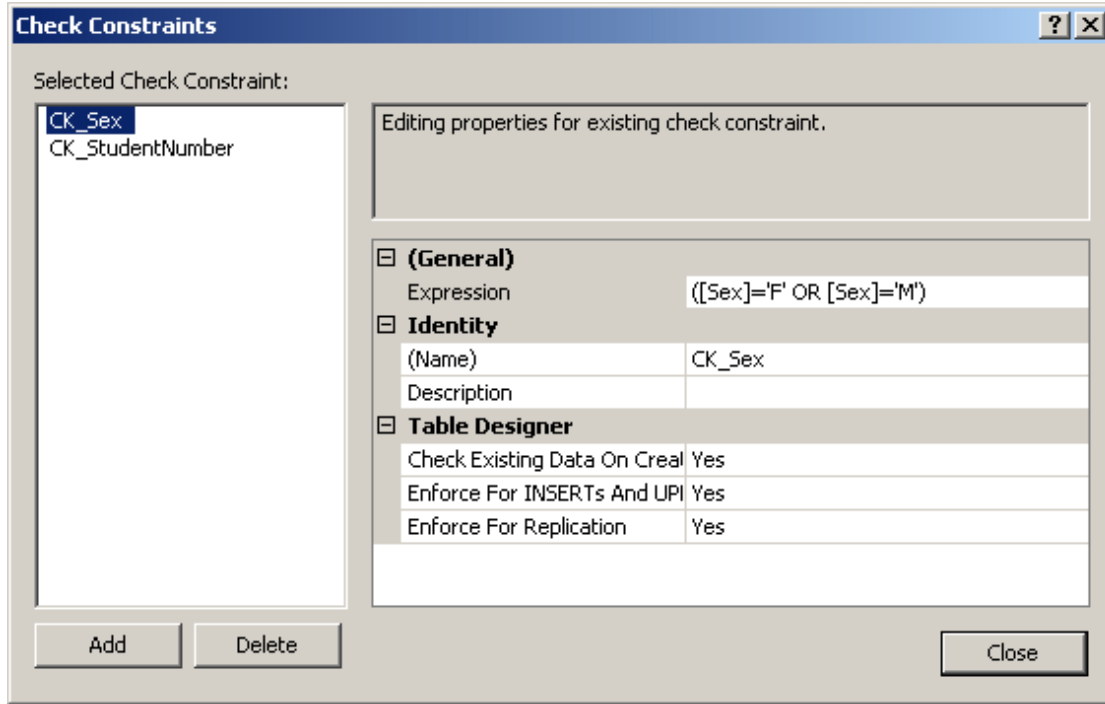
سيفتح مربع الحوار قيود البيانات، من هناك يمكنك النقر على زر الإضافة لإضافة قيد، ولأن القيد هو غرض بحد ذاته فيجب إذا إعطاؤه اسماً يميزه، وأهم خصائص القيد هي الآلية التي ستستخدم في التحقق من القيم، تكتب هذه الآلية على شكل عبارة حسابية، لإنشاء القيد يمكنك النقر على زر الخاصية Expression. وهنا ستواجه مربع الحوار لكتابة العبارة الحسابية (Check Constraint Expression).

لإنشاء التعبير أكتب أولاً اسم الحقل الذي سيطبق عليه القيد متبوعاً بقوسين، بداخل الأقواس يمكنك استخدام العمليات المنطقية AND/OR التي تدعمها SQL، في هذا المثال يتم التحقق من القيمة المدخلة حيث تقبل إذا كان عدد الطلاب في الجدول Students أكبر من 1000:



بعد إنشاء التعبير يمكنك النقر على زر الموافقة، إذا كان التعبير غير صحيح، ستتلقى خطأً وتعطى فرصة لتصحيحه.

يمكنك إنشاء العديد من القيود بحسب ما تقتضيه بيانات الجدول الخاص بك:



بعد إنشاء القيود وتحديد العبارات المناسبة، انقر على Close.

إنشاء قيود الفحص برمجياً:

برمجياً باستخدام SQL لإنشاء قيد التحقق يجب أولاً إنشاء جميع حقول الجدول بما فيها الحقل الذي سيطبق عليه القيد، وقبل غلق القوس نهاية تعريف الجدول، يمكنك تعريف القيد باستخدام الصيغة التالية:

```
CONSTRAINT name CHECK (expression)
```

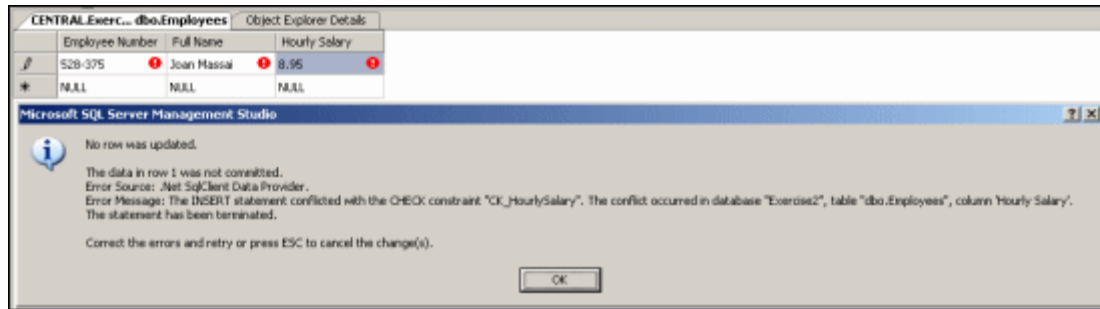
يجب تحديد اسم القيد (name) كما يجب كتابة عبارة صحيحة (expression)، وبداخل الأقواس التي تتبع الكلمة CHECK، أدخل التعبير الذي سيتم تطبيقه على الحقل، فيما يلي مثال التأكد من أن الراتب المحدد للموظف أكبر من 12.50:

```
CREATE TABLE Employees
(
    [Employee Number] nchar(7),
    [Full Name] varchar(80),
    [Hourly Salary] smallmoney,
    CONSTRAINT CK_HourlySalary CHECK ([Hourly Salary] > 12.50)
);
```

من المهم التأكد أن قيد التحقق من البيانات لا يمثل عبارة حسابية ولا دالة، وإنما يحتوي على تعبير حسابي كما قد يحتوي على دالة.

بعد إنشاء قيد (أو قيود) للجدول، يمكن في متصفح كائنات مشاهدته في البند Constraints تحت بند الجدول الذي تم إنشاء القيد عليه.

بعد تطبيق القيد، إذا قام المستخدم (أو استعلام SQL) بإدخال قيمة غير مسموح بها باعتبار القيد، فسيرسل SQL Server رسالة خطأ تخبر بذلك، مثال:



يمكن استخدام أحد الدوال المضمنة مع SQL Server أو إنشاء دالة جديدة واستعمالها في كتابة قيد لفحص البيانات كبديل عن التعبير الحسابي الذي يستخدم العمليات التقليدية.

أشكال وطرق أخرى لإدخال البيانات:

- **خاصية RowGuid:**

تسمح هذه الخاصية بتحديد الحقل الذي يحمل في الخاصية Identity القيمة Yes، بمثابة العمود .ROWGUID.

- **خاصية Collation:**

لأن لغات العالم تختلف في استخدام آليات من الأحرف الأبجدية، مما يمكن أن يؤثر على طريقة بعض خوارزميات الفرز أو الاستعلامات المطبقة على البيانات، يمكنك أن تحدد آلية لغة معينة بأن تطبق على قاعدة البيانات دون غيرها، وذلك من خلال الخاصية Collation إذا كنت على اطلاع بهذه الآليات. بخلاف ذلك عليك قبول الخيار الافتراضي المحدد للجدول.

استيراد البيانات:

هناك وسيلة أخرى لإدخال البيانات تتلخص في استيراد البيانات من قاعدة بيانات أخرى، يوفر SQL Server تقنيات متعددة لاستيراد البيانات من مصادر خارجية مختلفة.

من أشهر أنواع البيانات التي يمكن استيرادها إلى الخادم (والتي تتوفر عليه أغلب برامج تسيير قواعد البيانات)، هو ملف نصي ذو نسق خاص، بحيث يكون هذا الملف مهياً بشكل مناسب، من أجل تمييز الحقول من السجلات والبيانات عن بعضها البعض يتم وضع رمز فاصل بينها، هذا الفاصل يمكن أن يكون علامة تنصيص أو رمز القاطعة (;) أو غير ذلك، يعرف هذا النوع من الملفات النصية عادة باللاحقة CSV.

بعد استيراد البيانات ينبغي التحقق من مطابقتها لهيكل الجدول وربما استوجب ذلك تعديلاً لحقولها.

تطبيق عملي: استيراد البيانات من مصدر خارجي

1- احفظ النص الآتي في ملف نصي:

```
1.Sebastien.Porter,2/12/1995,Male,4442 Shane Ave,Silver Spring,MD,20902,(301) 591-6236,porters@rosh.md.us,John and Christine Porter,0,John Porter,(202) 662-4825
2.Suzie.Hoak,10/2/1991,Female,1866 Old Conference Rd. #1412,Hyattsville,MD,20782,(301) 736-0002,hoaks@rosh.md.us,Christiane Hoak,0,Christiane Hoak,(301) 364-0221
3.Annoinette.Clark,4/10/1996,Female,282 Centerway st.,Rockville,MD,20875,(301) 598-9292,clarka@rosh.md.us,Daniel and Henriette Clark,0,Henriette Clark,(301) 598-9292
4.Koko.Domba,2/5/1990,Male,5782 South Waterfront Blvd #14D2,Alexandria,VA,20212,(703) 363-1066,domba@rosh.md.us,Dr. Julie and Mr. Jeremiah Domba,0,Jeremiah Domba,
5.Jane.West,2/1996,Female,980 Washington Blvd.,Bethesda,MD,20872,(301) 515-2831,westj@rosh.md.us,Kirsten Farmer,1,Kirsten Farmer
6.Catherine.Chang,6/12/1997,Female,2186 Holm St. N.E.,Washington,DC,20010,(202) 434-0034,change@rosh.md.us,Julie Best and Dr. Peter Chang,0,Emie Change,(202) 215-6663
7.Nehemiah.Dean,12/2/1995,Male,442 Lakefront Ave.,Silver Spring,MD,20910,(301) 938-2763,deann@rosh.md.us,Marie Rodnat,1,Marie Rodnat,(703) 434-8756
8.Sherry.Ashburn,10/7/1995,Female,1771 N. Frederick Rd.,Chevy Chase,MD,20870,(301) 419-3346,ashburns@rosh.md.us,Shelia and Patrick Ashburn,0,Shelia Ashburn,(703) 292-8340
9.Santos.Pedroco,5/5/1998,Male,14277 Grand Misty Dr.,Arlington,VA,20202,(703) 652-4412,pedroco@rosh.md.us,Dr. Julia Santana,0,(202) 938-7772
10.Mohamed.Hussein,1/5/1998,Male,888 Democracy Rd. #D8,Washington,DC,20008,(202) 556-4766,husseinm@rosh.md.us,Drs. Phyllis and Ezra Hussein,0,Dr. Phyllis Hussein,(202) 631-8855
11.Dean.Chen,5/2/1997,Male,4608 St. Leonard St. Vienna,VA,20204,(703) 518-3372,chend@rosh.md.us,Alexandra Chen,1,Alexandra Chen
12.Ruby.DeGaram,10/11/1998,Female,5782 South Wells St.,Silver Spring,MD,20906,(301) 430-7712,degaram@rosh.md.us,Lynn and Anthony DeGaram,0,Anthony DeGaram,(301) 938-1220
13.Carole.Chance,10/22/1990,Female,645 Farmers Ave.,Rockville,MD,20875,(301) 724-1810,chanccc@rosh.md.us,Carole and Andy Chance,0,Andy Chance,(301) 422-1001
14.Justin.Vittas,5/4/1991,Male,12881 Northpoint Blvd.,#1410,Bethesda,MD,20815,(301) 549-4004,vittasj@rosh.md.us,Clarice Vittas,1,(301) 549-0020
15.Ismael.Zara,5/25/1990,Male,9009 Kentland St.,Laurel,MD,20707,(301) 356-1826,zara@rosh.md.us,Melinda and Robert Zara,0,Robert Zara,(202) 978-6642
16.Anselme.Waters,7/25/1990,Male,1872 Hunters Dr.,Washington,DC,20002,(202) 525-0160,watersa@rosh.md.us,Claudette and Dr. Celestin Waters,1,Celestin Waters,(703) 894-6624
17.Brenda.Lobo,10/5/1989,Female,2490 Chanting Rd.,Alexandria,VA,20214,(703) 894-2888,lobo@rosh.md.us,Priscilla and Michel Lobo,0,(703) 894-2888
18.Suzanna.Verde,3/12/1988,Female,3844 John M.Gill St.,Silver Spring,MD,20904,(301) 831-2420,verdes@rosh.md.us,Suzanne and Thommy Verder,0,Thommy Verde
19.Danilo.Chico,6/20/1997,Male,291 N. Nixon Av.,Chevy Chase,MD,20872,(301) 364-9212,chicod@rosh.md.us,Michelle and Stephen Chico,0,Stephen Chico,(202) 299-7001
20.Mircey.Franse,10/8/1991,Female,1788 Old Montgomery Av.,Laurel,MD,20707,(301) 799-1112,fransem@rosh.md.us,Dr. Melanie and Cleve Franse,0,Melanie Franse,(301) 986-4445
21.Arlene.Andriamirano,2/15/1989,Female,1124 Old Tribute Av. #12E2,Arlington,VA,20212,(703) 593-2820,Andriam@rosh.md.us,Christiane Burton,1,Christiane Andriamirano,(703) 593-2820
22.Gabriele.Ledoux,6/23/1989,Female,752 West Farm Blvd.,Hyattsville,MD,20782,(301) 780-4424,ledouxg@rosh.md.us,Brenda and Patrick Fasten,0
23.Koko.Lobila,8/20/1990,Female,2888 Wolf Terrace,Washington,DC,20002,(202) 538-2008,lobilak@rosh.md.us,Aquilla and Dr. Diallo Lobila,0,Aquilla Lobila,(301) 752-0002
24.Paul.Farms,4/30/1996,Male,4490 Presidents Square,Silver Spring,MD,20902,(301) 695-7727,farmsp@rosh.md.us,Rebecca and Peter Farms,0,Rebecca Farms,(301) 696-8281
25.Lester.Bell,12/6/1995,Male,2882 April Av. #D4,Silver Spring,MD,20815,(301) 979-0032,belll@rosh.md.us,Bernadette and Paul Bell,0,Bernadette Bell,(301) 979-0032
26.Christian.Liss,10/8/1988,Male,902 Continental Rd.,Washington,DC,20020,(202) 389-7487,lissec@rosh.md.us,Sandra and Paul Liss,0,Sandra Liss,(703) 797-9762
27.Arlette.Duma,9/28/1995,Female,10008 Paloma Drive #1204,Bethesda,MD,20850,(301) 206-8623,dumat@rosh.md.us,Ursula Duma,1,(301) 206-8623
28.Harriette.Sans,8/2/1996,Female,402 Dublin Rd.,Alexandria,VA,22032,(703) 684-1500,sans@rosh.md.us,Thomas Sans,0,Thomas Sans,(703) 684-1500
29.Cliff.Fuller,4/19/1996,Male,6812 Thousand Way,Silver Spring,MD,20904,(301) 490-6390,cliff@rosh.md.us,Shelley and Dr. Oregon Fuller,0,Shelley Fuller
30.Thomas.Moore,4/22/1988,Male,11312 Crane Drive #12,Columbia,MD,21102,(410) 730-8100,moort@rosh.md.us,Deborah Moore,0,Deborah Moore,(410) 730-8100
31.Bernadette.Howerson,6/11/1996,Female,4440 Public Sq.,Silver Spring,MD,20910,(301) 899-1800,hobw@rosh.md.us,Caron and Rick Howerson,0,Caron Howerson,(301) 899-1800
32.Tim.Amoros,2/14/1989,Male,125 Eulah Rd.,Rockville,MD,20854,(301) 262-1717,amorost@rosh.md.us,Eduardo Amoros,0,Helena
33.Judith.Steinberg,2/22/1989,Female,9 Alberrton Circle #D,Takoma Park,MD,20912,(301) 322-7130,steinbergj@rosh.md.us,Jim,0,Jim,(301) 322-7130
34.Elena.Napolis,4/24/1991,Female,820 Chisom Rd.,Silver Spring,MD,20906,(301) 925-7041,napolis@rosh.md.us,Sandra and Pedro Napolis,0,Sandra Napolis,(301) 925-7041
35.Maurice.Walken,10/26/1990,Male,7200 St Francis Ave #16,Washington,DC,20010,(202) 583-4228,walkenm@rosh.md.us,(202) 583-4228
36.Charles.Edelman,12/26/1996,Male,54 Sommerset Rd.,Hyattsville,MD,20774,(301) 656-4441,edelman@rosh.md.us,Jasmine Beaulieu,1,Jasmine Beaulieu,(301) 656-4441
37.Arthur.Milley,5/11/1995,Male,3754 Bond St.,Silver Spring,MD,20906,(301) 229-1600,milleya@rosh.md.us,Jeffrey Milley,0,Jeffrey Milley,(301) 229-1600
38.Martin.Davis,10/14/1996,Male,856 Worry Lane,Bethesda,MD,20860,(703) 856-1100,davis@rosh.md.us,0
39.Ann.Miller,6/21/1997,Female,1948 Hagers St.,Rockville,MD,20855,(301) 571-0400,millenn@rosh.md.us,0
40.Millicent.Broadskey,9/10/1989,Female,2800 Freight Way,Silver Spring,MD,20910,(301) 879-4567,broadskeym@rosh.md.us,0
41.Victoria.Milchen,4/18/1988,Female,105 Military Rd.,Silver Spring,MD,20905,(301) 540-2300,milchen@rosh.md.us,Paul Milchen,1,Paul Milchen,(301) 540-2300
42.Arthur.Junger,8/8/1991,Male,3022 Connecticut Ave.,Chevy Chase,MD,20815,(301) 656-4060,jungera@rosh.md.us,Helene Pitts,1,(301) 656-4060
43.Martine.Quares,2/4/1996,Female,808 Fatucent Rd.,Washington,DC,20004,(202) 833-4167,quares@rosh.md.us,0,Cirace Zarra
44.Julie.Laurens,6/22/1988,Female,4012 Sunset Rd.,Rockville,MD,20854,(301) 649-6900,laurensj@rosh.md.us,0
45.Martha.Bastens,2/12/1995,Female,506 Smallwall Circle,Chevy Chase,MD,20840,(301) 736-8747,bastensm@rosh.md.us,0,(301) 780-4424
46.Jeanette.Hutchins,4/24/1988,Female,704 Summit Ave.,Silver Spring,MD,20910,(301) 662-0013,hutchinsj@rosh.md.us,0
47.Paul.Marly,10/20/1990,Female,2900 Cooks Rd.,Silver Spring,MD,20905,(301) 661-5050,marlyp@rosh.md.us,Arlette Hutchins,1,Arlette Hutchins,(301) 661-5050
48.Donnie.Mart,6/8/1990,Female,3002 Magnuson Ave.,Rockville,MD,20854,(301) 276-6068,martd@rosh.md.us,Alan Marly,1,Alan Marly,(301) 276-6068
49.George.Orion,12/22/1995,Male,12004 Shepherd Way,Bethesda,MD,20850,(301) 805-2233,orion@rosh.md.us,Jennifer Freemont,1
50.Albert.Linken,5/14/1988,Male,5909 Freemont Ave.,Chevy Chase,MD,20845,(301) 530-0540,linkena@rosh.md.us,Sandra Kallahan,1,(301) 530-0540
51.Ralph.Hagers,2/21/1990,Male,312 Friendship Ave.,Chevy Chase,MD,20845,(301) 805-6008,hagers@rosh.md.us,0
52.Charles.Laurel,8/20/1990,Male,4422 Davison Rd #1205,College Park,MD,20747,(301) 628-3200,laurelc@rosh.md.us,0,Stephen Laurel,(301) 628-3200
```

2- قم بحفظ الملف باسم Students.txt إلى القرص الصلب في جهازك.

Students.txt

3- باستخدام SQL Server Management Studio انقر بالزر الأيمن على Databases

واختار New Database...

4- أكتب ROSH واضغط Enter

5- من متصفح الكائنات انقر بالزر الأيمن على قاعدة البيانات ROSH واختر Tasks ثم

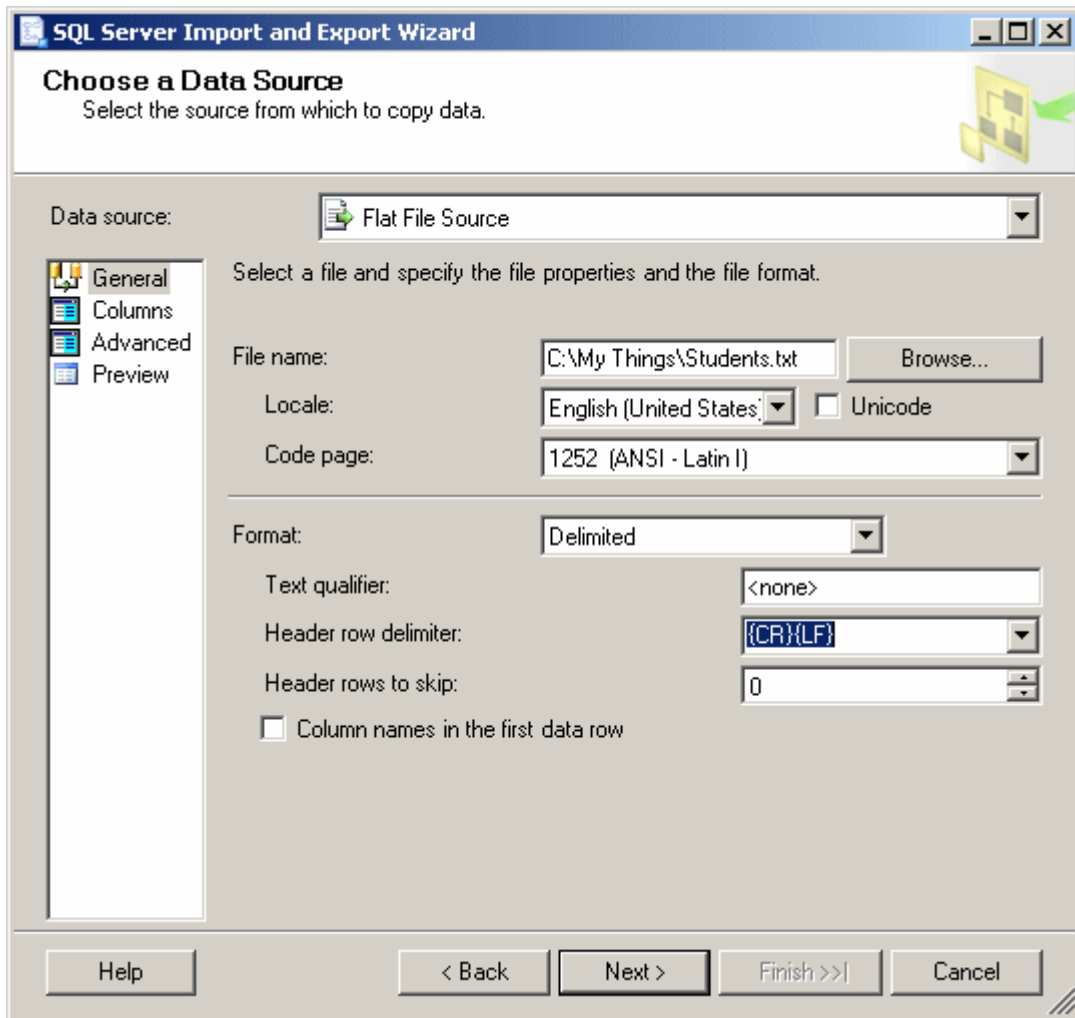
انقر على Import Data

6- من الصفحة الأولى من المعالج، انقر على Next

7- من الصفحة الثانية، اختر من مربع التحرير Data Source الخيار Flat File Source

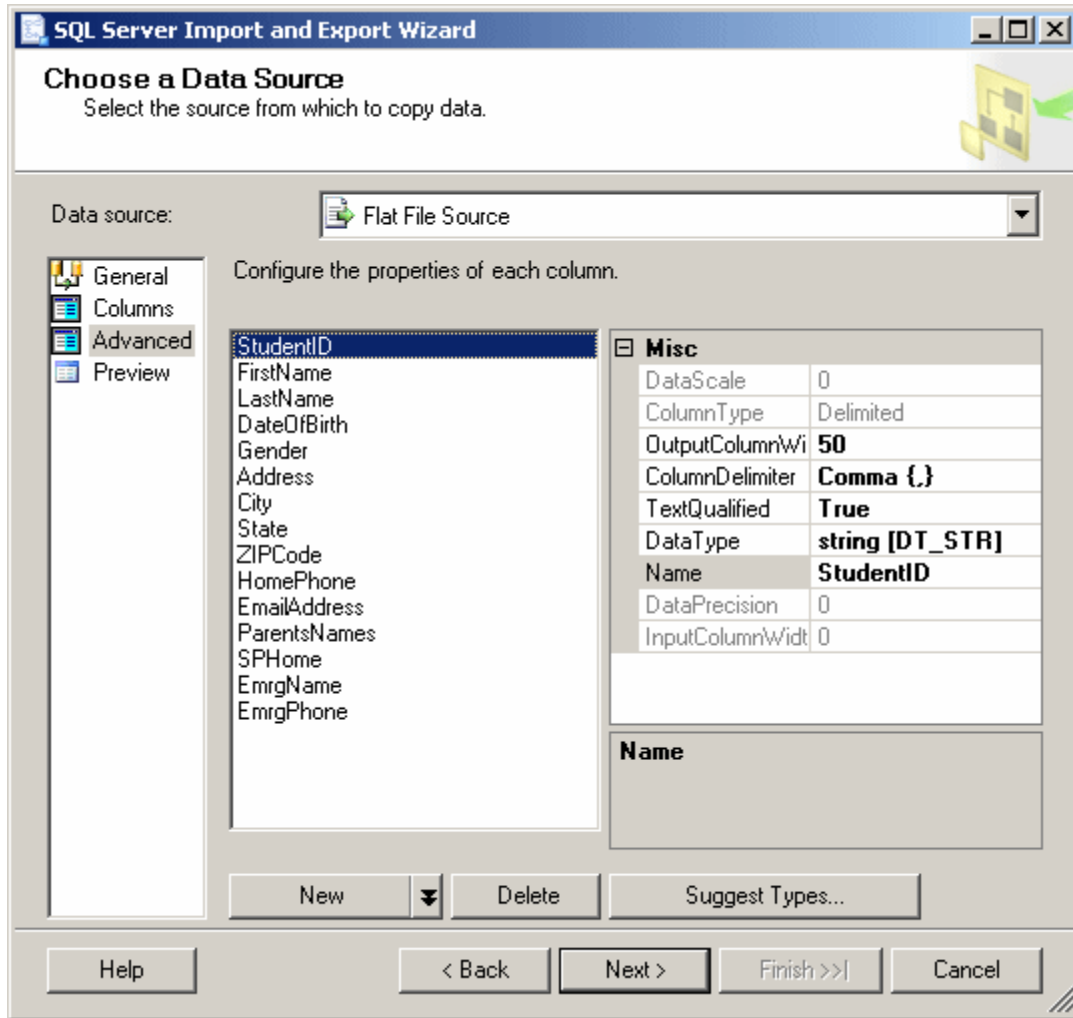
8- من الجانب الأيمن انقر على الزر Browse

9- حدد الملف Students.txt الذي حملته سابقا



10- تحت Data Source انقر على Advanced

11- بعد تحديد العمود في القائمة من الجهة اليمنى، انقر فوق Name واكتب StudentID



12- من القائمة التي في الوسط، انقر على كل حقل وغيره خاصيته Name من الجانب الأيمن على النحو التالي:

الحقل	اسم الحقل
Column0	StudentID
Column1	FirstName
Column2	LastName
Column3	DateOfBirth
Column4	Gender
Column5	Address
Column6	City
Column7	State
Column8	ZIPCode
Column9	HomePhone
Column10	EmailAddress
Column11	ParentsNames
Column12	SPHome

Column13	EmrgName
Column14	EmrgPhone

13- للاطلاع على قائمة الحقول، تحت Data Source، انقر على Columns

SQL Server Import and Export Wizard
Choose a Data Source
 Select the source from which to copy data.

Data source: Flat File Source

Specify the characters that delimit the source file:

Row delimiter: {CR}{LF}

Column delimiter: Comma (,)

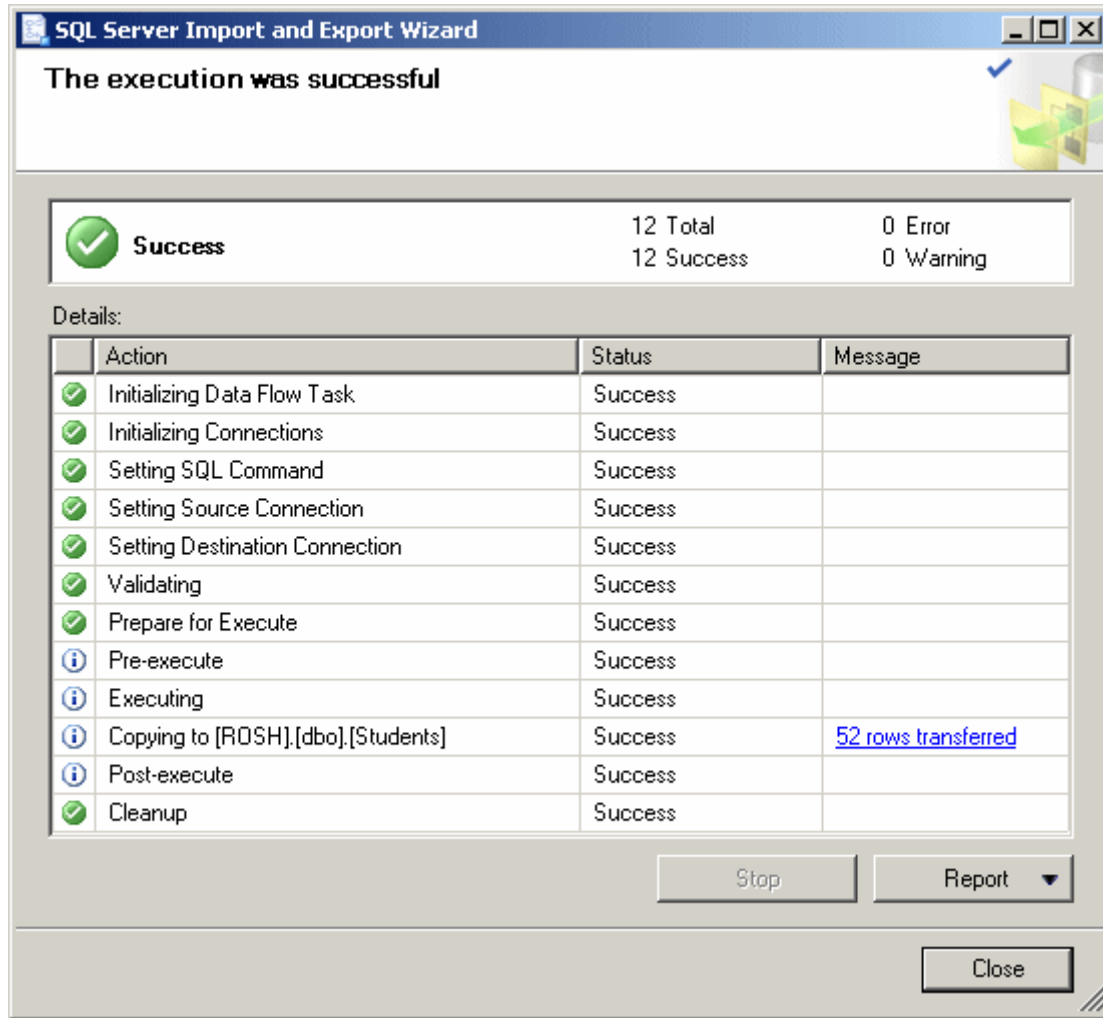
Preview rows 1-52:

StudentID	FirstName	LastName	DateOfBirth	Gender	Address
1	Sebastien	Porter	2/12/1995	Male	4442 Shane Ave
2	Suzie	Hoak	10/5/1991	Female	1866 Old Conf...
3	Antoinette	Clarck	4/10/1996	Female	282 Centerway...
4	Koko	Domba	2/5/1990	Male	8782 South W...
5	Janet	West	6/2/1996	Female	980 Washingt...
6	Catherine	Chang	6/12/1997	Female	2186 Holm St. ...
7	Nehemiah	Dean	12/2/1995	Male	442 Lakefront ...
8	Sherryl	Ashburn	10/7/1995	Female	1771 N. Freder...
9	Santos	Pacheco	5/5/1998	Male	14277 Grand ...
10	Mohamed	Husseini	1/5/1998	Male	888 Democrac...
11	Dean	Chen	5/2/1997	Male	4608 St. Leon...
12	Ruby	DeGaram	10/11/19...	Female	8782 South W...
13	Carole	Chance	10/22/19...	Female	445 Farmers Av.
14	Justin	Vittas	5/4/1991	Male	12881 Northpo...
15	Ismael	Zara	5/25/1990	Male	9009 Kentland...

Reset Columns

Help < Back Next > Finish >>| Cancel

14- انقر على التالي 4 مرات ثم انقر Finish



15- انقر على Close

16- من متصفح الكائنات قم بتوسيع قاعدة البيانات ROSH والبند Tables ثم انقر بالزر

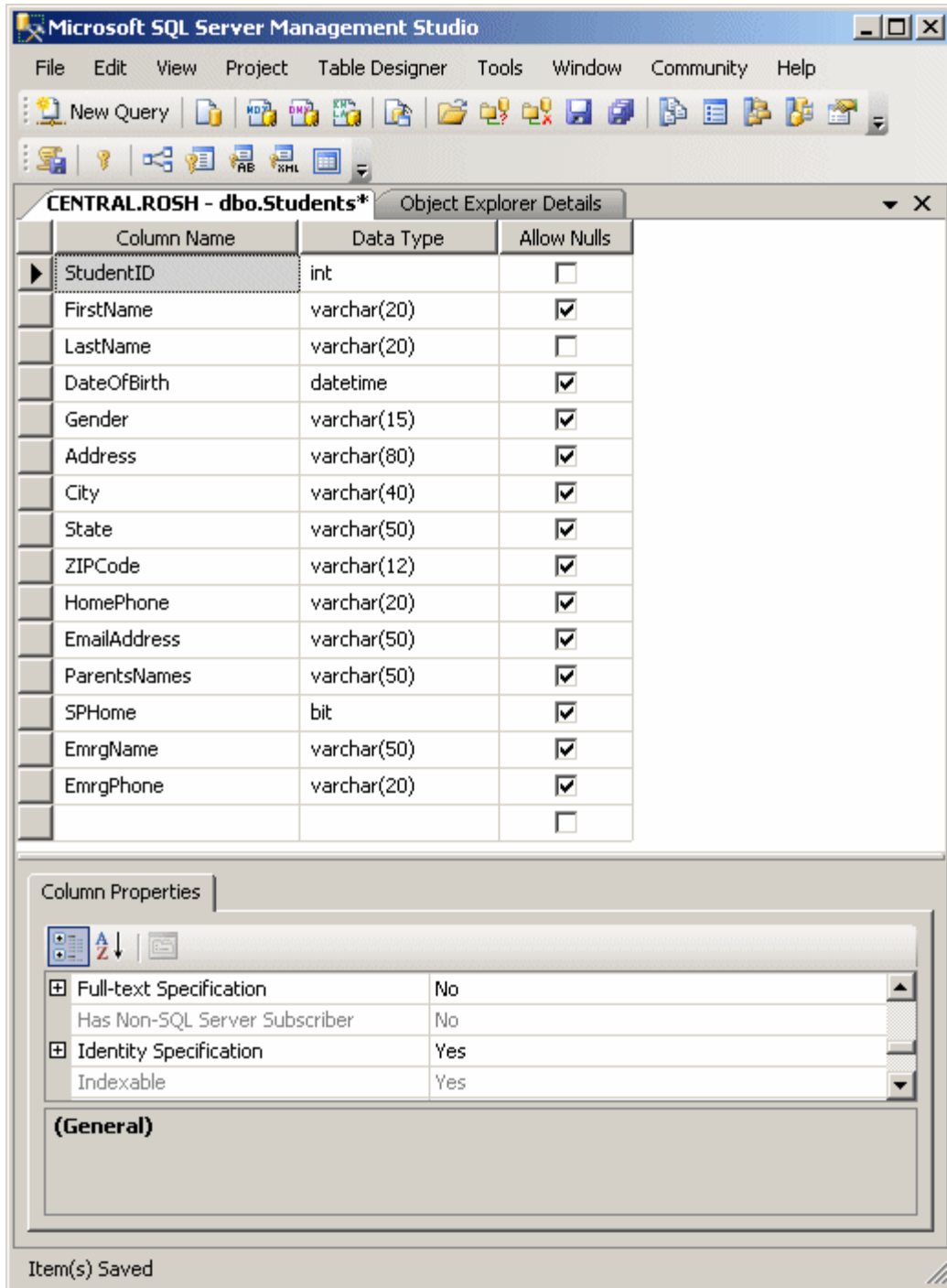
الأيمن على Students واختر Design

17- باعتبار أن الحقل StudentID محدد، اضغط على Tab وغير نوع بياناته int

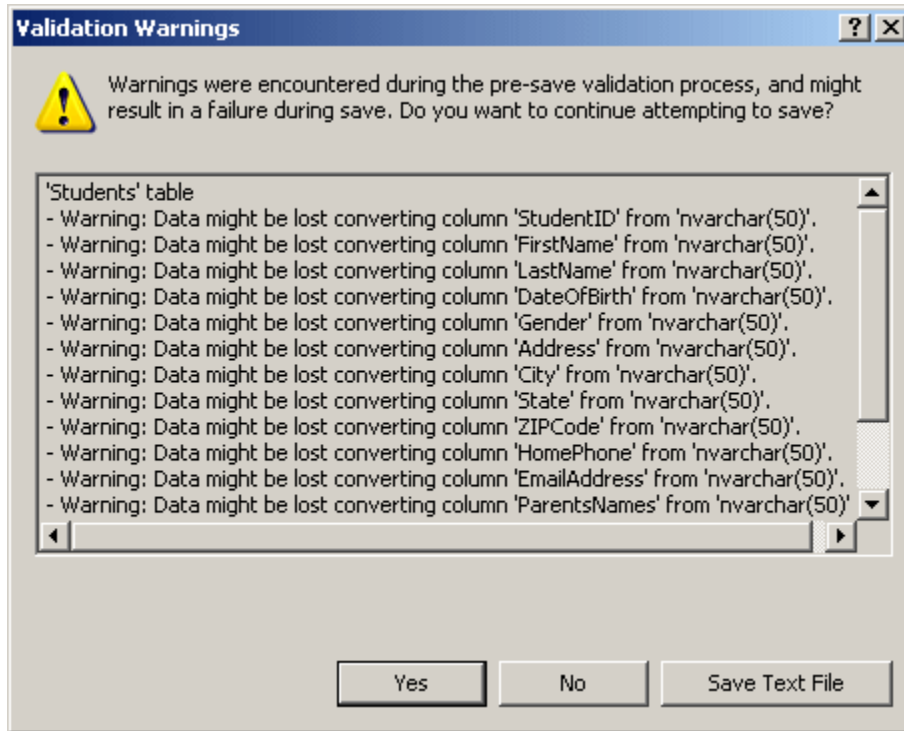
18- اضغط F6 وافتح التبويب Identity Specification، ثم انقر مرتين على Is Identity

لضبط قيمته على Yes

19- قم بتغيير الحقول الأخرى على النحو التالي:



20- لحفظ الجدول، انقر على الزر Save من شريط الأدوات القياسي



21- عند ظهور رسالة التنبيهات، انقر على Yes، ثم إغلاق الجدول

22- لعرض بيانات الجدول، انقر بالزر الأيمن في متصفح الكائنات على dbo.Students

وانقر على Open Table

فحص السجلات:

التحقق من وجود السجلات:

كمبرمج قواعد البيانات يمكنك الاطلاع على السجلات والتحقق من إدخال البيانات، هناك تقنيات مختلفة يمكنك استخدامها لأجل ذلك. من بينها الدالة المضمنة EXISTS التي يوفرها T-SQL، يتم استخدامها وفق الصيغة:

`BIT EXISTS(SELECT Something)`

تأخذ هذه الدالة وسيط (معلم) وحيد يجب أن يكون عبارة انتقاء **SELECT** التي تستخدم للحصول على قيمة يمكن فحص وجودها، مثلاً، رأينا سابقاً الجدول databases الذي يخزن أسماء قواعد البيانات التي تحتوي جميع قواعد البيانات المخزنة على الخادم، يمكنك استخدام الدالة **EXISTS()** للتحقق من وجود قاعدة بيانات معينة، والصيغة التي ستستخدم هي كالاتي:

```
IF EXISTS (
    SELECT name
```

```

FROM sys.databases
WHERE name = N'DatabaseName'
)

```

حيث *DatabaseName* يمثل اسم قاعدة البيانات.

انتقاء السجلات:

قبل أداء بعض العمليات على الجدول، يجب أولاً تحديد سجل أو أكثر. لتحديد أحد السجلات في واجهة استعراض البيانات، انقر بالزر الأيسر على الجانب الأيسر من الجدول:

VideoID	VideoTitle	Director	YearReleased	VideoLength	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	118 Minutes	R
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Eth...	NULL	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	R
*	NULL	NULL	NULL	NULL	NULL

يمكنك تحديد مجموعة من السجلات بالنقر على أول وآخر سجل تريد تحديده مع الاحتفاظ بالضغط على المفتاح **Shift**، أو اختيار مجموعة عشوائية مع الضغط على المفتاح **Ctrl**.

VideoID	VideoTitle	Director	YearReleased	VideoLength	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	118 Minutes	R
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Eth...	NULL	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	R
*	NULL	NULL	NULL	NULL	NULL

لتحديد كافة السجلات، انقر على الزر أعلى يسار العمود الأول:

VideoID	VideoTitle	Director	YearReleased	VideoLength	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	118 Minutes	R
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Eth...	NULL	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	R
*	NULL	NULL	NULL	NULL	NULL

بعد تحديد السجلات يمكنك حذفها أو نسخها إلى سجلات فارغة أو إلى برنامج خارجي آخر.

صيانة السجلات:

ويشمل ذلك عرض السجلات، البحث فيها، تعديلها، أو حذفها.

تعديل السجلات:

نعني بتعديل السجلات تغيير أحد قيم حقولها، لتعديل قيم أحد السجلات باستخدام SQL:

- في متصفح الكائنات يمكنك النقر بالزر الأيمن على الجدول ثم اختر Script Table As
ثم UPDATE To ثم New Query Editor Window .
- أو افتح نافذة استعلام جديدة واكتب ثم نفذ مباشرة استعلام تحديث البيانات.

للقيام بأحد عمليات صيانة السجلات تستخدم الكلمة المفتاحية **UPDATE** لتحديد الجدول الذي تريد إجراء عمليات عليه، يتم ذلك وفقا للصيغة العامة التالية:

```
UPDATE TableName  
SET ColumnName = Expression
```

في هذه الصيغة يجب تحديد اسم الجدول حيث المعامل *TableName*. تسمح الكلمة **SET** بتحديد قيمة جديدة للحقل *ColumnName* وفقا للعبارة *Expression* أو قيمة ثابتة مباشرة.

الاستعلام الآتي يقوم بإنشاء الجدول Videos بعد إنشاء قاعدة بيانات جديدة باسم VideoCollection، ثم يقوم بإدخال بعض السجلات إلى الجدول Videos كما يلي:

```
CREATE DATABASE VideoCollection;  
GO  
USE VideoCollection;  
GO  
CREATE TABLE Videos (  
    VideoID INT NOT NULL IDENTITY(1,1),  
    VideoTitle varchar(120) NOT NULL,  
    Director varchar(100) NULL,  
    YearReleased SMALLINT,  
    VideoLength varchar(30) NULL,  
    Rating varchar(6)  
);  
GO  
INSERT INTO Videos(VideoTitle, Director, YearReleased, VideoLength)  
VALUES('A Few Good Men', 'Rob Reiner', 1992, '138 Minutes');
```

```
INSERT INTO Videos(VideoTitle, Director, YearReleased, VideoLength)
VALUES('The Silence of the Lambs','Jonathan Demme',1991,'118
Minutes');
```

```
INSERT INTO Videos(VideoTitle, Director, VideoLength)
VALUES('The Distinguished Gentleman', 'James Groeling', '112
Minutes');
```

```
INSERT INTO Videos(VideoTitle, Director, VideoLength)
VALUES('The Lady Killers', 'Joel Coen & Ethan Coen', '104 Minutes');
```

```
INSERT INTO Videos(VideoTitle, Director, VideoLength)
VALUES('Ghosts of Mississippi', 'Rob Reiner', '130 Minutes');
GO
```

تعديل جميع السجلات دفعة واحدة:

تخيل أنك ترغب في تعديل جميع سجلات من أحد الحقول لأحد الجداول في وقت واحد. لا توجد طريقة خاصة لتحديث كافة السجلات من هذا جدول إلا عن طريق استعمال التحديث.

في لغة SQL يمكنك أن تستخدم الاستعلام UPDATE لتحديث كافة السجلات، مثال:

```
USE VideoCollection;
GO
UPDATE Videos
SET Rating = 'R';
GO
```

بهذا الاستعلام ستأخذ جميع السجلات من الجدول Videos قيمة R في الحقل Rating:

VideoID	VideoTitle	Director	YearReleased	VideoLength	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	118 Minutes	R
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Eth...	NULL	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	R
*	NULL	NULL	NULL	NULL	NULL

تحرير السجلات:

نعني بتحرير السجلات تغيير قيمة أحد الحقول لسجل محدد، للقيام بذلك افتح الجدول لعرض سجلاته أولاً، ثم قم بتحديد الحقل الذي سيجري عليه التحرير والقيمة التي سيتم تغييرها، ثم أسند إليه التعديلات المناسبة.

في SQL يجب أن يتعرف المترجم على موقع السجل بالتحديد ليقوم بتغيير قيمته. للقيام بذلك تستخدم كلمة التخصيص WHERE مع عبارة التحديث UPDATE وفق الصيغة التالية:

```
UPDATE TableName
SET ColumnName = Expression
WHERE Condition(s)
```

يسمح المشغل **WHERE** بتحديد القيمة التي سيتم تحريرها. ومن المهم جدا أن يكون المعيار المستخدم قادرا على التعرف بشكل فريد على القيمة المعدلة. في المثال التالي يقوم الاستعلام بتغيير السنة إلى 1996 حيث مدير الإنتاج Rob Reiner في الجدول السابق:

```
UPDATE Videos
SET YearReleased = 1996
WHERE Director = 'Rob Reiner';
```

في الجدول السابق، هناك سجلين يحملان اسم المخرج Rob Reiner، وعند تنفيذ هذا الاستعلام سيتم تغيير كافة السجلات التي تحمل اسم المخرج Rob Reiner، وإذا أردت استثناء السجلات التي لا تحتاج إلى هذا التغيير يجب استخدام الحقل التعريفي لتمييز السجل عن غيره، ذلك لأن مترجم SQL يقوم بإلحاق قيمة فريدة لكل سجل تميزه عن بقية السجلات.

المثال الآتي يستخدم لتحرير قيمة الحقل YearReleased رقم قياسي خاص:

```
UPDATE Videos
SET YearReleased = 1996
WHERE VideoID = 5;
GO
```

حذف جميع السجلات:

لحذف كافة السجلات مع الحفاظ على هيكل الجدول، قم بتحديد جميع السجلات واضغط الزر Delete. ستتلقى تنبيهها لأجل ذلك:

The screenshot shows a table named 'dbo.Videos' with the following data:

VideoID	VideoTitle	Director	YearReleased	VideoLength	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	118 Minutes	R
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Eth...	NULL	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	R
*	NULL	NULL	NULL	NULL	NULL

Below the table, a warning dialog box from Microsoft SQL Server Management Studio is displayed, stating: "You are about to delete 5 Row(s). Click Yes to permanently delete these rows. You won't be able to undo those changes." The dialog box has "Yes" and "No" buttons.

اختر Yes للموافقة على حذف جميع السجلات.

يمكن باستخدام SQL لحذف كافة السجلات وفق الصيغة التالية:

```
DELETE TableName;
```

حيث يمثل TableName اسم الجدول الذي سيتم حذف جميع سجلاته.

كن حذرا عند القيام بذلك لأنك لن تتمكن من استعادة السجلات التي تم حذفها كما أنك لن تتلقى أي تنبيه قبل ذلك.

حذف أحد السجلات:

لحذف أحد السجلات يمكنك النقر بالزر الأيمن أثناء استعراض الجدول على أحد الأسطر، ثم اختيار Delete من القائمة أو بالضغط مباشرة على الزر Delete، وستتلقى رسالة تحذير لتأكيد خيارك.

لحذف سجل باستخدام SQL يمكنك القيام بأحد العمليات:

- في متصفح الكائنات، يمكنك النقر بالزر الأيمن على الجدول ثم اختيار Script Table As ثم DELETE To وانقر على New Query Editor Window
- أو افتح نافذة استعلام جديدة واكتب ثم نفذ استعلام حذف السجلات.

لحذف سجل باستخدام لغة SQL تستخدم العبارة **DELETE FROM** وفق الصيغة:

```
DELETE FROM TableName  
WHERE Condition(s)
```

حيث يحدد العامل TableName اسم الجدول الذي سيحذف منه السجل، ويمثل العامل Condition(s) المعيار الذي يسمح بتحديد سجل أو مجموعة من السجلات التي تحمل مواصفات معينة.

مرة أخرى، تأكد من دقة المعايير المختارة بحيث لا يمكنها أن تحذف السجل الخطأ. هنا مثال لإزالة سجل معين من الجدول Videos:

```
DELETE FROM Videos  
WHERE VideoTitle = 'The Lady Killers';
```

وهذا المثال لحذف جميع سجلات الجدول Videos بطريقة أخرى:

DELETE FROM Videos;

تطبيق عملي: نهاية على الدرس

- 1- قم بغلق نافذة الاستعلام دون حفظ الملف
- 2- في متصفح الكائنات انقر بالزر الأيمن على قاعدة البيانات WorldStatistics، وانقر على Delete ثم وافق على خيار الحذف بالنقر على OK.

حوصلة الدرس:

رأينا في هذا الدرس:

- ✓ التحكم في السجلات
- ✓ التحكم في الجداول
- ✓ إدخال البيانات بعدة طرق
- ✓ القيم الافتراضية للحقول
- ✓ الحقول التعريفية
- ✓ العبارات الحسابية
- ✓ قيود إدخال البيانات
- ✓ استيراد البيانات
- ✓ انتقاء البيانات
- ✓ تحرير السجلات
- ✓ تعديل السجلات
- ✓ حذف السجلات

الكلمات المفتاحية، المشغلات والخصائص:

- 1- **NULL**
- 2- **NOT NULL**
- 3- **DEFAULT**
- 4- **IDENTITY**
- 5- Identity Specification
- 6- (Is Identity)

- 7- Identity Seed property
- 8- Identity Increment
- 9- **CONSTRAINT**
- 10- **CHECK**
- 11- Collation
- 12- **databases**
- 13- **EXISTS**
- 14- **UPDATE**
- 15- **DELETE**

تطبيق:

1- من خلال SQL Server Management Studio حدد قاعدة البيانات UtilityCompany1

2- افتح الجدول Employees وإدخال بعض السجلات:

EmployeeNumber	FirstName	LastName	Title
	Robert	Anson	
	Justine	Keys	
	Edward	Kirkland	
	Kimberly	Eisner	
	Jonathan	Adamson	
	Steve	Fox	
	Andrew	Boroughs	

3- افتح الجدول Customers وأنشئ عددا من السجلات

AccountNumber	DateAccountCreated	CustomerName	Address	City	State	EmailAddress

4- قم بإجراء العمليات التي رأيناها في هذا الدرس.

تحليل البيانات

مبادئ تحليل البيانات:

بعد إنشاء الجدول وتعبئته بالسجلات، قد ترغب بالقيام ببعض العمليات على البيانات المخزنة فيه. من أكثر العمليات التي يؤديها المستخدمون في قاعدة بيانات، البحث عن البيانات أو عزل بعضها أو التي تستجيب لمعيار معين. تدعى عملية البحث عن البيانات التي تطابق معيار معين بالاستعلام عن البيانات.

بصفتك مبرمج قواعد البيانات، فإنك تقوم بإرسال الاستعلامات عن طريق تمرير تعليمات إلى محرك قاعدة البيانات الذي يتولى تنفيذها. ويتم ذلك باستخدام لغة SQL.

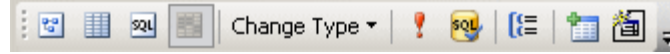
في Microsoft SQL Server يمكن تحليل البيانات باستخدام SQL Server Management Studio من خلال نافذة الاستعلام.

استعراض البيانات في جدول:

لاستعراض البيانات، من خلال متصفح الكائنات يمكنك النقر بالزر الأيمن على الجدول واختيار Open Table، لتواجهك نافذة مثل هذه:

Video ID	Video Title	Director	© Year	Length	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	1 hr. 18 Mns	
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Ethan Coen	2004	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	PG-13
6	Red Heat	Walter Hill	NULL	NULL	NULL
7	Mystic River	NULL	2003	2 hrs. 17 min	NULL
*	NULL	NULL	NULL	NULL	NULL

عند فتح الجدول يظهر (افتراضيا) شريط الأدوات مصمم الاستعلامات (Query Designer) كما يأتي:



عندما يتم فتح الجدول لاستعراض بياناته، يمكنك النقر على Query Designer من القائمة الرئيسية، أو بالزر الأيمن على أي مكان فوق الجدول، لتحصل على:

- في القائمة التي تظهر، حدد Pane واختر على Diagram
- مرة أخرى، افتح القائمة Pane وانقر على Criteria
- ثم افتح مرة أخرى القائمة Pane واختر هذه المرة SQL

Table - dbo.Videos* Summary

Videos




- * (All Columns)
- Video ID
- Video Title
- Director
- © Year
- Length
- Rating

Column	Alias	Table	Output	Sort Type	Sort Order
*			<input checked="" type="checkbox"/>		
			<input checked="" type="checkbox"/>		
			<input checked="" type="checkbox"/>		

```
SELECT *
FROM Videos
```

	Video ID	Video Title	Director	© Year	Length	Rating
▶	1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
	2	The Silence of the Lambs	Jonathan Demme	1991	1 hr. 18 Mns	
	3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
	4	The Lady Killers	Joel Coen & Ethan Coen	2004	104 Minutes	R
	5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	PG-13
	6	Red Heat	Walter Hill	NULL	NULL	NULL
	7	Mystic River	NULL	2003	2 hrs. 17 min	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

1 of 7 | Cell is Read Only.

بطريقة أخرى من شريط Query Designer، انقر على الزر  Show Diagram Pane، والزر  Show Criteria Pane، وكذلك  Show SQL Pane.

سينقسم إطار الجدول إلى أربعة أقسام هي:

- **الرسم البياني Diagram:** يشير القسم العلوي إلى إطار المخطط، وهو يعرض الجدول (أو الجداول) الذي يحتوي على الحقول التي تريد الاستعلام عنها. يعرض كل حقل على شكل مربع اختيار على اليسار واسم الحقل على اليمين، في البند الأول من القائمة تجد علامة الرمز * وعبارة All Columns على اليمين.
- **المعايير Criteria:** تحت الإطار الأول تجد القسم الثاني الذي يحتوي على المعايير، وهو يعرض قائمة الحقول المستخدمة لإنشاء الاستعلام.

- الاستعلام SQL: ويعرض الجزء الثالث من الواجهة الاستعلام الناتج عن الرسم البياني أو المعايير المحددة.
- النتائج Results: وأخيرا القسم الأسفل يعرض النتيجة التي تظهر عند تنفيذ الاستعلام.

إذا كنت أردت إخفاء أو إظهار قسم معين، يمكنك النقر بالزر الأيمن فوق أي مكان من الجدول ثم حدد Pane وانقر على اسم القسم:

The screenshot shows the Microsoft Access interface with a query named 'Videos' from the 'dbo.Videos*' database. The query is displayed in a grid view. A context menu is open over the table, with the 'Pane' option selected. The menu includes options like 'Execute SQL', 'Add Group By', 'Add Table...', 'Add New Derived Table', 'Clear Results', 'Diagram', 'Criteria', 'SQL', and 'Results'. The table data is visible below the menu.

Column	Alias	Table	Output	Sort Type
*			<input checked="" type="checkbox"/>	
			<input checked="" type="checkbox"/>	
			<input checked="" type="checkbox"/>	

```

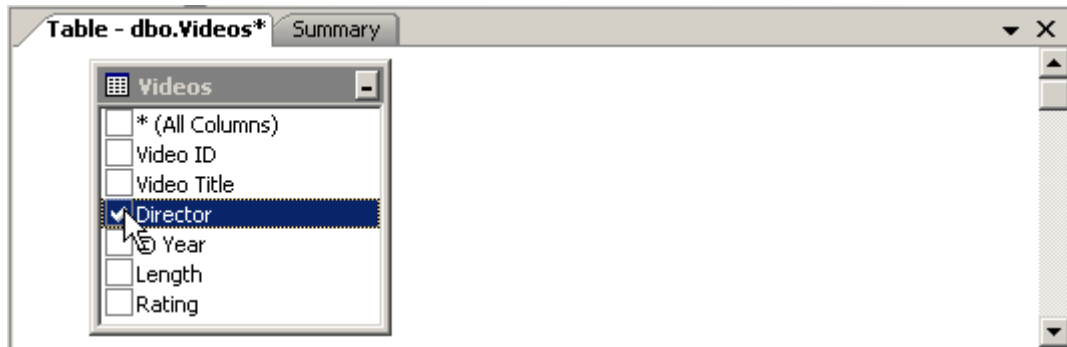
SELECT *
FROM Videos

```

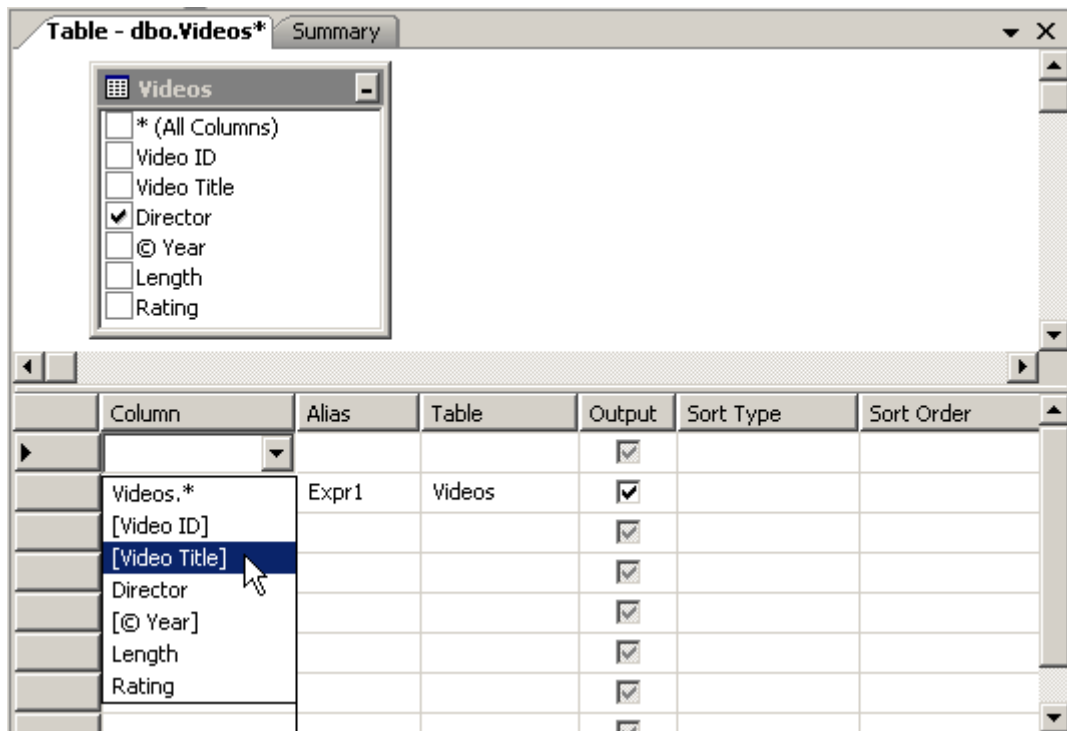
Video ID	Video Title	Director	© Year	Length	Rating
1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	The Silence of the Lambs	Jonathan Demme	1991	1 hr. 18 Mns	
3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	The Lady Killers	Joel Coen & Ethan Coen	2004	104 Minutes	R
5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	PG-13
6	Red Heat	Walter Hill	NULL	NULL	NULL
7	Mystic River	NULL	2003	2 hrs. 17 min	NULL
*	NULL	NULL	NULL	NULL	NULL

تحديد الحقول:

يتم إنشاء الاستعلام تلقائياً بناءً على اختيار الحقول التي تريد عرضها، لتحديد أحد الحقول في قسم الرسم التخطيطي لواجهة التصميم يمكنك النقر على مربع الاختيار:



بعد تحديد الحقل يصبح هذا الحقل محددًا في قسم المعايير أيضًا، ويظهر اسمه في القسم الاستعلام، ويمكنك أيضًا تحديد حقل بالنقر على المربع تحت العمود Column في قسم المعايير.




بعد تحديد أحد الحقول في قسم تحديد المعايير يضاف تلقائياً في قسم المخطط ويضاف أيضاً في قسم الاستعلام.

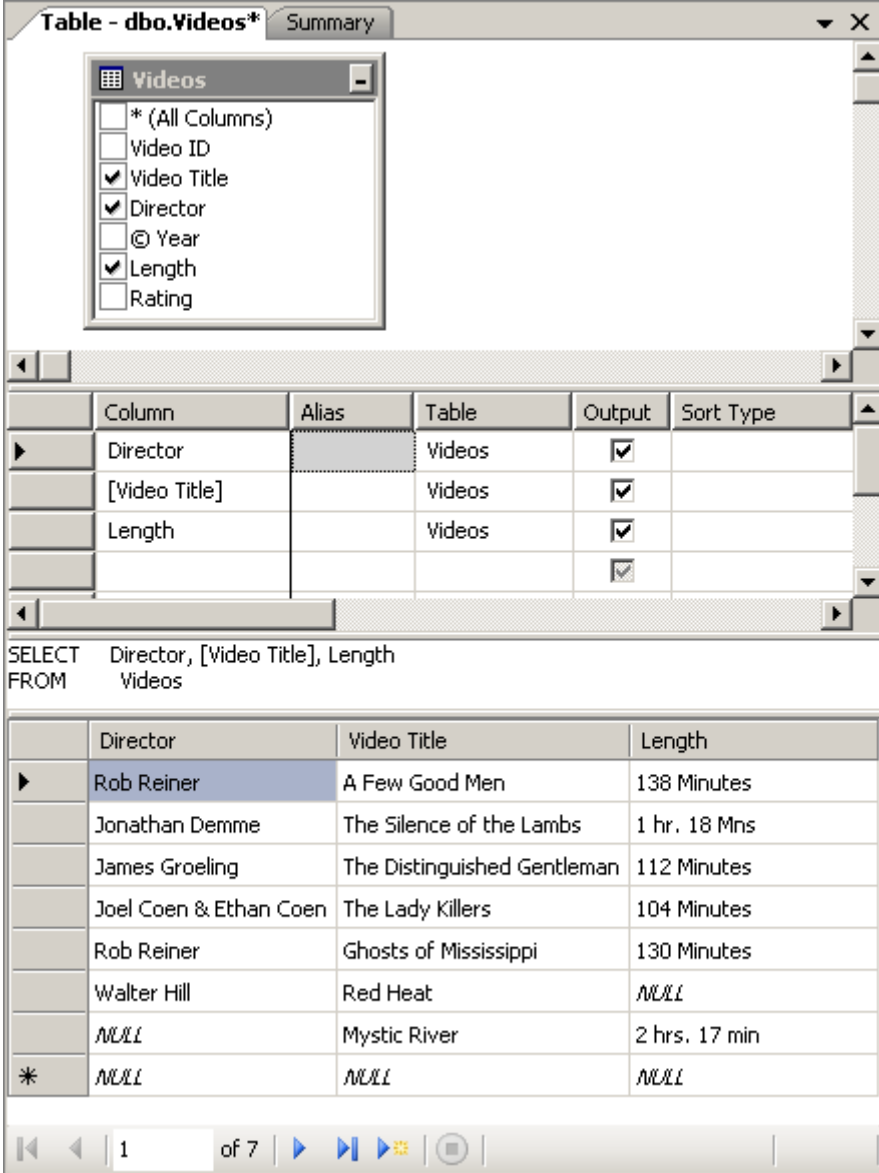
إذا كنت تعرف اسم الحقل الذي تريد إضافته حيث يظهر في المخطط، يمكنك أن تضيف اسمه مباشرة في الاستعلام.

يمكنك بأي من الأساليب الثلاثة المذكورة أن تختار الحقول (أو الحقول) لإنشاء الاستعلام.

تنفيذ الاستعلام:

بعد إنشاء الاستعلام SQL يمكنك عرض نتائجه من خلال تنفيذه. للقيام بذلك يمكنك النقر بالزر الأيمن على أي مكان في إطار الجدول واختار Execute SQL، أو من شريط الأدوات على الزر  Execute.

بعد تنفيذ الاستعلام، يظهر في القسم السفلي الحقول المحددة من الجدول. مثال:



The screenshot shows a window titled "Table - dbo.Videos*" with a "Summary" tab. A "Videos" table is selected, and the following columns are checked: Video Title, Director, Length, and Rating. Below the table selection, a table shows the columns selected for the query: Director, [Video Title], Length, and an empty column. The SQL query is displayed as:

```
SELECT Director, [Video Title], Length
FROM Videos
```

The query results are shown in a table with the following data:

Director	Video Title	Length
Rob Reiner	A Few Good Men	138 Minutes
Jonathan Demme	The Silence of the Lambs	1 hr. 18 Mns
James Groeling	The Distinguished Gentleman	112 Minutes
Joel Coen & Ethan Coen	The Lady Killers	104 Minutes
Rob Reiner	Ghosts of Mississippi	130 Minutes
Walter Hill	Red Heat	NULL
NULL	Mystic River	2 hrs. 17 min
*	NULL	NULL

The bottom of the window shows a pagination bar with "1 of 7" and navigation icons.

محلل الاستعلامات (Transact-SQL) وتحليل البيانات:

يتم تحليل البيانات باستخدام استعلامات SQL وفق معايير محددة، من أجل تحليل البيانات يمكنك استخدام SQL وكتابة استعلام مناسب مع تحديد قاعدة البيانات والجدول الذي تريد تحليل بياناته.

تطبيق عملي: التمهيد لتحليل البيانات

1- قم بتشغيل SQL Server Management Studio واتصل بالخادم

2- من القائمة الرئيسية انقر على File -> New -> Query With Current Connection

3- لإعداد قاعدة بيانات التطبيق، اكتب ما يلي:

```
-- =====
-- Author:    FunctionX
-- Database:  RealEstatel
-- =====
IF EXISTS (
    SELECT *
        FROM sys.databases
        WHERE name = N'RealEstatel'
)
    DROP DATABASE RealEstatel
GO
CREATE DATABASE RealEstatel;
GO

-- =====
-- Author:    FunctionX
-- Database:  RealEstatel
-- Table:    Properties
-- =====
USE RealEstatel;
GO
CREATE TABLE Properties
(
    PropertyID int identity(1,1) NOT NULL,
    PropertyNumber char(6),
    Address varchar(100),
    City varchar(50),
    State char(2),
    ZIPCode varchar(12),
    PropertyType varchar(40),
    Condition varchar(32),
    Bedrooms smallint,
    Bathrooms float,
    FinishedBasement bit,
    IndoorGarage bit,
    Stories smallint,
    YearBuilt smallint,
    MarketValue money
);
```



```

GO

INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('524880', '1640 Lombardo Ave', 'Silver Spring', 'MD',
    '20904', 'Single Family', 'Good', 4, 2.5, 3, 1, 3, 1995,
    495880.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('688364', '10315 North Hacht Rd', 'College Park', 'MD',
    '20747', 'Single Family', 'Excellent', 4, 3.5, 3,
    1, 2, 2000, 620724.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, FinishedBasement,
    Stories, MarketValue)
VALUES('611464', '6366 Lolita Drive', 'Laurel', 'MD',
    '20707', 'Single Family', 'Good', 1, 2, 422625.00);
GO
INSERT INTO Properties(Address, City, PropertyType,
    Bedrooms, MarketValue)
VALUES('9002 Palasko Hwy', 'Tysons Corner',
    'Condominium', 2, 422895.00);
GO
INSERT INTO Properties(PropertyNumber, State,
    ZIPCode, Bedrooms, YearBuilt, MarketValue)
VALUES('420115', 'DC',
    '20011', 2, 1982, 312555);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
    PropertyType, Bedrooms, YearBuilt, MarketValue)
VALUES('917203', 'Alexandria', '22024',
    'Single Family', 3, 1965, 345660.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    PropertyType, Condition, Bedrooms, Bathrooms, MarketValue)
VALUES('200417', '4140 Holisto Crt', 'Germantown', 'MD',
    'Condominium', 'Excellent', 2, 1, 215495.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('927474', '9522 Lockwood Rd', 'Chevy Chase', 'MD',
    '20852', 'Townhouse', 'Bad Shape', 3, 2.5, 3, 0, 3,
    1992, 415665.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('207850', '14250 Parkdoll Rd', 'Rockville', 'MD',
    '20854', 'Townhouse', 'Good', 3, 2.5, 2, 1, 2,
    1988, 325995.00);
GO
INSERT INTO Properties(City, PropertyType, Bedrooms,
    YearBuilt, MarketValue)
VALUES('Washington', 'Townhouse', 4, 1975, 366775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,

```

```

        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        YearBuilt, MarketValue)
VALUES('288540', '10340 Helmes Street #408', 'Silver Spring', 'MD',
        '20906', 'Condominium', 'Good', 1, 1, 2000, 242775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('247472', '1008 Coppen Street', 'Silver Spring', 'MD',
        '20906', 'Single Family', 'Excellent',
        3, 3, 3, 1, 3, 1996, 625450.00);
GO
INSERT INTO Properties(City, ZIPCode, PropertyType,
        Stories, YearBuilt, MarketValue)
VALUES('Chevy Chase', '20956', 'Single Family',
        3, 2001, 525450.00);
GO
INSERT INTO Properties(Address, City, State,
        PropertyType, Condition, Bedrooms, MarketValue)
VALUES('686 Herod Ave #D04', 'Takoma Park', 'MD',
        'Condominium', 'Excellent', 2, 360885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297446', '14005 Sniders Blvd', 'Laurel', 'MD',
        '20707', 'Townhouse', 'Needs Repair',
        4, 1.5, 3, 1, 2, 2002, 412885.00);
GO
INSERT INTO Properties(City, ZIPCode, Condition, Bedrooms,
        Stories, YearBuilt)
VALUES('Silver Spring', '20905', 'Good',
        4, 2, 1965);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('924792', '680 Prushia Rd', 'Washington', 'DC',
        '20008', 'Single Family', 'Good',
        5, 3.5, 3, 0, 3, 2000, 555885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('294796', '14688 Parrison Street', 'College Park', 'MD',
        '20742', 'Single Family', 'Excellent',
        5, 2.5, 2, 1, 2, 1995, 485995.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        YearBuilt, MarketValue)
VALUES('811155', '10340 Helmes Street #1012', 'Silver Spring',
        'MD', '20906', 'Condominium', 'Good',
        1, 1, 2000, 252775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
        FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('447597', '4201 Vilamar Ave', 'Hyattsville', 'MD',
        '20782', 'Townhouse', 'Excellent',
        3, 2, 2, 1, 3, 1992, 365880.00);

```

```

GO
INSERT INTO Properties(Address, ZIPCode, Bathrooms)
VALUES('1622 Rombard Str', 20904, 2.5);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297415', '980 Phorwick Street', 'Washington', 'DC',
    '20004', 'Single Family', 'Good',
    4, 3.5, 3, 3, 1, 2004, 735475.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('475974', '9015 Marvin Crow Ave', 'Gaithersburg', 'MD',
    '20872', 'Single Family', 'Needs Repair',
    4, 2.5, 3, 1, 1, 1965, 615775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('836642', '3016 Feldman Court', 'Rockville', 'MD',
    '20954', 'Single Family', 'Bad Shape',
    5, 3, 3, 1, 3, 1960, 528555.00);
GO
INSERT INTO Properties(City, State, PropertyType, Stories)
VALUES('Rockville', 'MD',
    'Townhouse', 1);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('208304', '7307 Everett Hwy', 'Washington', 'DC',
    '20012', 'Townhouse', 'Excellent',
    2, 2.5, 2, 0, 4, 2006, 420550.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms,
    Bathrooms, YearBuilt, MarketValue)
VALUES('644114', '10340 Helmes Street#1006', 'Silver Spring',
    'MD', '20906', 'Condominium', 'Good',
    2, 2, 2000, 258445.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('937966', '7303 Warfield Court', 'Tysons Corner', 'VA',
    '22131', 'Single Family', 'Good',
    3, 2.5, 3, 1, 4, 2006, 825775.00);
GO
INSERT INTO Properties(City, ZIPCode, Condition, Bedrooms,
    Stories, YearBuilt)
VALUES('Fairfax', '22232', 'Good', 3, 3, 1985);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297497', '12401 Conniard Ave', 'Takoma Park', 'MD',
    '20910', 'Townhouse', 'Good',
    3, 2.5, 3, 1, 3, 2004, 280775.00);
GO

```

```

INSERT INTO Properties(City, ZIPCode, PropertyType,
    Bedrooms, Bathrooms, MarketValue)
VALUES('Alexandria', '22035', 'Condominium',
    2, 2, 425775.00);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
    PropertyType, Condition, Bedrooms, Bathrooms,
    YearBuilt, Stories, MarketValue)
VALUES('855255', 'Laurel', '20707', 'Single Family',
    'Needs Repair', 3, 2, 1962, 2, 342805.00);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode, PropertyType,
    Condition, Bedrooms, Bathrooms, MarketValue)
VALUES('225227', 'Rockville', '20857', 'Condominium', 'Good',
    1, 1, 525885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('469750', '6124 Falk Rd', 'Arlington', 'VA',
    '22031', 'Single Family', 'Needs Repair',
    4, 3.5, 3, 1, 1, 1982, 635995.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('826927', '5121 Riehl Ace', 'Fairfax', 'VA',
    '22232', 'Townhouse', 'Excellent',
    3, 1.5, 2, 0, 1, 2002, 325620.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('287064 ', '9533 Pensulian Rd', 'Silver Spring', 'MD',
    '20904', 'Single Family', 'Bad Shape',
    3, 1.5, 3, 1, 2, 1992, 485775.00);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
    PropertyType, Condition, Bedrooms, YearBuilt, Stories)
VALUES('724001 ', '705 Helios Ave', '20004',
    'Townhouse', 'Bad Shape', 3, 1974, 4);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('209275', '944 Fryer Ave', 'Chevy Chase', 'MD',
    '20852', 'Single Family', 'Excellent',
    5, 2.5, 3, 0, 2, 2002, 625665.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('204759', '1950 Galego Street', 'Germantown', 'MD',
    '20874', 'Single Family', 'Excellent',
    4, 3.5, 2, 1, 4, 2007, 428665.00);
GO
INSERT INTO Properties(City, State, PropertyType,
    Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES('Takoma Park', 'MD',
    'Conbominium', 2, 2, 2000, 225885.00);
GO

```

```

INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('937259', '12366 Fowler Ave', 'Alexandria', 'VA',
    '22031', 'Townhouse', 'Good',
    3, 1.5, 3, 1, 3, 2007, 402815.00);
GO


```

4- اضغط F5 لتنفيذ الاستعلام

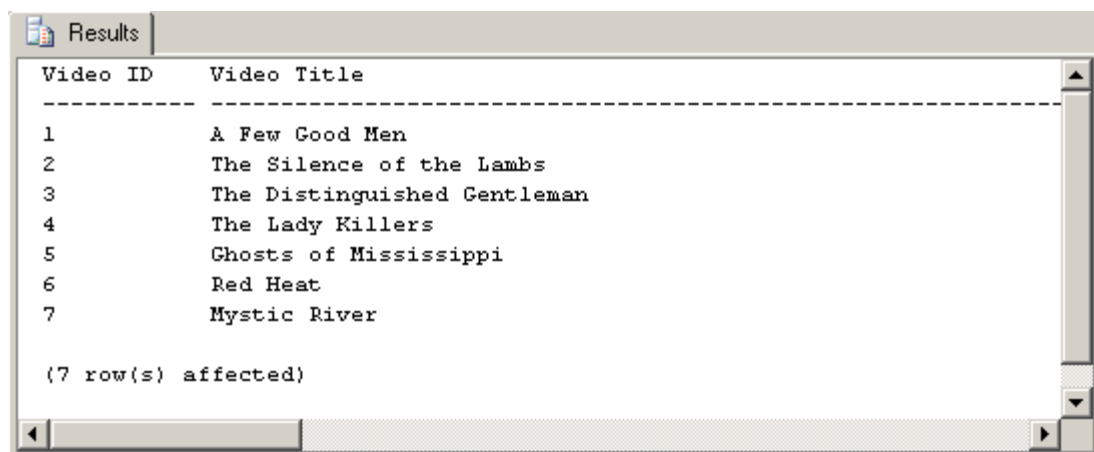
5- قم بحفظه في ملف باسم RealEstate1 وأغلق نافذة الاستعلام.

استعراض نتائج SQL وتحليل البيانات:

في SQL Server Management Studio وأثناء تصميم الاستعلام يعرض القسم السفلي نتيجة الاستعلام المنفذ، وهناك خيارين يمكنك عرض النتيجة بهما، يمكنك إيجادهما في شريط الأدوات SQL Editor، لعرض شريط الأدوات SQL Editor، اختر من القائمة الرئيسية View SQL Editor -> Toolbars ->، أو انقر بالزر الأيمن فوق أي شريط أدوات واختر SQL Editor.

لتحديد طريقة عرض نتائج الاستعلام، لديك خياران: لعرض النتيجة في شكل نصي، يمكنك من شريط الأدوات SQL Editor النقر على زر Result To Text ، أو النقر بزر الأيمن على أي مكان في الجدول واختر Results To Text.

ستظهر النتائج في عمودين من النصوص، مثال:



Video ID	Video Title
1	A Few Good Men
2	The Silence of the Lambs
3	The Distinguished Gentleman
4	The Lady Killers
5	Ghosts of Mississippi
6	Red Heat
7	Mystic River

(7 row(s) affected)

لتظهر النتيجة على شكل جدول انقر من شريط الأدوات SQL Editor على الزر Result
 To Grid ، بطريقة أخرى يمكنك النقر على الجدول ثم اختر Results To Grid وانقر على زر
 .Results To Grid

ستظهر النتائج في جدول ذو أعمدة مختلفة:

	Video ID	Video Title	Director	© Y...	Length	Rating
1	1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
2	2	The Silence of the Lambs	Jonathan Demme	1991	1 hr. 18 Mns	
3	3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
4	4	The Lady Killers	Joel Coen & Ethan Coen	2004	104 Minutes	R
5	5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	PG-13
6	6	Red Heat	Walter Hill	NULL	NULL	NULL
7	7	Mystic River	NULL	2003	2 hrs. 17 min	NULL

تطبيق عملي: إنشاء استعلام

- 1- في متصفح الكائنات انقر بالزر الأيمن على Databases ثم انقر على Refresh
- 2- قم بتوسيع البند Databases ثم قاعدة البيانات RealEstate1
- 3- انقر بالزر الأيمن على RealEstate1 واختر New Query

تحديد الحقول:

الكلمة الأساسية المستخدمة في SQL لجلب الحقول هي **SELECT**، وهنا يجب تحديد
 الحقول لإجراء تحليل البيانات، تستخدم **SELECT** وفق التالي:

SELECT What FROM WhatObject;

لتحديد كل حقول الجدول يمكن استخدام الرمز * و/أو مجموعة من القيم، مثلاً لعرض
 كل السجلات من الجدول Students، يمكنك كتابة:

SELECT * FROM Students;

بعد كتابة الاستعلام يمكنك تنفيذه لمشاهدة النتيجة، هذا مثال:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```
SELECT * FROM Students;
GO
```

	StudentID	FirstName	LastName	DateOfBirth	Gender	Address
1	1	Sebastien	Porter	1995-02-12 00:00:00.000	Male	4442 Shane A
2	2	Suzie	Hoak	1991-10-05 00:00:00.000	Female	1866 Old Conf
3	3	Antoinette	Clarck	1996-04-10 00:00:00.000	Female	282 Centerway
4	4	Koko	Domba	1990-02-05 00:00:00.000	Male	8782 South W
5	5	Janet	West	1996-06-02 00:00:00.000	Female	980 Washingtc
6	6	Catherine	Chang	1997-06-12 00:00:00.000	Female	2186 Holm St.
7	7	Nehemiah	Dean	1995-12-02 00:00:00.000	Male	442 Lakefront.
8	8	Sherryl	Ashburn	1995-10-07 00:00:00.000	Female	1771 N. Freder
9	9	Santos	Pacheco	1998-05-05 00:00:00.000	Male	14277 Grand M
10	10	Mohamed	Husseini	1998-01-05 00:00:00.000	Male	888 Democrac
11	11	Dean	Chen	1997-05-02 00:00:00.000	Male	4608 St. Leon
12	12	Ruby	DeGaram	1998-10-11 00:00:00.000	Female	8782 South W
13	13	Carole	Chance	1990-10-22 00:00:00.000	Female	445 Farmers A
14	14	Justin	Vittas	1991-05-04 00:00:00.000	Male	12881 Northpc
15	15	Ismael	Zara	1990-05-25 00:00:00.000	Male	9009 Kentland
16	16	Anselme	Waters	1990-07-23 00:00:00.000	Male	1872 Hunters l
17	17	Brenda	Lobo	1989-10-05 00:00:00.000	Female	2490 Chanting
18	18	Suzanna	Verde	1988-03-12 00:00:00.000	Female	3844 John Mcl
19	19	Danilo	Chico	1997-06-20 00:00:00.000	Male	291 Nixon Av.
20	20	Mincy	Franse	1991-10-08 00:00:00.000	Female	1788 Old Mont

Query... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

يمكنك أيضا إدراج الرمز * مسبقا باسم الجدول ثم رمز النقطة (.)، مثال:

```
SELECT Students.* FROM Students;
```

أو باستخدام اسم مستعار كما رأينا سابقا، حيث يمكن كتابة العبارة أعلاه على النحو التالي:

```
SELECT std.* FROM Students std;
```

عوض جلب جميع البيانات، يمكنك أيضا تحديد حقل (أو أكثر) من البيانات التي يحتويها

الجدول. مثال للحصول على قائمة بأسماء الطلاب نستخدم الآتي:

```
SELECT Students.LastName FROM Students;
```

لإعداد قائمة بعدة حقول، حدد أسماء الحقول بعد SELECT مع الفصل بينها بفاصلة

باستثناء اسم الحقل الأخير وفق الصيغة:

```
SELECT Column1, Column2, Column_n FROM WhatObject;
```

مثال لعرض قائمة تتضمن عدة حقول من الجدول Students:

```
SELECT FirstName, LastName, Gender, EmailAddress, HomePhone
FROM Students;
```

يمكن تحديد كتابة اسم الجدول قبل اسم كل حقل، وليس ضروريا تحديد اسم الجدول لكافة الحقول، بل يمكن تحديد البعض دون البعض الآخر، حيث الاستعلام أعلاه يعادل:

```
SELECT Students.FirstName,
        LastName,
        Students.Gender,
        EmailAddress,
        HomePhone
FROM Students;
```

ملاحظة: وهذا ما ينبغي كتابته عند تحديد عدة حقول لها نفس الاسم من جداول مختلفة، وإلا فسيقع SQL Server في التباس، مثال:

```
SELECT Students.FirstName,
        Teachers. FirstName,
        Students. LastName,
        Teachers. LastName,
        EmailAddress,
        HomePhone
FROM Students, Teachers;
```

تذكر أنه يمكنك استخدام اسم مستعار للجدول قبل كل عمود بحرف أو كلمة ثم نقطة، مثال:

```
SELECT std.FirstName, std.LastName, std.Gender, std.EmailAddress,
std.HomePhone
FROM Students std;
```

تطبيق عملي: تحديد البيانات

1- أكتب في نافذة الاستعلام:

```
USE RealEstate1;
GO
SELECT p.* FROM Properties p;
GO
```

2- اضغط F5 لتنفيذ العبارة

3- للحصول على قائمة للحقول PropertyNumber و PropertyType، أكتب الاستعلام التالي:

```
SELECT dbo.Properties.PropertyNumber,
        dbo.Properties.PropertyType
FROM Properties;
GO
```

4- قم بتحديد الاستعلام السابق واضغط F5 لتنفيذه

5- للحصول على قائمة بأنواع العقارات من الجدول Properties، غير الاستعلام على النحو التالي:

```
SELECT dbo.Properties.PropertyNumber ,
       dbo.Properties.PropertyType
FROM Properties;
GO
```

6- انقر بالزر الأيمن على أي مكان في الجدول ثم حدد Results To واختر Results To Text ثم F5 لتنفيذ الاستعلام.

استخدام اسم مستعار لتحديد الحقول:

بعد تحديد الحقول وتنفيذ الاستعلام يظهر اسم كل حقل في عمود، وهنا يمكنك عرض الحقول في أعمدة بأسماء من اختيارك.

لتحديد اسم مغاير لاسم الحقل في واجهة تحليل بيانات الجدول التي استعرضناها سابقاً، اكتب الاسم المستعار في العمود Alias لكل حقل أو لبعض الحقول، مثال :

Table - dbo.Students* Summary

Students

- StudentNbr
- FirstName
- MI
- LastName
- DateOfBirth

Column	Alias	Table	Output	Sort Type
LastName	[Last Name]	Students	<input checked="" type="checkbox"/>	
FirstName	[First Name]	Students	<input checked="" type="checkbox"/>	
DateOfBirth	[Date of Birth]	Students	<input checked="" type="checkbox"/>	
SPHome	[Lives in a Single-Parent Home]	Students	<input checked="" type="checkbox"/>	

```

SELECT LastName AS [Last Name],
       FirstName AS [First Name],
       DateOfBirth AS [Date of Birth],
       SPHome AS [Lives in a Single-Parent Home]
FROM Students

```

Last Name	First Name	Date of Birth	Lives in a Single-Parent Home
Porter	Sebastien	2/12/1985 0:00:00	False
Hoak	Suzie	10/5/1991 0:00:00	False
Clarck	Antoinette	4/10/1986 0:00:00	False
Domba	Koko	2/5/1990 0:00:00	False
West	Janet	6/2/1986 0:00:00	True
Dean	Nehemiah	12/2/1985 0:00:00	True
Ashburn	Sherryl	10/7/1985 0:00:00	False
Pacheco	Santos	5/5/1988 0:00:00	False
Chen	Dean	5/2/1987 0:00:00	True
DeGaram	Ruby	10/11/1988 0:0...	False

1 of 160

باستخدام قسم كتابة استعلام، أضف على الجانب الأيمن من اسم كل حقل اسم مستعاراً مسبوفاً بالكلمة AS، إذا كان الاسم المستعار مكون من كلمة وحيدة يمكنك كتابته مباشرة، مثال:

```

SELECT FirstName,
       LastName,
       HomePhone AS PhoneNumber,
       ParentsNames AS NamesOfParents
FROM Students;
GO

```

إذا كان الاسم المستعار مكون من عدة كلمات فيجب إدراجه بين علامتي اقتباس ("") أو بين معقوفين (()). مثال:

```

SELECT FirstName AS [First Name],
       LastName AS [Last Name],
       HomePhone AS [Phone Number],
       ParentsNames AS [Names of Parents]
FROM   Students;
GO

```

يمكن كتابة الاستعلام السابق على النحو التالي:

```

SELECT Students.FirstName AS [First Name],
       Students.LastName AS [Last Name],
       Students.HomePhone AS [Phone Number],
       Students.ParentsNames AS [Names of Parents]
FROM   Students;
GO

```

وأيضاً:

```

SELECT dbo.Students.FirstName AS [First Name],
       dbo.Students.LastName AS [Last Name],
       dbo.Students.HomePhone AS [Phone Number],
       dbo.Students.ParentsNames AS [Names of Parents]
FROM   Students;
GO

```

وكذلك:

```

SELECT std.FirstName AS [First Name],
       std.LastName AS [Last Name],
       std.HomePhone AS [Phone Number],
       std.ParentsNames AS [Names of Parents]
FROM   Students std;
GO

```

تطبيق عملي: استخدام أسماء مستعارة

1- لاستخدام أسماء مستعارة لعرض الحقول السابقة، غير الاستعلام بالشكل التالي:

```

SELECT house.PropertyNumber AS [Prop #],
       house.ZIPCode AS [Location],
       house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.Bedrooms AS [Beds],
       house.Bathrooms AS [Baths],
       house.MarketValue AS [Value]
FROM   Properties house;
GO

```

2- بالزر الأيمن انقر في أي مكان من الجدول ثم حدد Results To واختر

Grid

3- اضغط F5 للتنفيذ.

استخدام عبارة لعرض الحقول:

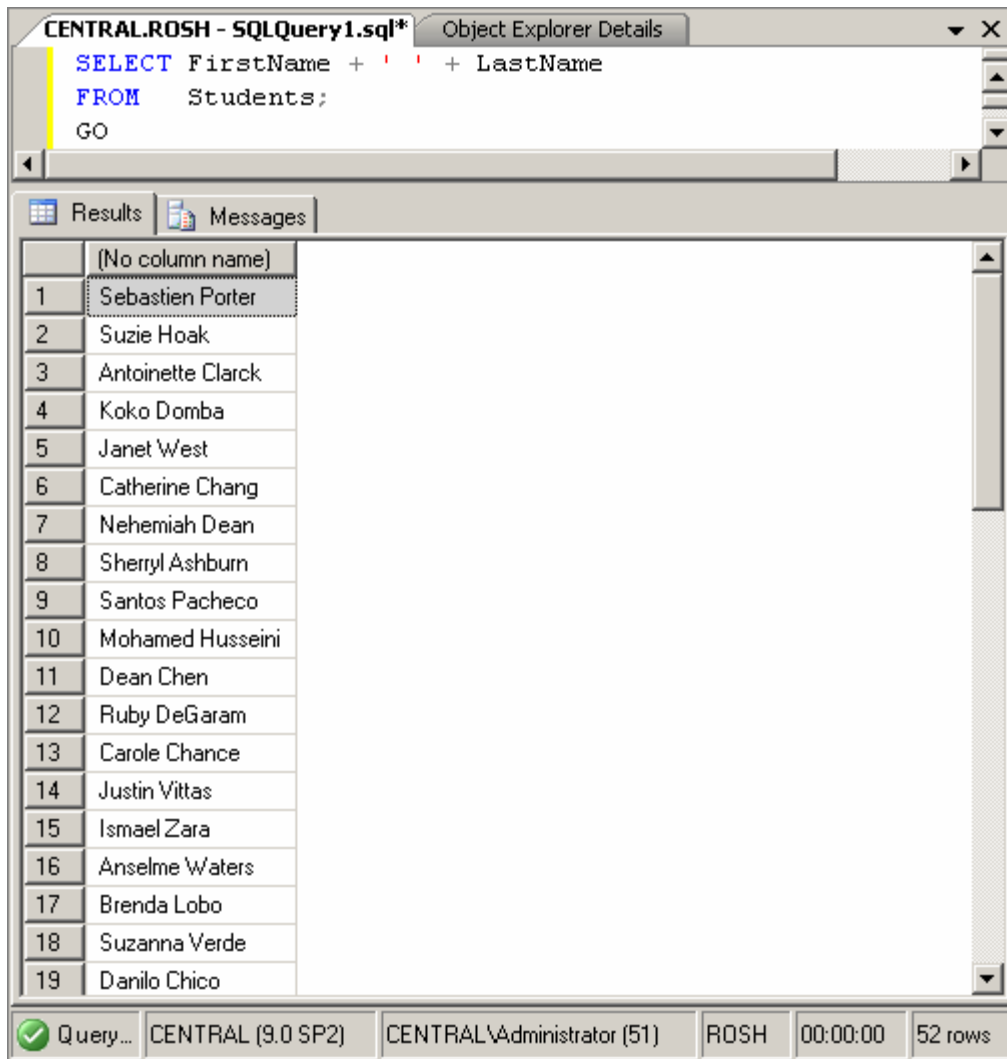
باستخدام الكلمة **SELECT** التي رأيناها سابقا يمكن إنشاء قائمة من الحقول منفردة، حيث تظهر الأعمدة منفصلة عن بعضها البعض، يمكنك الجمع بين الأعمدة لإنشاء سلسلة من القيم، أو بتعبير آخر عبارة من الأعمدة.

على سبيل المثال، يمكنك الجمع بين الحقلين **First Name** و **Last Name** لإنتاج حقل يعرض الاسم الكامل **Full Name** كتعبير من الأعمدة، ويمكن استخدام تعابير متنوعة من عرض بيانات مفصلة عن الحقول، مثلا يمكنك أن تستخدم إجراء عملية حسابية على حقلين أو أكثر، مثل عرض عمود يحسب جداء حقل راتب الساعة الواحدة أسبوعيا مضروبا في حقل عدد ساعات العمل الأسبوعية، للحصول على الراتب الأسبوعي.

الطريقة الأكثر شيوعا لإنشاء حقل عمود مركب من عدة حقول، هي باستخدام الجمع بين بيانات الحقول للحصول على حقل واحد جديد، مثال:

```
SELECT FirstName + ' ' + LastName  
FROM Students;  
GO
```

وهذا الاستعلام من شأنه أن ينتج:



بالإضافة إلى ذلك يمكن أيضا استخدام قيم عديدة حيث يمكن القيام بجميع العمليات الحسابية الأخرى، كمثال يمكنك حساب جداء ساعات العمل الأسبوعية إلى راتب كل ساعة للحصول على الراتب الأسبوعي على النحو التالي:

```
SELECT WeeklyHours * HourlySalary
FROM Payroll
```

ملاحظة: عند إنشاء عبارة مركبة من عدة حقول، يظهر الحقل الناتج دون أي عنوان، ويشير إليه SQL Server Management Studio بالعبارة: (No column name)

يمكنك لذلك إنشاء اسم مستعار للتعبير عن الحقل الناتج من تركيب عدة حقول لعرضه باسم مناسب، للقيام بذلك أضف على الجانب الأيمن من عبارة الحقول كلمة AS متبوعة بالاسم المستعار المناسب، مثال:

```
SELECT FirstName + ' ' + LastName AS 'Full Name',
```

```

SELECT FirstName + ' ' + LastName AS [Full Name],
       EmrgName + ' ' + EmrgPhone AS [Emergency Contact]
FROM Students;
GO

```

نتائج الاستعلام أعلاه:

	Full Name	Emergency Contact
1	Sebastien Porter	John Porter (202) 662-4825
2	Suzie Hoak	Christianne Hoak (301) 364-0221
3	Antoinette Clarck	Henriette Clarck (301) 598-9292
4	Koko Domba	Jeremiah Domba
5	Janet West	Kirsten Farmer
6	Catherine Chang	Ernie Change (202) 215-6663
7	Nehemiah Dean	Marie Rodnat (703) 434-8756
8	Sherryl Ashburn	Shelia Ashburn (703) 292-8340
9	Santos Pacheco	(202) 938-7772
10	Mohamed Hussein	Dr. Phyllis Hussein (202) 631-8855
11	Dean Chen	Alexandra Chen
12	Ruby DeGaram	Anthony DeGaram (301) 938-1220
13	Carole Chance	Andy Chance (301) 422-1001
14	Justin Vittas	(301) 549-0020
15	Ismael Zara	Robert Zara (202) 978-6642
16	Anselme Waters	Celestin Waters (703) 894-6624
17	Brenda Lobo	(703) 894-2888
18	Suzanna Verde	Thommy Verde

علامة التعيين (=):

عند إنشاء تعبير مركب من عدة حقول باستخدام العوامل الحسابية، يظهر الحقل جديد دون أي اسم، يسمح لك SQL بتحديد اسم مختلف لأي حقل (أو عبارة مركبة) خلال تحليل البيانات، ويتم ذلك باستخدام علامة المساواة "=".

لتغيير اسم العمود خلال تحليل البيانات اكتب الاسم الذي تريده ثم أضف على يمينه علامة المساواة (=) متبوعة بالاسم الفعلي للحقل، مثال:

```

SELECT LastName,
       EmergencyName = EmrgName,

```

```

EmergencyPhone = EmrgPhone
FROM Students;
GO

```

نتائج الاستعلام:

The screenshot shows a SQL query window titled 'CENTRAL.ROSH - SQLQuery1.sql*' with the following query:

```

SELECT LastName,
EmergencyName = EmrgName,
EmergencyPhone = EmrgPhone
FROM Students;
GO

```

The Results tab shows a table with 18 rows and 2 columns: 'Full Name' and 'Emergency Contact'. The status bar at the bottom indicates '52 rows'.

	Full Name	Emergency Contact
1	Sebastien Porter	John Porter (202) 662-4825
2	Suzie Hoak	Christianne Hoak (301) 364-0221
3	Antoinette Clarck	Henriette Clarck (301) 598-9292
4	Koko Domba	Jeremiah Domba
5	Janet West	Kirsten Farmer
6	Catherine Chang	Ernie Change (202) 215-6663
7	Nehemiah Dean	Marie Rodnat (703) 434-8756
8	Sherryl Ashburn	Shelia Ashburn (703) 292-8340
9	Santos Pacheco	(202) 938-7772
10	Mohamed Hussein	Dr. Phyllis Hussein (202) 631-8855
11	Dean Chen	Alexandra Chen
12	Ruby DeGaram	Anthony DeGaram (301) 938-1220
13	Carole Chance	Andy Chance (301) 422-1001
14	Justin Vittas	(301) 549-0020
15	Ismael Zara	Robert Zara (202) 978-6642
16	Anselme Waters	Celestin Waters (703) 894-6624
17	Brenda Lobo	(703) 894-2888
18	Suzanna Verde	Thommy Verde

نفس الطريقة لعرض أكثر من حقل مع الفصل بين الحقول بفواصل. مثال:

```

SELECT LastName,
EmergencyName = EmrgName,
EmergencyPhone = EmrgPhone
FROM Students;
GO

```

ونتيجة الاستعلام:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT LastName,
       EmergencyName = EmrgName,
       EmergencyPhone = EmrgPhone
FROM   Students;
GO

```

Results Messages

	Full Name	Emergency Contact
1	Sebastien Porter	John Porter (202) 662-4825
2	Suzie Hoak	Christianne Hoak (301) 364-0221
3	Antoinette Clarck	Henriette Clarck (301) 598-9292
4	Koko Domba	Jeremiah Domba
5	Janet West	Kirsten Farmer
6	Catherine Chang	Ernie Change (202) 215-6663
7	Nehemiah Dean	Marie Rodnat (703) 434-8756
8	Sherryl Ashburn	Shelia Ashburn (703) 292-8340
9	Santos Pacheco	(202) 938-7772
10	Mohamed Hussein	Dr. Phyllis Hussein (202) 631-8855
11	Dean Chen	Alexandra Chen
12	Ruby DeGaram	Anthony DeGaram (301) 938-1220
13	Carole Chance	Andy Chance (301) 422-1001
14	Justin Vittas	(301) 549-0020
15	Ismael Zara	Robert Zara (202) 978-6642
16	Anselme Waters	Celestin Waters (703) 894-6624
17	Brenda Lobo	(703) 894-2888
18	Suzanna Verde	Thommy Verde

Query... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

يمكنك أيضا إدراج اسم الحقل بين قوسين أو بين معقوفتين. أمثلة:

```

SELECT LastName + ', ' + FirstName AS [Full Name],
       [Emergency Name] = EmrgName,
       'Emergency Phone' = EmrgPhone
FROM   Students;
GO

```

والنتيجة:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT LastName + ', ' + FirstName AS [Full Name],
       [Emergency Name] = EmrgName,
       'Emergency Phone' = EmrgPhone
FROM   Students;
GO

```

Results Messages

	Full Name	Emergency Name	Emergency Phone
1	Porter, Sebastien	John Porter	(202) 662-4825
2	Hoak, Suzie	Christianne Hoak	(301) 364-0221
3	Clarck, Antoinette	Henriette Clarck	(301) 598-9292
4	Domba, Koko	Jeremiah Domba	
5	West, Janet	Kirsten Farmer	
6	Chang, Catherine	Ernie Change	(202) 215-6663
7	Dean, Nehemiah	Marie Rodnat	(703) 434-8756
8	Ashburn, Sherryl	Shelia Ashburn	(703) 292-8340
9	Pacheco, Santos		(202) 938-7772
10	Husseini, Mohamed	Dr. Phyllis Husseini	(202) 631-8855
11	Chen, Dean	Alexandra Chen	
12	DeGaram, Ruby	Anthony DeGaram	(301) 938-1220
13	Chance, Carole	Andy Chance	(301) 422-1001
14	Vittas, Justin		(301) 549-0020
15	Zara, Ismael	Robert Zara	(202) 978-6642
16	Waters, Anselme	Celestin Waters	(703) 894-6624
17	Lobo, Brenda		(703) 894-2888
18	Verde, Suzanna	Thommy Verde	

Query... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

تطبيق عملي: نهاية الدرس

- 1- انطلاقا من قاعدة البيانات RealEstate1 قم بإنشاء استعلام يعرض عبارة مركبة من عدة حقول أخرى، مستخدما في ذلك أسماء مستعارة.
- 2- أغلق واجهة تحليل البيانات ثم أغلق SQL Server Management Studio محتفظا بقاعدة البيانات RealEstate1 للدرس المقبل.

تحليل البيانات (الجزء الثاني)

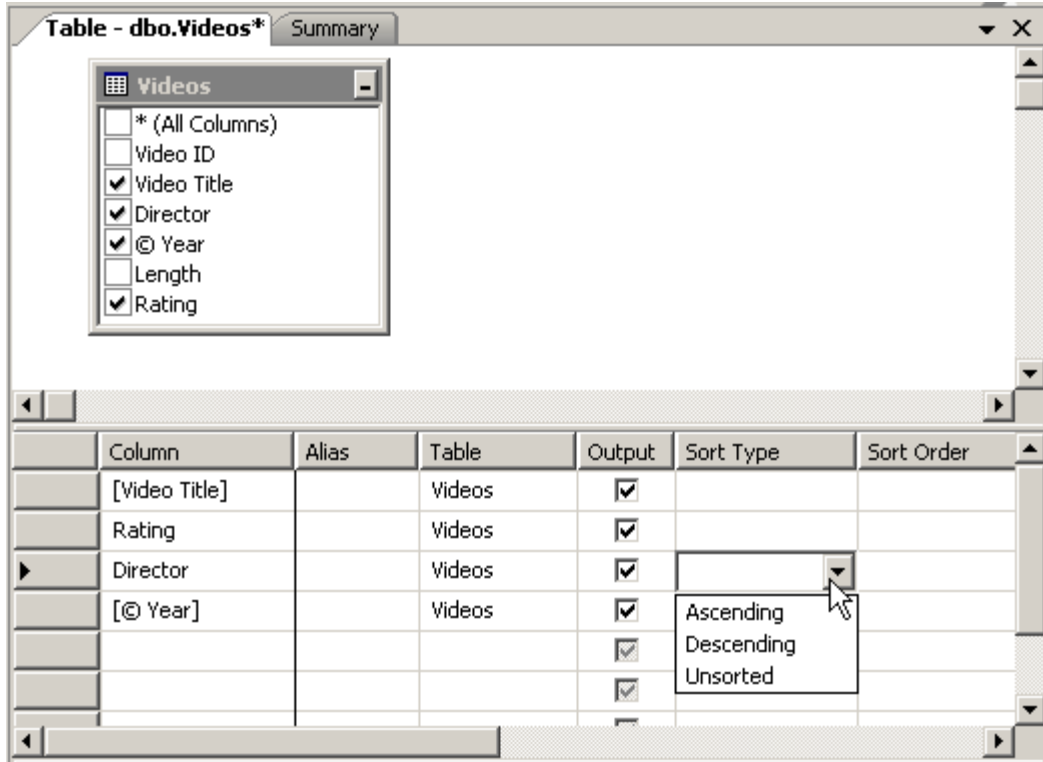
عمليات فرز¹ البيانات:

يتم ترتيب البيانات التي حصلنا عليها باستخدام عبارة SELECT حسب الترتيب الوارد في الجدول، تسمح SQL بترتيب السجلات وفق الترتيب الأبجدي (للحروف اللاتينية) للحقول النصية، أو الترتيب الزمني للحقول التاريخية أو حسب الترقيم للحقول العددية. لعرض السجلات وفق ترتيب أبجدي حسب حقل محدد يجب أن تخبر محرك قاعدة البيانات بالحقل الذي تود أن تستخدمه كمرجع في الترتيب.

إذا كنت تستخدم واجهة تصميم الاستعلام في SQL Server Management Studio وترغب بتحديد الترتيب، فقم بما يلي:

- في قسم الرسم التخطيطي، يمكنك النقر بالزر الأيمن على الحقل وحدد ترتيب تصاعدي أو ترتيب تنازلي.
- في جزء المعايير من واجهة تصميم الاستعلام، تحت العمود Sort Type، انقر على خيار الترتيب تصاعدي أو تنازلي:

¹ يقصد بالفرز (Sort) إعادة ترتيب السجلات تبعاً لمحتويات حقل معين ترتيباً تصاعدياً أو تنازلياً. ويقصد بالتصفية (Filter) عرض السجلات التي تتطابق مع شرط معين.



إذا كان نوع بيانات الحقل عند ترتيب البيانات تصاعديا (Ascending):

- نصي (char، varchar، وبدائها) فسيكون ترتيب السجلات أبجدي.
- تأريخ أو توقيت (datetime أو smalldatetime) ترتب السجلات وفق ترتيب زمني.
- أعداد (int، decimal)، ترتب السجلات تدريجيا حسب القيمة.
- منطقية المستندة إلى (bit) فإن السجلات التي تحمل القيمة False ستظهر أولا.

يمكن عكس الترتيب عند اختيار الفرز بشكل التنازلي (Descending). بعد اختيار طريقة الفرز يمكنك تنفيذ الاستعلام.

تطبيق عملي: استخدام الشروط وتحليل البيانات

قم بتشغيل SQL Server Management Studio، وانقر على RealEstate1 بالزر الأيمن في متصفح الكائنات واختر استعلام جديد.

لتحديد طريقة الفرز في لغة SQL، يستخدم التعبير ORDER BY، بناء للجملة التالية:

```
SELECT What FROM WhatObject ORDER BY WhatField;
```

يجب أن يحدد الحقل الذي يستخدم كمرجع في عملية الفرز، في المثال التالي الحصول على قائمة الطلاب وفق الترتيب الأبجدي على أساس الحقل LastName:

```
SELECT FirstName,
        LastName,
        Gender,
        ParentsNames,
        SPHome
FROM Students
ORDER BY LastName;
GO
```

وهذا من شأنه أن ينتج:

	FirstName	LastName	Gender	ParentsNames	SPHome
1	Tim	Amorros	Male	Eduardo Amorros	0
2	Arlene	Andriamirano	Female	Christianne Burton	1
3	Sherryl	Ashburn	Female	Shelia and Patrick Ashburn	0
4	Martha	Bastens	Female		0
5	Lester	Bell	Male	Bernadette and Paulin Bell	0
6	Millicent	Broadskey	Female		0
7	Carole	Chance	Female	Carole and Andy Chance	0
8	Catherine	Chang	Female	Julie Best and Dr. Peter Chang	0
9	Dean	Chen	Male	Alexandra Chen	1
10	Danilo	Chico	Male	Michelle and Stephen Chico	0
11	Antoinette	Clarck	Female	Daniel and Henriette Clarck	0
12	Martin	Davis	Male		0
13	Nehemiah	Dean	Male	Marie Rodnat	1
14	Ruby	DeGaram	Female	Lynn and Anthony DeGaram	0
15	Koko	Domba	Male	Dr. Julie and Mr. Jeremiah Domba	0
16	Arlette	Duma	Female	Ursula Duma	1
17	Charles	Edelman	Male	Jasmine Beaulieu	1
18	Paul	Farms	Male	Rebecca and Peter Farms	0

Query... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

بنفس الطريقة يمكنك الحصول على قائمة بأسماء الفتيات تليها قائمة الفتيان بإصدار أوامر قائمة مرتبة ترتيباً ألفبائياً على أساس الجنس، يمكن أن يكتب استعلام لذلك على النحو التالي:

```
SELECT FirstName, LastName, Gender, EmailAddress
FROM Students
ORDER BY Gender;
GO
```

للحصول على قائمة جميع الطلاب مرتبة أبجدياً حسب الحقل LastName نكتب:

```
SELECT * FROM Students
ORDER BY LastName;
GO
```

افتراضياً، يتم ترتيب السجلات تصاعدياً، ومع ذلك يمكن فرض الترتيب التصاعدي باستخدام كلمة **ASC** بعد الحقل، مثلاً لفرز الأسماء السابقة في ترتيب تصاعدي تكتب العبارة على النحو التالي:

```
SELECT * FROM Students
ORDER BY LastName ASC;
GO
```

من ناحية أخرى، إذا كنت تريد فرز السجلات في ترتيب عكسي، يمكنك استخدام الكلمة **DESC** بدلاً من ذلك، مثال:

```
SELECT FirstName,
       LastName,
       Gender,
       ParentsNames,
       SPHome
FROM Students
ORDER BY LastName DESC;
GO
```

هذا من شأنه أن ينتج:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT FirstName,
       LastName,
       Gender,
       ParentsNames,
       SPHome
FROM Students
ORDER BY LastName DESC;
GO

```

Results Messages

	FirstName	LastName	Gender	ParentsNames	SPHome
1	Ismael	Zara	Male	Melinda and Robert Zara	0
2	Janet	West	Female	Kirsten Farmer	1
3	Anselme	Waters	Male	Claudette and Dr. Celestin Waters	1
4	Maurice	Walken	Male		0
5	Justin	Vittas	Male	Clarice Vittas	1
6	Suzanna	Verde	Female	Suzanne and Thommy Verder	0
7	Judith	Steinberg	Female	Jim	0
8	Harriette	Sans	Female	Thomas Sans	0
9	Martine	Quarles	Female		0
10	Sebastien	Porter	Male	John and Christine Porter	0
11	Santos	Pacheco	Male	Dr. Julia Santana	0
12	George	Orion	Male	Jennifer Freemont	1
13	Ella	Napolis	Female	Sandra and Pedro Napolis	0
14	Thomas	Moore	Male	Deborah Moore	0
15	Arthur	Milley	Male	Jeffrey Milley	0
16	Ann	Miller	Female		0
17	Victoria	Milchen	Female	Paul Milchen	1
18	Donnie	Mart	Female	Alan Marly	1

Q CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

تطبيق عملي: فرز السجلات

1- لعرض قائمة بقيم الخصائص بدءا من القيم المجهولة تليها الأقل كلفة منها، اكتب العبارة التالية:

```

SELECT house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.Bedrooms AS [Beds],
       house.Bathrooms AS [Baths],
       house.MarketValue AS [Value]
FROM Properties house
ORDER BY house.MarketValue
GO

```

2- اضغط F5 للتنفيذ.

3- لعرض قائمة بقيم الممتلكات مرتبة زمنيا بدءا من الأحدث غير الاستعلام كالاتي:

```
SELECT house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.Bedrooms AS [Beds],
       house.Bathrooms AS [Baths],
       house.MarketValue AS [Value]
FROM Properties house
ORDER BY YearBuilt DESC
GO
```

4- اضغط F5 للتنفيذ.

العمليات وتحليل البيانات:

رأينا طريقة تحليل البيانات باستخدام واجهة تصميم الاستعلام، مثال:

The screenshot shows a Microsoft Access window titled 'Table - dbo.Videos*' with a 'Summary' tab. A dropdown menu for 'Videos' is open, showing options for columns: Video ID, Video Title, Director, © Year, Length, and Rating. Below the menu is a table with columns: Column, Alias, Table, Output, Sort Type, and Sort Order. The 'Output' column has checkboxes checked for the first three rows. Below this is the SQL query: 'SELECT * FROM Videos'. At the bottom is a data grid with the following content:

	Video ID	Video Title	Director	© Year	Length	Rating
▶	1	A Few Good Men	Rob Reiner	1992	138 Minutes	R
	2	The Silence of the Lambs	Jonathan Demme	1991	1 hr. 18 Mns	
	3	The Distinguished Gentleman	James Groeling	NULL	112 Minutes	R
	4	The Lady Killers	Joel Coen & Ethan Coen	2004	104 Minutes	R
	5	Ghosts of Mississippi	Rob Reiner	NULL	130 Minutes	PG-13
	6	Red Heat	Walter Hill	NULL	NULL	NULL
	7	Mystic River	NULL	2003	2 hrs. 17 min	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom of the window, there are navigation buttons and a status bar showing '1 of 7' and 'Cell is Read Only.'

رأينا كيفية تحليل البيانات عن طريق إنشاء وتنفيذ الاستعلام، وبدلاً من تحديد جميع البيانات كما فعلنا حتى الآن باستخدام كلمة SELECT، يمكنك كتابة شرط يطبق على قاعدة البيانات لعزل سجلات معينة.

عند تحليل البيانات أو عند القيام بتصميم استعلام باستخدام واجهة التصميم، يمكنك كتابة تعبير يستخدم شرط (أو أكثر) من العمليات المنطقية التي رأيناها سابقاً، مثال للتعبير عن شرط: < '1993/31 /12'

وهذا يعني تحديد التواريخ التي بعد عام 1993.

التصفية باستخدام WHERE:

بعد كتابة عبارة الانتقاء SELECT يمكنك استخدام الكلمة WHERE وفق الصيغة الأساسية على النحو التالي:

```
SELECT What FROM WhatObject WHERE Expression;
```

يمكن إدراج معايير باستخدام عبارات جبرية، منطقية، سلاسل حرفية، كما يمكن دمج مجموعة من العبارات لإنشاء ما يسمى **معيار الفرز**، ويكتب التعبير باستخدام الصيغة:

```
ColumnName=Value
```

يجب أن يكون العامل ColumnName حقل موجود في جدول، تليه علامة المساواة = ثم العامل Value الذي يمثل القيمة التي من شأنها أن تحدد الشرط.

هنا مثال من قاعدة بيانات الطلاب، للحصول على قائمة من الطالبات:

```
SELECT DateOfBirth, LastName, FirstName,  
       Gender, State, ParentsNames  
FROM Students  
WHERE Gender='Female';  
GO
```

نتائج الاستعلام:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT DateOfBirth, LastName, FirstName,
       Gender, State, ParentsNames
FROM Students
WHERE Gender='Female';
GO

```

Results Messages

	DateOfBirth	LastName	FirstName	Gender	State	ParentsNames
1	1991-10-05 00:00:00.000	Hoak	Suzie	Female	MD	Christianne Hoak
2	1996-04-10 00:00:00.000	Clark	Antoinette	Female	MD	Daniel and Henriette Clark
3	1996-06-02 00:00:00.000	West	Janet	Female	MD	Kirsten Farmer
4	1997-06-12 00:00:00.000	Chang	Catherine	Female	DC	Julie Best and Dr. Peter Chang
5	1995-10-07 00:00:00.000	Ashburn	Sherryl	Female	MD	Shelia and Patrick Ashburn
6	1998-10-11 00:00:00.000	DeGaram	Ruby	Female	MD	Lynn and Anthony DeGaram
7	1990-10-22 00:00:00.000	Chance	Carole	Female	MD	Carole and Andy Chance
8	1989-10-05 00:00:00.000	Lobo	Brenda	Female	VA	Priscilla and Michel Lobo
9	1988-03-12 00:00:00.000	Verde	Suzanna	Female	MD	Suzanne and Thommy Verder
10	1991-10-08 00:00:00.000	Franse	Mincy	Female	MD	Dr. Melanie and Cleve Franse
11	1989-02-15 00:00:00.000	Andriamirano	Arlene	Female	VA	Christianne Burton
12	1989-06-23 00:00:00.000	Ledoux	Gabrielle	Female	MD	Brenda and Patrick Fasten
13	1990-08-20 00:00:00.000	Lobila	Koko	Female	DC	Aquilla and Dr. Diallo Lobila
14	1995-09-28 00:00:00.000	Duma	Arlette	Female	MD	Ursula Duma
15	1996-08-02 00:00:00.000	Sans	Harriette	Female	VA	Thomas Sans
16	1996-06-11 00:00:00.000	Howerson	Bernadette	Female	MD	Caron and Rick Howerson
17	1989-02-22 00:00:00.000	Steinberg	Judith	Female	MD	Jim
18	1991-04-24 00:00:00.000	Napolis	Ella	Female	MD	Sandra and Pedro Napolis

Query executed succ... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROOSH 00:00:00 27 rows

عند استعمال WHERE يمكنك أيضا استخدام الترتيب باستخدام ORDER BY لفرز قائمة السجلات استنادا لحقل محدد، مثال:

```

SELECT DateOfBirth, LastName, FirstName,
       Gender, State, ParentsNames
FROM Students
WHERE State='MD'
ORDER BY LastName;
GO

```

نتيجة الاستعلام:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT DateOfBirth, LastName, FirstName,
       Gender, State, ParentsNames
FROM Students
WHERE State='MD'
ORDER BY LastName;
GO

```

Results Messages

	DateOfBirth	LastName	FirstName	Gender	State	ParentsNames
1	1989-02-14 00:00:00.000	Amorros	Tim	Male	MD	Eduardo Amoros
2	1995-10-07 00:00:00.000	Ashburn	Sherryl	Female	MD	Shelia and Patrick Ashburn
3	1995-02-12 00:00:00.000	Bastens	Martha	Female	MD	
4	1995-12-06 00:00:00.000	Bell	Lester	Male	MD	Bernadette and Paulin Bell
5	1989-09-10 00:00:00.000	Broadskey	Millicent	Female	MD	
6	1990-10-22 00:00:00.000	Chance	Carole	Female	MD	Carole and Andy Chance
7	1997-06-20 00:00:00.000	Chico	Danilo	Male	MD	Michelle and Stephen Chico
8	1996-04-10 00:00:00.000	Clarck	Antoinette	Female	MD	Daniel and Henriette Clarck
9	1996-10-14 00:00:00.000	Davis	Martin	Male	MD	
10	1995-12-02 00:00:00.000	Dean	Nehemiah	Male	MD	Marie Rodnat
11	1998-10-11 00:00:00.000	DeGaram	Ruby	Female	MD	Lynn and Anthony DeGaram
12	1995-09-28 00:00:00.000	Duma	Arlette	Female	MD	Ursula Duma
13	1996-12-26 00:00:00.000	Edelman	Charles	Male	MD	Jasmine Beaulieu
14	1996-04-30 00:00:00.000	Farms	Paul	Male	MD	Rebecca and Peter Farms
15	1991-10-08 00:00:00.000	Franse	Mincy	Female	MD	Dr. Melanie and Cleve Franse
16	1996-04-19 00:00:00.000	Fuller	Clint	Male	MD	Shelley and Dr. Oregon Fuller
17	1990-02-21 00:00:00.000	Hagers	Ralph	Male	MD	
18	1991-10-05 00:00:00.000	Hoak	Suzie	Female	MD	Christianne Hoak

Query executed s... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 39 rows

تطبيق عملي: فرز بالبيانات باستخدام WHERE

1- للاطلاع على قائمة الممتلكات فقط في منطقة معينة، اكتب ما يلي:

```

SELECT house.PropertyNumber AS [Prop #],
       house.Address,
       house.City,
       house.State,
       house.ZIPCode AS [Location],
       house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.MarketValue AS [Value]
FROM Properties house
WHERE house.ZIPCode < 20500
GO

```

2- اضغط F5 لتنفيذ الاستعلام.

3- للحصول على قائمة الممتلكات في ولاية فرجينيا فقط، اكتب الآتي:

```
SELECT house.PropertyNumber AS [Prop #],
       house.Address,
       house.City,
       house.State,
       house.ZIPCode AS [Location],
       house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.MarketValue AS [Value]
FROM Properties house
WHERE house.State = 'va'
GO
```

4- اضغط F5 للتنفيذ

5- للحصول على قائمة الممتلكات التي تم بناؤها في أو بعد عام 2000، اكتب:

```
SELECT house.Address,
       house.City,
       house.State,
       house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.MarketValue AS [Value]
FROM Properties house
WHERE house.YearBuilt >= 2000
GO
```

6- اضغط F5 لتنفيذ.

CENTRAL.RealEs...1 - Results.sql Object Explorer Details

```

SELECT house.Address,
       house.City,
       house.State,
       house.YearBuilt AS [Year Built],
       house.PropertyType AS [Type],
       house.MarketValue AS [Value]
FROM Properties house
WHERE house.YearBuilt >= 2000
GO

```

	Address	City	State	Year Built	Type	Value
1	10315 North Hacht Rd	College Park	MD	2000	Single Family	620724.00
2	10340 Helmes Street #408	Silver Spring	MD	2000	Condominium	242775.00
3	NULL	Chevy Chase	NULL	2001	Single Family	525450.00
4	14005 Sniders Blvd	Laurel	MD	2002	Townhouse	412885.00
5	680 Prushia Rd	Washington	DC	2000	Single Family	555885.00
6	10340 Helmes Street #1012	Silver Spring	MD	2000	Condominium	252775.00
7	980 Phorwick Street	Washington	DC	2004	Single Family	735475.00
8	7307 Everett Hwy	Washington	DC	2006	Townhouse	420550.00
9	10340 Helmes Street#1006	Silver Spring	MD	2000	Condominium	258445.00
10	7303 Warfield Court	Tysons Corner	VA	2006	Single Family	825775.00
11	12401 Conniard Ave	Takoma Park	MD	2004	Townhouse	280775.00
12	5121 Riehl Ace	Fairfax	VA	2002	Townhouse	325620.00
13	944 Fryer Ave	Chevy Chase	MD	2002	Single Family	625665.00
14	1950 Galego Street	Germantown	MD	2007	Single Family	428665.00
15	NULL	Takoma Park	MD	2000	Conbominium	225885.00
16	12366 Fowler Ave	Alexandria	VA	2007	Townhouse	402815.00

Quer... CENTRAL (9.0 SP2) CENTRAL\Administrator (52) RealEstate1 00:00:00 16 rows

7- للحصول على قائمة الممتلكات في ولاية ماريلاند مرتبة حسب الرمز البريدي، غير

العبارة على النحو التالي:

```

SELECT house.PropertyType AS Type,
       house.YearBuilt AS [Year Built],
       house.City,
       house.ZIPCode,
       house.Bedrooms AS Beds,
       house.Bathrooms AS Baths,
       house.MarketValue AS Value
FROM Properties house
WHERE house.State = 'md'
ORDER BY house.ZIPCode
GO

```

8- اضغط F5 للتنفيذ.

إلغاء بعض السجلات:

رأينا سابقا إمكانية استخدام المشغل NOT الذي ينفي صحة التعبير المنطقي، فمثلا للحصول على قائمة من الطلاب الذكور فقط، يمكنك أن تنفي الشرط السابق. للقيام بذلك اكتب NOT قبل ذلك الشرط، ولجعل هذا الشرط أسهل للقراءة يستحسن أن يكتب بين قوسين على النحو التالي:

```
SELECT DateOfBirth, LastName, FirstName,  
       Gender, State, ParentsNames  
FROM Students  
WHERE NOT (Gender = 'Female');  
GO
```

في هذا المثال، يمكنك الاستعلام عن قائمة السجلات التي لا تحمل قيمة فارغة في حقل معين:

```
SELECT DateOfBirth, LastName, FirstName,  
       State, ParentsNames  
FROM Students  
WHERE State IS NOT NULL;  
GO
```

عندما يتم تنفيذ هذا الاستعلام ستعرض قائمة الطلاب حيث الحقل State يحمل قيمة.

تطبيق عملي: الفرز مع استثناء السجلات

1- للاطلاع على قائمة الممتلكات التي تحمل قيمة للحقل PropertyNumber أكتب

استعلام على النحو التالي:

```
SELECT dbo.Properties.PropertyNumber AS [Prop #],  
       dbo.Properties.PropertyType AS Type,  
       dbo.Properties.YearBuilt AS [Year Built],  
       dbo.Properties.City,  
       dbo.Properties.State,  
       dbo.Properties.ZIPCode AS [ZIP Code],  
       dbo.Properties.Bedrooms AS Beds,  
       dbo.Properties.Bathrooms AS Baths,  
       dbo.Properties.MarketValue AS Value  
FROM dbo.Properties  
WHERE dbo.Properties.PropertyNumber IS NOT NULL  
GO
```

اضغط F5 لتنفيذ الاستعلام.

عمليات تحليل البيانات

استخدام الشروط لفرز البيانات:

رأينا طريقة فرز البيانات وفق شرط واحد، لكنك ستحتاج في كثير من الأحيان إلى الفرز وفق شرط يعتمد على شرط آخر، وهنا تسمح لك العبارات المنطقية بالجمع بين شرطين واستخدام الناتج.

تطبيق عملي: تمهيد لاستخدام الشروط لتحليل البيانات

1- قم بتشغيل Microsoft SQL Server and the SQL Server Management Studio.

2- انقر بالزر الأيمن على قاعدة البيانات RealEstate1 واختر New Query.

عبارات الشرط المنطقية:

لنفترض في شركة العقارات أن هناك زبون يود شراء منزل عائلة واحدة في أنحاء Silver Spring بولاية Maryland، بناء على هذا الطلب سنقوم بالتحقق من قائمة العقارات من خلال الاستعلام باستخدام شرطين لكل الممتلكات:



- يجب أن يكون بيت لأسرة واحدة
- يجب أن يتواجد في أنحاء Maryland.

الآن يمكنك أن تبدأ باستخدام الشروط لتصميم استعلام يعرض فقط منازل عائلة واحدة،

مثال:

The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Object Explorer Details' pane is active, displaying a table with columns: Column, Table, Output, Filter, and Or... The 'Filter' column contains the text '= 'Single Family''. Below this, the SQL query is shown:

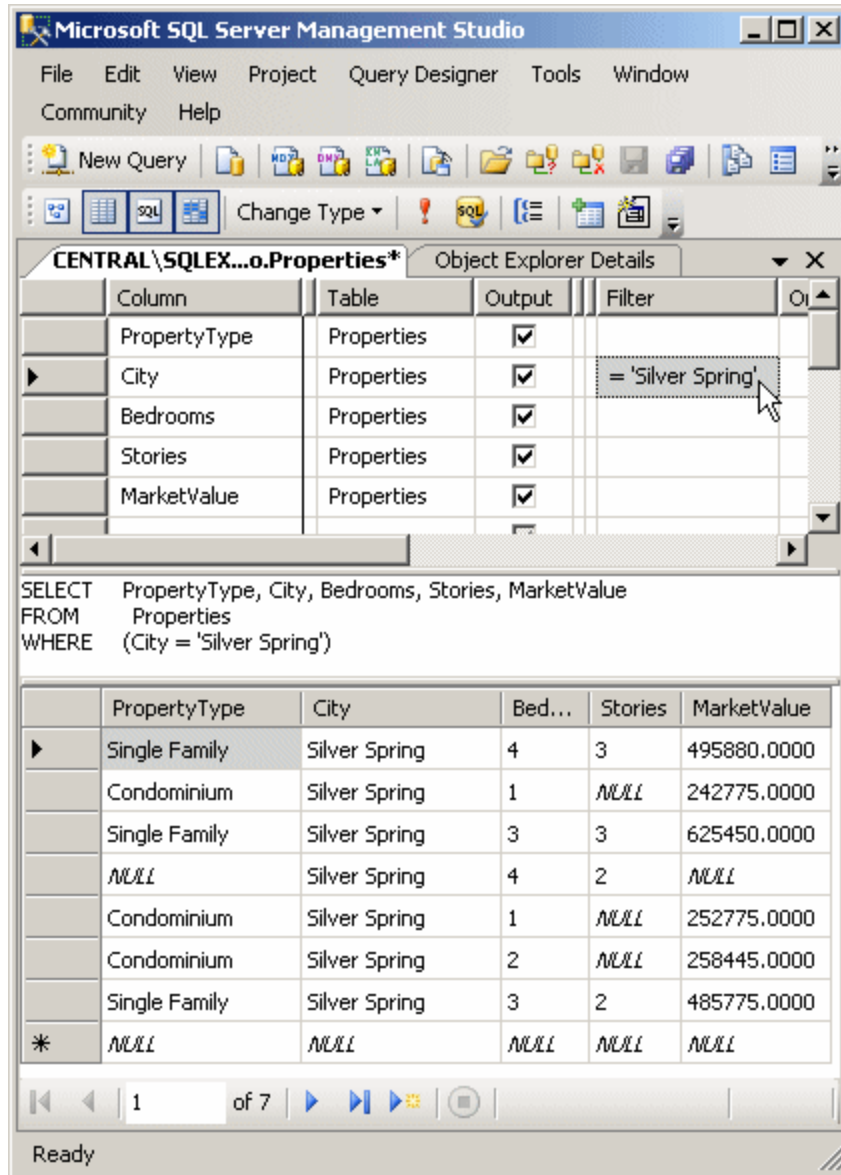
```
SELECT PropertyType, City, Bedrooms, Stories, MarketValue
FROM Properties
WHERE (PropertyType = 'Single Family')
```

The query results are displayed in a table with the following data:

PropertyType	City	Bed...	Stories	MarketValue
Single Family	Silver Spring	4	3	495880.0000
Single Family	College Park	4	2	620724.0000
Single Family	Laurel	NULL	2	422625.0000
Single Family	Alexandria	3	NULL	345660.0000
Single Family	Silver Spring	3	3	625450.0000
Single Family	Chevy Chase	NULL	3	525450.0000
Single Family	Washington	5	3	555885.0000
Single Family	College Park	5	2	485995.0000
Single Family	Washington	4	1	725475.0000

The status bar at the bottom indicates 'Ready'.

الشرط الآخر يتطلب أن يكون المنزل موجودا في Silver Spring، يمكن تحديد ذلك على النحو التالي:



للجمع بين الشروط يتم استخدام عبارة منطقية وحيدة تجمع الشرطين السابقين، للربط بين الجمل المنطقية في SQL يستخدم المعامل AND وفق الصيغة التالية:

```
SELECT WhatColumn(s)
FROM WhatObject
WHERE Condition1 AND Condition2
```

حيث WhatColumn(s) و WhatObject هي نفسها العوامل التي رأيناها سابقا، والكلمة

AND هي الوسيط الذي يستخدم للجمع بين الشروط، حيث يكتب كل شرط باستخدام الصيغة:

Column operator Value

Condition1 يمثل الشرط الأول الذي يتم التحقق منه، وإذا لم يتحقق الشرط الأول فلن يتم تنفيذ الاستعلام ولن يتم التحقق من الشرط الثاني. إذا تم التحقق من صحة الشرط الأول فسيتم فحص الشرط الثاني.

مثال: لنفترض أننا نريد الحصول على قائمة من الطالبات اللاتي يقطن في ولاية Maryland. يمكن كتابة هذا الاستعلام على النحو التالي:

```
SELECT FirstName, LastName, Gender, City, State
FROM Students
WHERE Gender = 'female' AND State = 'md';
```

والنتيجة:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
SELECT FirstName, LastName, Gender, City, State
FROM Students
WHERE (Gender = 'female') AND (State = 'md')
```

The results pane displays the following data:

FirstName	LastName	Gender	City	State
Suzie	Hoak	Female	Hyattsville	MD
Antoinette	Clarck	Female	Rockville	MD
Janet	West	Female	Bethesda	MD
Sherryl	Ashburn	Female	Chevy Chase	MD
Ruby	DeGaram	Female	Silver Spring	MD
Carole	Chance	Female	Rockville	MD
Suzanna	Verde	Female	Silver Spring	MD
Mincy	Franse	Female	Laurel	MD

يمكنك أيضا نفي تحقق الشرط باستعمال المعامل NOT.

تطبيق عملي: استخدام الارتباط المنطقي بين الشروط

1- للاطلاع على قائمة منازل عائلة واحدة المتواجدة في Silver Spring، اكتب الآتي:

```
SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.State,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'Single Family') AND (City = 'Silver
Spring');
GO
```

2- اضغط F5 لتنفيذ الاستعلام

3- لنفترض أن الزبون يود شراء منزل في البلدة لكنه لا يستطيع أن ينفق أكثر من 400,000 دولار.

في هذه الحالة نقوم بالاستعلام عن قائمة المنازل المطلوبة التي تكلف أقل من 400,000

دولار، غير عبارة الاستعلام على النحو التالي:

```
SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.State,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'Townhouse') AND (h.MarketValue < 400000);
GO
```

4- اضغط F5 لتنفيذ الاستعلام:

CENTRAL.RealEs...ts\Results.sql Object Explorer Details

```

SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.State,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'Townhouse') AND
      (h.MarketValue < 400000);
GO

```

Results Messages

	Prop #	Type	Year Built	City	State	ZIP Code	Beds	Baths	Value
1	207850	Townhouse	1988	Rockville	MD	20854	3	2.5	325995.00
2	NULL	Townhouse	1975	Washington	NULL	NULL	4	NULL	366775.00
3	447597	Townhouse	1992	Hyattsville	MD	20782	3	2	365880.00
4	297497	Townhouse	2004	Takoma Park	MD	20910	3	2.5	280775.00
5	826927	Townhouse	2002	Fairfax	VA	22232	3	1.5	325620.00

Query exe... CENTRAL (9.0 SP2) CENTRAL\Administrator (52) RealEstate1 00:00:00 5 rows

5- لنفترض أن العميل يفضل أن يكون المنزل في مدينة Rockville.

للحصول على قائمة من المنازل في مدينة Rockville التي تكلف أقل من 400,000 دولار، قم بتغيير عبارة الاستعلام إلى النحو التالي:

```

SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'Townhouse') AND
      (h.MarketValue < 400000) AND
      (h.City = 'rockville');
GO

```

6- اضغط F5 لتنفيذ الاستعلام.

الفصل المنطقي (OR) في عبارة الشرط:

لنفترض أن الزبون يود شراء منزل أسرة واحدة أو منزل في البلدة، لإعداد قائمة العقارات المطلوبة يجب الاستعلام عن الممتلكات التي تحقق أحد هذين الخيارين.

لكتابته استعلام من هذا النوع تستخدم عبارات شرطية مع الفصل بينها بالمعامل **OR** وفقا للصيغة التالية:

```
SELECT WhatColumn(s)
FROM WhatObject
WHERE Condition1 OR Condition2
```

تطبيق عملي: استخدام الفصل المنطقي في عبارة الشرط

1- للاطلاع على قائمة تحقق الشرط السابق يمكن تغيير الاستعلام إلى النحو التالي:

```
SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'single family') OR
      (h.PropertyType = 'Townhouse');
GO
```

2- اضغط F5 للتنفيذ.

عمليات منطقية أخرى في الاستعلامات:

BETWEEN: مجال الاختيار

إذا كانت لديك مجموعة من القيم وأردت أن تعرف فيما إذا كانت قيمة معينة محصورة بين القيمتين معلومتين يمكنك استعمال العامل **BETWEEN** ضمن العبارة الشرطية **WHERE**، حيث تتركب العبارة الشرطية من الكلمة **BETWEEN** والكلمة **AND** بين قيمتين وفقا للصيغة الأساسية التالية:

```
WHERE Expression BETWEEN Start AND End
```

حيث **Expression** اسم الحقل الذي تريد فحص بياناته، العامل **Start** هي القيمة الابتدائية المعتبرة لمجال الفحص، العامل **End** هو أعلى قيمة من مجال الفحص. تنفيذ هذا الشرط ينتج قائمة من القيم بين القيمتين **Start** و **End**.

لإنشاء استعمال باستخدام BETWEEN حدد الحقل المطلوب ثم من تحت العمود Filter الذي يوافق الحقل الذي تريد فحص الشرط عليه اكتب عبارة الشرط. في المثال الآتي استعمال يعيد قائمة الطلاب الذين يقطنون حيث الرمز البريدي محصور بين 20500 و21000:

```
SELECT FirstName, LastName, Gender, City, State, ZIPCode, SPHome
FROM Students
WHERE ZIPCode BETWEEN '20500' AND '21000'
```

عادة ما تكتب عبارة BETWEEN بين قوسين. الاستعمال السابق ينتج:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL query:

```
SELECT FirstName, LastName, Gender, City, State, ZIPCode, SPHome
FROM Students
WHERE (ZIPCode BETWEEN '20900' AND '21000')
```

The results pane shows a table with the following data:

FirstName	LastName	Gender	City	State	ZIPCode	SPHo...
Sebastien	Porter	Male	Silver Spring	MD	20902	False
Nehemiah	Dean	Male	Silver Spring	MD	20910	True
Ruby	DeGaram	Female	Silver Spring	MD	20906	False
Suzanna	Verde	Female	Silver Spring	MD	20904	False
Paul	Farms	Male	Silver Spring	MD	20902	False
Clint	Fuller	Male	Silver Spring	MD	20904	False
Bernadette	Howerson	Female	Silver Spring	MD	20910	False
Judith	Steinberg	Female	Takoma Park	MD	20912	False
Ella	Napolis	Female	Silver Spring	MD	20906	False

The status bar at the bottom indicates 'Ready' and shows '1 of 14' records.

تطبيق عملي: الاستعلام عن السجلات المحصورة بين قيمتين

1- للحصول على قائمة من العقارات المحصورة أسعارها بين 350,000 و 450,000 دولار اكتب استعلاما على النحو التالي:

```
SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.MarketValue BETWEEN 350000 AND 450000);
GO
```

2- اضغط F5 لتنفيذ العبارة.

3- لن تجد بعد تنفيذ الاستعلام أي عقار بهذه المواصفات في مدينة Silver Spring، لذا سنقترح على الزبون إلقاء نظرة على المنازل الأخرى في أنحاء Silver Spring، لإنشاء استعلام بذلك سنحتاج لسحب قائمة المنازل حيث الرقم البريدي بين 20800 و 20999 (مقاطعة Montgomery بولاية Maryland)، لأجل ذلك نغير الاستعلام كما يلي:

```
SELECT h.PropertyNumber AS [Prop #],
       h.PropertyType AS Type,
       h.YearBuilt AS [Year Built],
       h.City,
       h.ZIPCode AS [ZIP Code],
       h.Bedrooms AS Beds,
       h.Bathrooms AS Baths,
       h.MarketValue AS Value
FROM Properties h
WHERE (h.PropertyType = 'Townhouse') AND
      (h.MarketValue < 400000) AND
      (h.ZIPCode = (SELECT h.ZIPCode
                    WHERE h.ZIPCode BETWEEN '20500' AND '21000'));
GO
```

4- اضغط F5 لتنفيذ الاستعلام.

سلسلة الاختيارات باستخدام IN:

إذا كانت لديك مجموعة من السجلات وتريد العثور على سجل أو مجموعة من السجلات من بينها، يمكنك استخدام العامل IN عن طريق إضافته إلى عبارة الشرط WHERE. يمثل هذا العامل طريقة أخرى لاستخدام OR ضمن العبارة الشرطية، وتستخدم IN حسب الصيغة:

```
IN(Expression1, Expression2, Expression_n)
```

يمثل كل عامل من العبارة السابقة أحد قيم الحقول، وهو ما يعادل:

```
Expression1 OR Expression2 OR Expression_n, etc.
```

لإنشاء عبارة الشرط أثناء تصميم الاستعلام حدد الأعمدة ثم تحت العمود Filter للحقل الذي تود تطبيق الشرط عليه واكتب عبارة الشرط باستخدام IN.

مثال للحصول على قائمة الطلبة الذين يقطنون في Silver Spring أو Rockville أو Chevy Chase، يكتب على النحو التالي:

```
SELECT FirstName, LastName, Gender, City, State, ZIPCode, SPHome  
FROM Students  
WHERE City IN ('silver spring', 'rockville', 'chevy chase');
```

يستحسن كتابة العبارة WHERE بين قوسين. العبارة أعلاه من شأنها أن تنتج:

Microsoft SQL Server Management Studio

File Edit View Project Query Designer Tools Window Community Help

Object Explorer Details

Column	Table	Output	Filter
FirstName	Students	<input checked="" type="checkbox"/>	
LastName	Students	<input checked="" type="checkbox"/>	
Gender	Students	<input checked="" type="checkbox"/>	
City	Students	<input checked="" type="checkbox"/>	IN ('silver spring', 'rockville', 'chevy chase')
State	Students	<input checked="" type="checkbox"/>	
ZIPCode	Students	<input checked="" type="checkbox"/>	
SPHome	Students	<input checked="" type="checkbox"/>	

```

SELECT FirstName, LastName, Gender, City, State, ZIPCode, SPHome
FROM Students
WHERE (City IN ('silver spring', 'rockville', 'chevy chase'))

```

FirstName	LastName	Gender	City	State	ZIPCode	SPHo...
Sebastien	Porter	Male	Silver Spring	MD	20902	False
Antoinette	Clark	Female	Rockville	MD	20875	False
Nehemiah	Dean	Male	Silver Spring	MD	20910	True
Sherryl	Ashburn	Female	Chevy Chase	MD	20870	False
Ruby	DeGaram	Female	Silver Spring	MD	20906	False
Carole	Chance	Female	Rockville	MD	20875	False
Suzanna	Verde	Female	Silver Spring	MD	20904	False
Danilo	Chico	Male	Chevy Chase	MD	20872	False
Paul	Farms	Male	Silver Spring	MD	20902	False
Lester	Bell	Male	Silver Spring	MD	20815	False
Clint	Fuller	Male	Silver Spring	MD	20904	False
Bernadette	Howerson	Female	Silver Spring	MD	20910	False

Ready

تطبيق عملي: استخدام العامل IN

1- للحصول على قائمة منازل المدينة لعائلة واحدة أكتب استعلام على النحو التالي:

```

SELECT house.PropertyNumber AS [Prop #],
       house.PropertyType AS Type,
       house.YearBuilt AS [Year Built],
       house.City,
       house.State,
       house.ZIPCode AS [ZIP Code],
       house.Bedrooms AS Beds,
       house.Bathrooms AS Baths,

```



```

house.MarketValue AS Value
FROM Properties house
WHERE house.PropertyType IN ('Single Family', 'Townhouse');
GO

```

2- اضغط F5 لتنفيذ الاستعلام.

3- للحصول على قائمة المنازل لعائلة واحدة الواقعة في Rockville، غير عبارة الاستعلام كالتالي:

```

SELECT house.PropertyNumber AS [Prop #],
       house.PropertyType AS Type,
       house.YearBuilt AS [Year Built],
       house.City,
       house.State,
       house.ZIPCode AS [ZIP Code],
       house.Bedrooms AS Beds,
       house.Bathrooms AS Baths,
       house.MarketValue AS Value
FROM Properties house
WHERE (house.PropertyType IN ('Single Family', 'Townhouse')) AND
      (house.City = 'Rockville');
GO

```

4- اضغط F5 للتنفيذ.

المعامل LIKE:

في معظم المعايير المحددة سابقا بالكلمة WHERE ينبغي التطابق التام مع القيم المحددة، قد لا تذكر بالضبط - في بعض الحالات - القيمة التي تود الفرز باستخدامها ولكن بإمكانك أن تحدد قيمة مشابهة، للاستعلام عن قيم مشابهة يمكنك استخدام المشغل LIKE.

في واجهة تصميم الاستعلام انقر على الخانة المقابلة للعمود Filter والحقل الذي تود تطبيق الشرط عليه، ثم اكتب عبارة الشرط باستخدام LIKE على نحو الصيغة التالية:

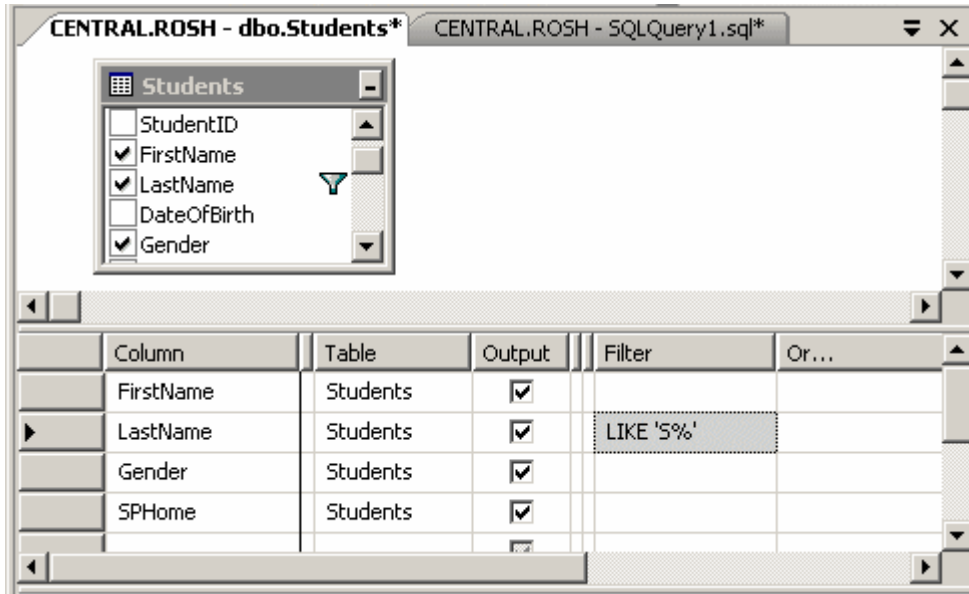
Expression LIKE pattern

يمثل Expression التعبير الذي سيتم تقييمه، حيث ينبغي أن يكون تعبيراً واضحاً وصحيحاً، العامل pattern يمكن أن يكون مطابقاً للقيمة التي تود الاستعلام عنها (وفي هذه الحالة تقوم LIKE مقام العلامة =)، أو قيمة مشابهة لها.

فكرة استخدام العامل LIKE هو الاستعلام عن قيمة تقريبية للنتيجة التي تريدها، وهنا يمكن الاستعانة بالرمز %.

استخدام LIKE مع أي حرف %:

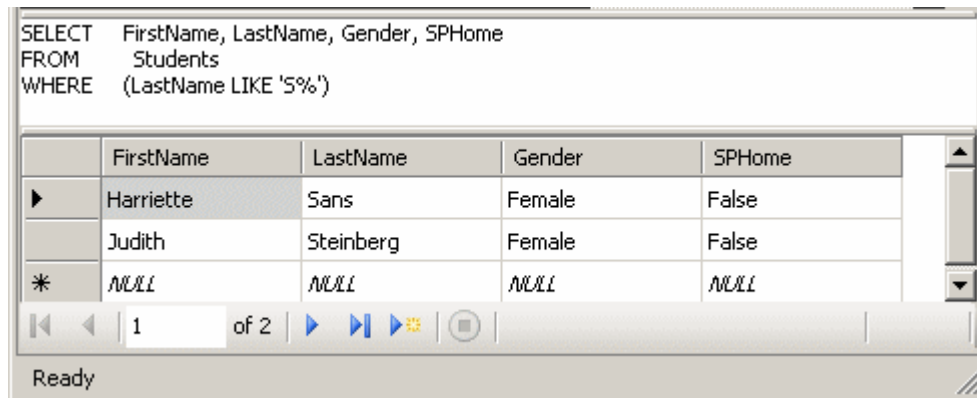
إذا كنت تريد البحث عن قيمة مشابهة استخدم في أي موضع من جملة البحث الرمز %،
 إذا كتبت حرفاً قبل الرمز % مثلاً 'S%'، فإن الشرط يعني البحث عن أي سلسلة نصية تبدأ
 بالحرف S، كما في المثال التالي:



مثل للاستعلام عن قائمة الطلبة الذي تبدأ أسماءهم بالحرف S:

```
SELECT FirstName, LastName, Gender, SPHome
FROM Students
WHERE (LastName LIKE 'S%')
```

والنتيجة:



ويمكن الاستعلام بنفي تحقق الشرط السابق بإضافة NOT، مثال:

```
SELECT FirstName, LastName, Gender, SPHome
FROM Students
WHERE (NOT (LastName LIKE 'S%'))
```

والنتيجة:

The screenshot shows a SQL Server query window titled 'CENTRAL.ROSH - SQLQuery1.sql*'. The 'Students' table is selected, and the following columns are checked: FirstName, LastName, DateOfBirth, and Gender. The query editor contains the following SQL code:

```
SELECT FirstName, LastName, Gender, SPHome
FROM Students
WHERE (NOT (LastName LIKE '5%'))
```

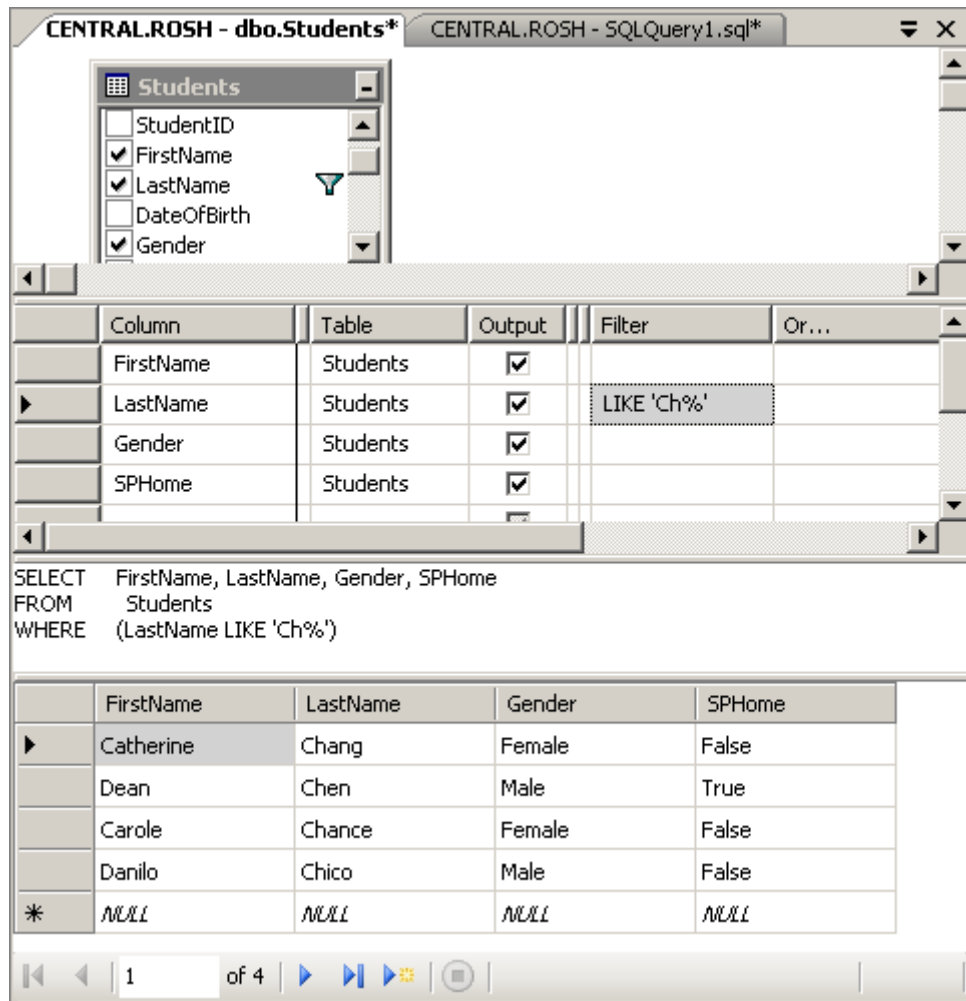
The results grid displays the following data:

FirstName	LastName	Gender	SPHome
Sebastien	Porter	Male	False
Suzie	Hoak	Female	False
Antoinette	Clarck	Female	False
Koko	Domba	Male	False
Janet	West	Female	True
Catherine	Chang	Female	False
Nehemiah	Dean	Male	True
Sherryl	Ashburn	Female	False

مثال بنفس الطريقة هذه المرة لعرض قائمة بأسماء الطلبة الذين تبتدئ أسماؤهم بالحرفين Ch:

```
SELECT      FirstName, LastName, Gender, SPHome
FROM        Students
WHERE       (LastName LIKE 'Ch%')
```

ونائج الاستعلام:



بدلاً من البحث عن القيم التي تبدأ بحرف أو مجموعة من الأحرف يمكن ابتداء عبارة LIKE بالرمز % للبحث عن القيم التي تنتهي بحرف أو أحرف معينة، في هذه الحالة كافة السلاسل النصية التي تنتهي بالحرف أو الأحرف المحددة ستؤخذ بعين الاعتبار.

إذا لم تتذكر بداية أو نهاية السلسلة النصية (القيمة) التي تريد البحث عنها، يمكنك البحث باستخدام قيمة متضمنة خلال السلسلة، مثلاً؛ للبحث عن أسماء الطلبة الذين تتضمن أسماءهم الحرفين "an" يمكنك استخدام LIKE عن القيمة '%an%'، مثال:

```
SELECT      FirstName, LastName, Gender, SPHome
FROM        Students
WHERE       (LastName LIKE '%an%')
```

ونتيجة الاستعلام:

The screenshot shows a SQL query window with the following content:

```

SELECT FirstName, LastName, Gender, SPHome
FROM Students
WHERE (LastName LIKE '%an%')

```

Column	Table	Output	Filter	Or...
FirstName	Students	<input checked="" type="checkbox"/>		
LastName	Students	<input checked="" type="checkbox"/>	LIKE '%an%'	
Gender	Students	<input checked="" type="checkbox"/>		
SPHome	Students	<input checked="" type="checkbox"/>		

	FirstName	LastName	Gender	SPHome
▶	Catherine	Chang	Female	False
	Nehemiah	Dean	Male	True
	Carole	Chance	Female	False
	Mincy	Franse	Female	False
	Arlene	Andriamirano	Female	True
	Harriette	Sans	Female	False
	Charles	Edelman	Male	True
*	NULL	NULL	NULL	NULL

مثل باقي عبارات SQL يمكن البحث بنفي تحقق الشرط باستخدام العامل NOT.

استخدام LIKE مع مجال من الأحرف []:

يستخدم الرمز % مسبقا أو متقدما بأي حرف أو مجموعة من الأحرف. إذا أردت تحديد مجال البحث فقط في مجموعة من الأحرف، يمكنك تحديد هذه المجموعة بكتابة أول وآخر حرف بين معقوفين.

للقيام بذلك اكتب الرمز [يليه الحرف الأدنى لمجال الأحرف، يليه الرمز - متبوعا بالحرف الأعلى لمجال أحرف البحث ثم الرمز].

مثال للبحث باستخدام مجال الأحرف بين p و s، استخدم '[p-s]'. ثم من جهة اليسار أو اليمين أو في كلا الطرفين أضف الرمز % لتحديد فيما إذا كان نص البحث يتضمن أي حرف أو مجموعة من الحروف قبل أو بعد مجال الأحرف المحدد. مثال:

```
SELECT    FirstName, LastName, Gender, SPHome
FROM      Students
WHERE     (LastName LIKE '%[p-s]')
```

نتيجة البحث في هذه الحالة تكون قائمة الطلاب الذين تنتهي أسماؤهم بأحد الأحرف p، r، q. يمكنك ملاحظة ذلك في ناتج الاستعلام:

Column	Table	Output	Filter	Or...
FirstName	Students	<input checked="" type="checkbox"/>		
LastName	Students	<input checked="" type="checkbox"/>	LIKE '%[p-r]'	
Gender	Students	<input checked="" type="checkbox"/>		
SPHome	Students	<input checked="" type="checkbox"/>		

```
SELECT    FirstName, LastName, Gender, SPHome
FROM      Students
WHERE     (LastName LIKE '%[p-r]')
```

FirstName	LastName	Gender	SPHome
Sebastien	Porter	Male	False
Clint	Fuller	Male	False
Ann	Miller	Female	False
Arthur	Junger	Male	True
NULL	NULL	NULL	NULL

البحث عن نصوص لا تنتهي بمجموعة من الأحرف:

للبحث عن أي عبارة نصية لا تحتوي على حرف أو مجموعة من الأحرف يمكنك استخدام الرمز ^ بداخل المعقوفين قبل كتابة مجال مجموعة الأحرف. مثال:

```
SELECT    FirstName, LastName, Gender, SPHome
FROM      Students
```

```
WHERE (LastName LIKE '%[^p-r]')
```

وستكون النتيجة قائمة الطلاب الذين تنتهي أسماؤهم بغير الأحرف p، q، r، s.

يمكنك الاستغناء عن هذه العبارة لتحصل على نفس النتيجة باستخدام عبارة النفي NOT

.LIKE

الدوال وتحليل البيانات:

استخدام الدوال المدمجة:

رأينا سابقاً أن بإمكانك استخدام الدوال المدمجة في SQL Server، فإن لم تجد ما يناسبك من الدوال قم بإنشاء دالتك المناسبة كما رأينا في درس سابق.

لاستخدام أحد الدوال المدمجة اكتب اسم الدالة ثم بين قوسين أضف المعلمات التي تحتاجها، محترماً في ذلك تطابق نوع البيانات. في هذا المثال تستخدم الدالة **DATEDIFF** التي تعيد الفرق بين تاريخين لعرض أعمار الطلاب:

```
SELECT FirstName, LastName, Gender,  
       DATEDIFF(year, DateOfBirth, GETDATE()) AS Age  
FROM Students;  
GO
```

تحسب الدالة أعمار الطلاب بحساب الفرق بين تاريخ اليوم وتاريخ الميلاد، والنتائج:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT FirstName, LastName, Gender,
       DATEDIFF(year, DateOfBirth, GETDATE()) AS Age
FROM Students;
GO

```

Results Messages

	FirstName	LastName	Gender	Age
1	Sebastien	Porter	Male	12
2	Suzie	Hoak	Female	16
3	Antoinette	Clark	Female	11
4	Koko	Domba	Male	17
5	Janet	West	Female	11
6	Catherine	Chang	Female	10
7	Nehemiah	Dean	Male	12
8	Sherryl	Ashburn	Female	12
9	Santos	Pacheco	Male	9
10	Mohamed	Husseini	Male	9
11	Dean	Chen	Male	10
12	Ruby	DeGaram	Female	9
13	Carole	Chance	Female	17
14	Justin	Vittas	Male	16
15	Ismael	Zara	Male	17
16	Anselme	Waters	Male	17

Q CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

وباستخدام الدوال المدمجة يمكنك إدراج أي من العبارات التي استعرضناها آنفا. مثال:

```

SELECT FirstName, LastName, Gender, DateOfBirth, SPHome
FROM Students
WHERE (DateOfBirth BETWEEN CONVERT(DATETIME, '1995-01-01', 102) AND
      CONVERT(DATETIME, '1999-12-31', 102))

```

كما يمكنك أيضا أن تمرر إلى الدالة أيًا من العبارات التي استعرضنا سابقا.

استخدام دوالك الخاصة:

إذا لم تكن أي من الدوال المضمنة تلبي حاجتك، يمكنك إنشاء دالة خاصة بك واستخدامها خلال تحليل البيانات. طبعًا، يجب عليك أولاً إنشاؤها. مثال إنشاء دالتين في قاعدة البيانات

:ROSH

```

/* =====
Author:      FunctionX
Create date: Friday 6 April, 2007
Description: This function is used
              to get the full name of a student
=====*/

```



```

CREATE FUNCTION GetFullName
(
    @FName varchar(20),
    @LName varchar(20)
)
RETURNS varchar(41)
AS
BEGIN
    RETURN @LName + ', ' + @FName;
END;
GO
/* =====
Author:      FunctionX
Create date: Saturday 7 April, 2007
Description: This function is used
              to display Yes or No
===== */
CREATE FUNCTION ShowYesOrNo
(
    @SPHomeStatus bit
)
RETURNS varchar(3)
AS
BEGIN
    DECLARE @Result varchar(3);

    IF @SPHomeStatus = 0
        SET @Result = 'No';
    ELSE
        SET @Result = 'Yes';

    RETURN @Result;
END;
GO

```

بعد إعداد الدالة يمكن استدعاؤها عند كتابة الاستعلام، لأجل ذلك أكتب dbo متبوعة بنقطة ثم اسم الدالة وبعدها اكتب بين الأقواس المعالم (إن وجدت) مع القيم المناسبة لها. مثال:

```

SELECT StudentID,
       dbo.GetFullName(FirstName, LastName) AS [Student's Name],
       Gender,
       dbo.ShowYesOrNo(SPHome) AS [Live's in a Single Parent Home?],
       ParentsNames AS [Parents' Names]
FROM Students;
GO

```

والناتج:

CENTRAL.ROSH - SQLQuery1.sql* Object Explorer Details

```

SELECT StudentID,
       dbo.GetFullName(FirstName, LastName) AS [Student's Name],
       Gender,
       dbo.ShowYesOrNo(SPHome) AS [Live's in a Single Parent Home?],
       ParentsNames AS [Parents' Names]
FROM Students;
GO

```

StudentID	Student's Name	Gender	Live's in a Single Parent Home?	Parents' Names
1	Porter, Sebastien	Male	No	John and Christine Porter
2	Hoak, Suzie	Female	No	Christianne Hoak
3	Clark, Antoinette	Female	No	Daniel and Henriette Clark
4	Domba, Koko	Male	No	Dr. Julie and Mr. Jeremiah Domba
5	West, Janet	Female	Yes	Kirsten Farmer
6	Chang, Catherine	Female	No	Julie Best and Dr. Peter Chang
7	Dean, Nehemiah	Male	Yes	Marie Rodnat
8	Ashburn, Sherryl	Female	No	Shelia and Patrick Ashburn
9	Pacheco, Santos	Male	No	Dr. Julia Santana
10	Husseini, Moha...	Male	No	Drs. Phyllis and Ezra Hussein
11	Chen, Dean	Male	Yes	Alexandra Chen
12	DeGaram, Ruby	Female	No	Lynn and Anthony DeGaram
13	Chance, Carole	Female	No	Carole and Andy Chance
14	Vittas, Justin	Male	Yes	Clarice Vittas
15	Zara, Ismael	Male	No	Melinda and Robert Zara
16	Waters, Anselme	Male	Yes	Claudette and Dr. Celestin Waters
17	Lobo, Brenda	Female	No	Priscilla and Michel Lobo
18	Verde, Suzanna	Female	No	Suzanne and Thommy Verder

Query executed successful... CENTRAL (9.0 SP2) CENTRAL\Administrator (51) ROSH 00:00:00 52 rows

تقنيات أخرى لتحليل البيانات:

• التجميع Count:

إحصاء السجلات:

```

SELECT COUNT(GenderID) AS Sexes
FROM Persons

```

حساب عدد السجلات على أساس معيار محدد، على سبيل المثال حساب عدد الطالبات في المدرسة:

```

SELECT COUNT(*) AS Girls
FROM Students
WHERE (Sex = 'Female')

```

تطبيق نسبة خصم (20%) لجميع المبيعات الموجودة في المتجر:

```

SELECT ItemNumber, ItemName, ItemSize, UnitPrice,
       UnitPrice - UnitPrice * 0.20 AS [Discounted Price]

```

```
FROM StoreItems
```

باستخدام CASE لتخصيص كل حالة:

```
SELECT ItemNumber, ItemName, ItemSize, ItemCategoryID, UnitPrice,  
       CASE ItemCategoryID  
         WHEN 1 THEN UnitPrice - UnitPrice * 0.10  
         WHEN 2 THEN UnitPrice - UnitPrice * 0.50  
         WHEN 3 THEN UnitPrice - UnitPrice * 0.35  
         ELSE UnitPrice - UnitPrice * 0.45  
       END AS [Discounted Price]  
FROM StoreItems
```

العلاقات وتكامل البيانات

المفتاح الأساسي:

قواعد البيانات العلائقية:

قاعدة البيانات العلائقية هي النظام الذي يسمح بتدفق المعلومات بين كائنات قاعدة بيانات. مثلا في قاعدة بيانات البنك يمكنك استخدام كائن لإنشاء حسابات للعملاء واستخدام كائنات أخرى لمعالجة المعاملات التي قام بها أصحاب الحسابات المصرفية، حيث أن نفس العميل قد يحتاج إلى إجراء معاملات أخرى مختلفة لنفس الحساب.

لتطبيق قوانين قواعد البيانات العلائقية يمكنك إنشاء كائنات تتولى أمر تماسك البيانات بين مكونات قاعدة البيانات.

عند تبادل البيانات بين كائنات مختلفة من قاعدة بيانات ينبغي التأكد من إمكانية وصول كائن إلى آخر، تتمثل الكائنات التي تخزن البيانات في **الجدول**.

لتسيير تدفق البيانات من جدول إلى آخر، يجب على الجدول الذي يحتوي البيانات أن يجعلها متاحة للجدول الآخر، حيث يتوجب على الجدول:

- تمييز السجلات بقيمة فريدة لكل سجل من الجدول لتجنب الالتباس

- إتاحة البيانات للجدول أخرى

- ألا يقوم بغرض يقوم به جدول آخر.

يتم تحقيق هذه النقاط عن طريق تحديد حقل خاص كمفتاح لكل جدول. يسمى هذا الحقل **بالمفتاح الأساسي**.

في قاعدة بيانات العلائقية، كما هو الحال بالنسبة لمعظم قواعد البيانات، ينبغي على كل جدول أن يحتوي على الأقل مفتاح أساسيا. فمثلا، يمكن تعيين مفتاح أساسي في جدول حساب

العميل من قاعدة بيانات البنك، لأن لكل حساب مصرفي للعميل يجب أن يكون فريدا. ويمكن أن يحتوي الجدول أكثر من حقل كمفتاح أساسي إذا اقتضى الأمر.

بعد تحديد حقل المفتاح الأساسي، يجب أيضا تحديد نوع البيانات حقل المفتاح الأساسي. إذا كان الجدول يحتوي حقلا معروفا وواضحا بأنه فريد من نوعه، كرقم الكتاب في مكتبة مثلا. يمكنك تعيين نوع بياناته char أو varchar وجعله مفتاح أساسي.

في حالات أخرى كثيرة، لا يمكنك ضمان عدم تكرار بيانات في أي حقل من الجدول، أي لا يمكن تحديد حقل من شأنه أن يضم بيانات فريدة، في هذه الحالة ينبغي إنشاء حقل فريد وجعله مفتاح أساسي. يكون نوع بيانات هذا الحقل في الغالب عددا صحيحا.

تطبيق عملي: مقدمة إلى العلاقات

1- افتح SQL Server Management Studio وقم بإجراء اتصال بالخادم.

2- بالزر الأيمن انقر على اسم الخادم ثم انقر على New Query.

3- لإنشاء قاعدة بيانات جديدة، اكتب الاستعلام التالي:


```
-- =====
-- Database:      YugoNationalBank
-- Author:        FunctionX
-- Date Created:  Monday 09 April 2007
-- =====
USE master
GO

-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'YugoNationalBank'
)
DROP DATABASE YugoNationalBank
GO

CREATE DATABASE YugoNationalBank
GO
```

إنشاء مفتاح أساسي أثناء التصميم:

لإنشاء مفتاح أساسي باستخدام SQL Server Management Studio، قم بإنشاء الحقل ثم حدد نوع البيانات الخاصة به.

- ثم من شريط الأدوات، انقر على زر تعيين المفتاح الأساسي 
- يمكنك أيضا النقر بالزر الأيمن على الحقل ثم اختيار Set Primary Key، مثال:

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
FirstName	varchar(20)	<input checked="" type="checkbox"/>
LastName	varchar(20)	<input type="checkbox"/>

إنشاء مفتاح أساسي باستخدام SQL:

لإنشاء حقل مفتاح أساسي باستخدام SQL، اكتب على يمين تعريف الحقل PRIMARY KEY. مثال:

```
CREATE TABLE Persons
(
    PersonID int identity(1,1) PRIMARY KEY NOT NULL,
    FirstName varchar(20),
    LastName varchar(20) NOT NULL
);
```

قيد المفتاح الأساسي (Primary Key Constraint):

يمكنك تمييز المفتاح الأساسي باستخدام الصيغة التالية:

```
CONSTRAINT PrimaryKeyName PRIMARY KEY(ColumnName)
```

في هذه الصيغة الكلمات المفتاحية CONSTRAINT و PRIMARY KEY مطلوبة، وتمثل PrimaryKeyName اسم المفتاح الأساسي، وبين القوسين يتم تحديد اسم العمود الذي سيتم استخدامه كمفتاح أساسي. اصطلاحا يبدأ اسم المفتاح الأساسي بالأحرف PK_ يليها اسم الجدول. مثال:

```
CREATE TABLE Persons
(
    PersonID int identity(1,1) NOT NULL,
    FirstName varchar(20),
    LastName varchar(20) NOT NULL,
    CONSTRAINT PK_Persons PRIMARY KEY(PersonID)
);
GO
```

المفتاح الخارجي (Foreign Key):

لنتخيل في قاعدة بيانات المصرف أن العميل سيأتي لإيداع أموال، في هذه الحالة نحن بحاجة إلى تحصيل معلومات كافية عن حساب العميل، ثم تقديم تلك المعلومات إلى جدول المعاملات المالية. حيث يجب -كما ذكرنا سابقا- على جدول العملاء أن يوفر البيانات اللازمة للجداول الأخرى التي قد تحتاج إلى هذه البيانات، لجعل هذا التدفق من المعلومات ممكنا من جدول إلى آخر، يجب إنشاء علاقة بين هذه الجداول.

إنشاء "حقل غريب" أثناء تصميم الجدول:

لتجعل من الممكن للجدول "ب" تلقي البيانات من الجدول "أ"، يجب أن يحتوي الجدول "ب" على حقل يمثل الجدول "أ"، يقوم هذا الحقل مقام "السير" أو "الرابط" بين الجدولين. كما هو شأن السير... هذا الحقل الذي في الجدول "ب" لا ينتمي إليه بل هو تمثيل للجدول "أ" يسمح لكلا الجدولين بالتواصل، لهذا السبب يسمى بالحقل الغريب.

الحقل الغريب هو حقل من الجدول يخزن بيانات من جدول آخر.

لإنشاء حقل غريب أثناء تصميم الجدول، قم بإنشاء حقل مراعي القواعد التالية:

- ينبغي أن يأخذ الحقل الغريب نفس اسم الحقل الأساسي للجدول الذي يمثله (ليس شرطاً).
- يجب أن يكون الحقل الغريب والحقل الأساسي من نفس نوع بيانات.

في هذا المثال، الحقل GenderID هو مفتاح خارجي:

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
FirstName	varchar(20)	<input checked="" type="checkbox"/>
LastName	varchar(20)	<input type="checkbox"/>
GenderID	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

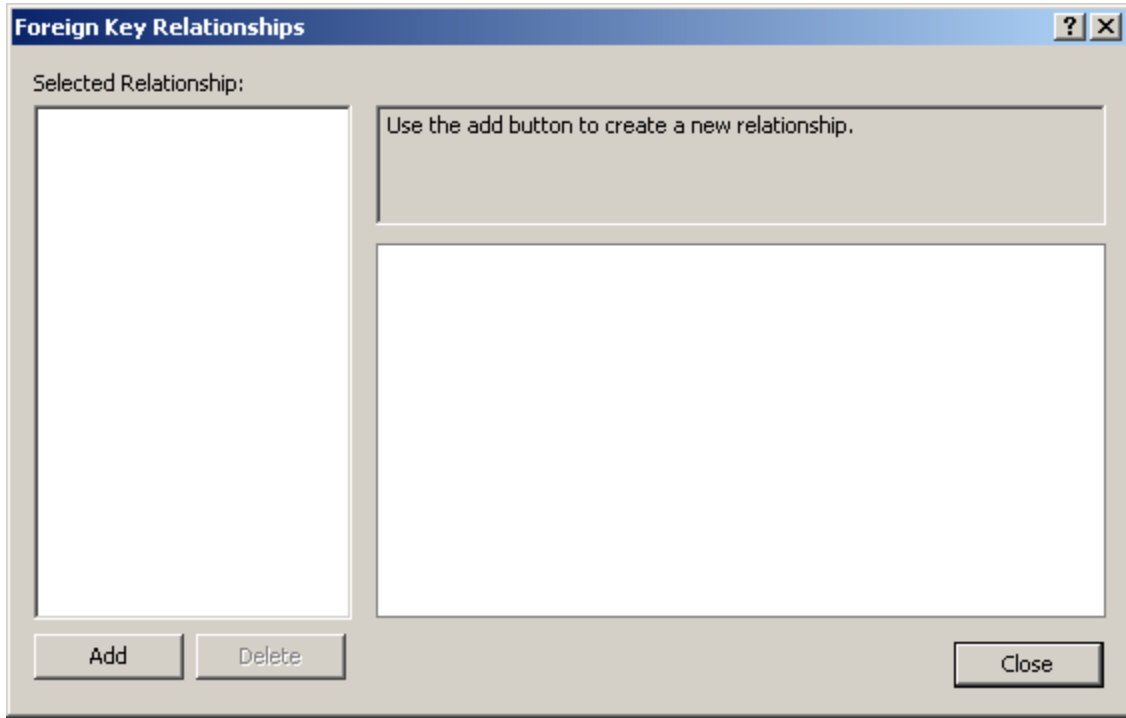
يسمى الجدول الذي يحتوي على المفتاح الأساسي والذي يحمل البيانات التي ستستخدم في جدول آخر بالجدول الأساسي أو الجدول الأب. ويسمى الجدول الذي سيتلقى البيانات من الجدول الأب بالجدول الغريب أو الجدول الابن.

إنشاء مفتاح خارجي أثناء التصميم:

لإنشاء علاقة بين الجدولين:

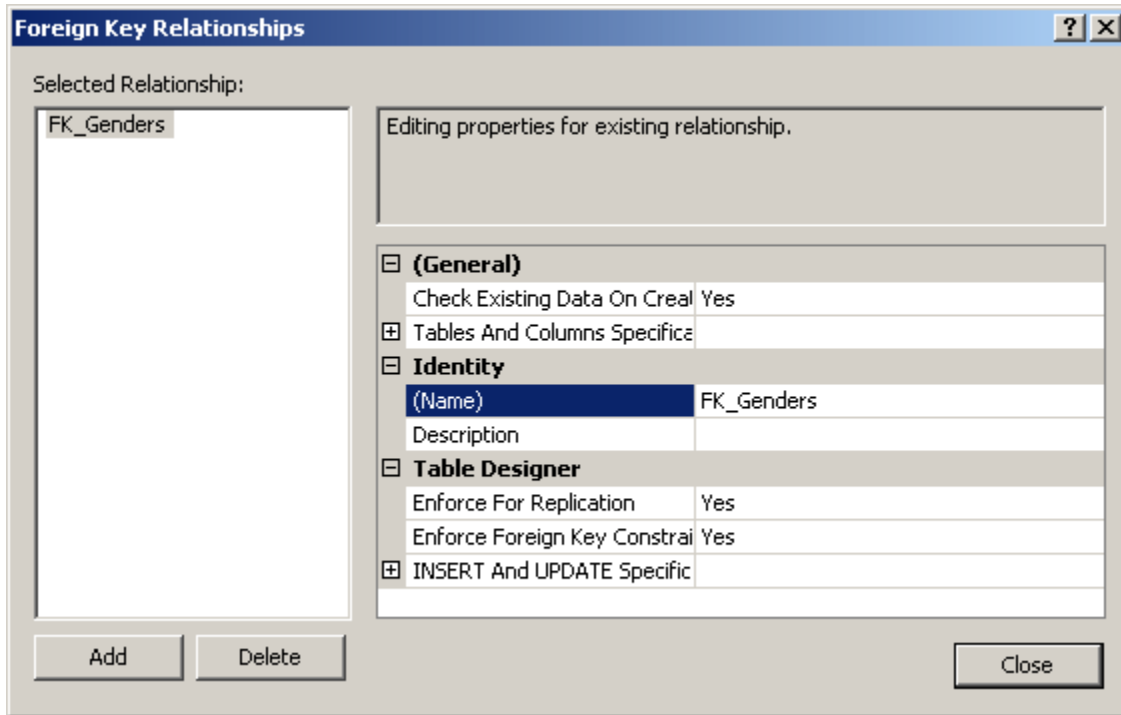
1- من متصفح الكائنات، افتح الجدول الابن في واجهة التصميم.

2- بالزر الأيمن وفي مكان من الجدول انقر على **Relationships...**



3- انقر فوق إضافة من علبة الحوار Foreign Key Relationships

4- سيقترح عليك المحرر اسماً يبدأ بالأحرف FK_ للدلالة على Foreign Key. يمكنك قبول أو تغيير هذا الاسم في الجانب الأيمن، قم بتوسيع النطاق Identity وكتب الاسم المناسب في الخانة (Name):



5- بنفس الطريقة يمكنك إنشاء علاقات أخرى إذا لزم الأمر، لأجل ذلك انقر على Add.

احذف أي منها قم بتحديدتها وانقر على Delete.

عند الانتهاء، انقر على Close.

إنشاء مفتاح خارجي باستخدام SQL:

يمكنك إنشاء علاقة بين جدولين بإضافة مفتاح خارجي عند إنشاء الجدول باستخدام استعمال SQL، وفقا للصيغة التالية:

```
FOREIGN KEY REFERENCES ParentTableName(ForeignKeyCcolumn)
```

العبارة FOREIGN KEY والكلمة REFERENCES ضروريتان. في موضع الكلمة ParentTableName، أدخل اسم الجدول الأساسي الذي يحتوي على البيانات التي سيتم الوصول إليها في الجدول الحالي، وبين القوسين، أدخل اسم الحقل الأساسي للجدول الأصل،
مثال:

```
CREATE TABLE Persons
(
    PersonID int identity(1,1) PRIMARY KEY NOT NULL,
    FirstName varchar(20),
    LastName varchar(20) NOT NULL,
    GenderID int NULL FOREIGN KEY REFERENCES Genders(GenderID)
);
```

قيد المفتاح الخارجي (Foreign Key Constraint):

لاحظ أن المفتاح الخارجي لا يملك اسما كما هو حال المفتاح الأساسي. إذا لم تقم بتحديد اسم للمفتاح الخارجي، فسيقوم مترجم SQL تلقائياً بإنشاء اسم افتراضي. إذا أردت بنفسك تحديد الاسم، بعد عبارة إنشاء الحقل أضف CONSTRAINT يليها الاسم المرغوب واستمر في بقية الاستعلام كما رأينا أعلاه. مثال:

```
CREATE TABLE Persons
(
    PersonID int identity(1,1) PRIMARY KEY NOT NULL,
    FirstName varchar(20),
    LastName varchar(20) NOT NULL,
    GenderID int NULL CONSTRAINT FKGender
    FOREIGN KEY REFERENCES Genders(GenderID)
);
```

العلاقات بين الجداول:

قاعدة البيانات العلائقية هي التي تسمح بتدفق البيانات من جدول إلى آخر، لإعداد الجداول لأجل ذلك قم بإنشاء المفاتيح الأساسية والمفاتيح الخارجية كالذي قمنا به حتى الآن. بعد ذلك يمكنك ربط جدولين بإنشاء علاقة بينهما.

إذا لم تكن قد أنشأت مفتاح خارجي باستخدام SQL، يمكنك إنشاؤه عند إنشاء علاقة بين الجدولين.

إنشاء علاقة بين جدولين:

لإنشاء علاقة بين جدولين:

1- افتح الجدول الابن في واجه التصميم

2- انقر بالزر الأيمن على Relationships... إذا اقتضى الأمر، انقر على Add لإنشاء مفتاح خارجي، أعطه اسما من (Name) تحت Identity.

3- قم بتحديد المفتاح الخارجي المسؤول عن العلاقة.

4- في الجانب الأيمن، قم بتوسيع Tables And Columns Specification

5- انقر على زر عرض التفاصيل

6- في قائمة الاختيارات Primary Key Table، حدد الجدول الأب الذي يحمل البيانات الأولية

7- تحت الجدول الأب، قم بتحديد حقل المفتاح الأساسي

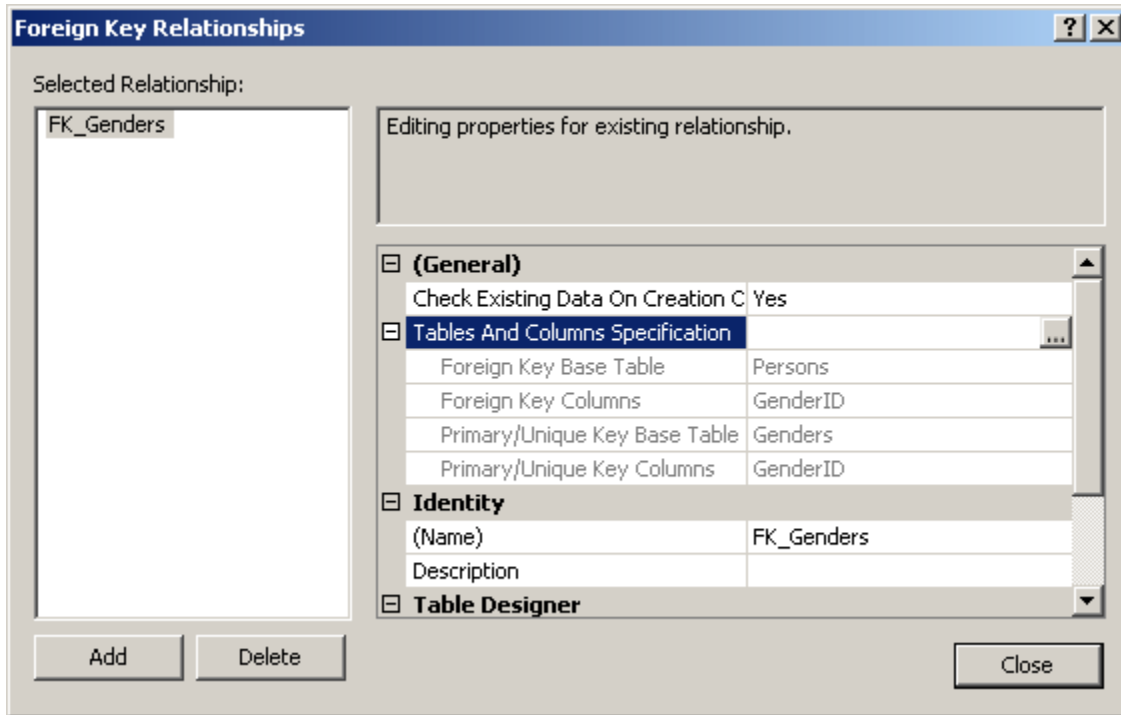
8- تحت Foreign Key Table، تأكد من تحديد الجدول الحالي.

تحت اسم الجدول الابن، انقر على اسم الحقل الغريب. مثال:

The screenshot shows the 'Tables and Columns' dialog box. The 'Relationship name' field is set to 'FK_Genders'. The 'Primary key table' is 'Genders' and the 'Foreign key table' is 'Persons'. The 'Primary key column' is 'GenderID' and the 'Foreign key column' is 'GenderID'. The 'OK' and 'Cancel' buttons are visible at the bottom.

9- انقر OK.

بعد إنشاء العلاقة بين الجدولين ستظهر في النطاق Tables And Column Specification



10- بنفس الطريقة، يمكنك إنشاء علاقات أخرى بالنقر على Add وإنشاء الارتباط.

عند الانتهاء، انقر على Close.

تطبيق علمي: إنشاء علاقات بين الجداول

لإتمام إنشاء قاعدة البيانات مثالنا، وإنشاء المفاتيح الأساسية والمفاتيح الخارجية، وإدخال بعض السجلات في الجداول، قم بكتابة الاستعلام التالي:

```
-- =====
-- Database:      YugoNationalBank
-- Author:        FunctionX
-- Date Created:  Monday 09 April 2007
-- =====
USE master
GO

-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'YugoNationalBank'
)
DROP DATABASE YugoNationalBank
GO

CREATE DATABASE YugoNationalBank
GO
-- =====
-- Database: YugoNationalBank
```

```

-- Table:      Locations
-- =====
USE YugoNationalBank
GO

IF OBJECT_ID('dbo.Locations', 'U') IS NOT NULL
    DROP TABLE dbo.Locations
GO

CREATE TABLE Locations
(
    LocationID int Identity(1,1) NOT NULL,
    LocationCode varchar(10) NOT NULL,
    Address varchar(120),
    City varchar(50),
    State varchar(50),
    Notes text NULL,
    CONSTRAINT PK_Locations PRIMARY KEY (LocationID)
);
GO
-- =====
-- Database: YugoNationalBank
-- Table:      AccountTypes
-- =====
USE YugoNationalBank
GO

IF OBJECT_ID('dbo.AccountTypes', 'U') IS NOT NULL
    DROP TABLE dbo.AccountTypes
GO

CREATE TABLE AccountTypes
(
    AccountTypeID int Identity(1,1) NOT NULL,
    AccountType varchar(40) NOT NULL,
    Notes text NULL,
    CONSTRAINT PK_AccountTypes PRIMARY KEY (AccountTypeID)
);
GO
-- =====
-- Database: YugoNationalBank
-- Table:      Employees
-- =====
IF OBJECT_ID('dbo.Employees', 'U') IS NOT NULL
    DROP TABLE dbo.Employees
GO
CREATE TABLE Employees
(
    EmployeeID int identity(1,1) NOT NULL,
    EmployeeNumber char(6),
    FirstName varchar(32),
    LastName varchar(32) NOT NULL,
    Title varchar(50),
    CanCreateNewAccount bit,
    HourlySalary smallmoney,
    EmailAddress varchar(100),
    Username varchar(20),
    Password varchar(20),
    Notes text,
    CONSTRAINT PK_Employees PRIMARY KEY (EmployeeID)
);

```

```

);
GO

-- =====
-- Database: YugoNationalBank
-- Table: Customers
-- =====
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers
GO
CREATE TABLE Customers
(
    CustomerID int Identity(1,1) NOT NULL,
    DateCreated datetime,
    AccountNumber varchar(20),
    AccountTypeID int Constraint FK_TypeOfAccount
        References AccountTypes(AccountTypeID),
    CustomerName varchar(50) NOT NULL,
    DateUpdated smallDateTime,
    Notes text,
    CONSTRAINT PK_Customers PRIMARY KEY (CustomerID)
);
GO

-- =====
-- Database: YugoNationalBank
-- Table: ChargeReasons
-- =====
IF OBJECT_ID('dbo.ChargeReasons', 'U') IS NOT NULL
    DROP TABLE dbo.ChargeReasons
GO

-- =====
-- Database: YugoNationalBank
-- Table: Deposits
-- =====
IF OBJECT_ID('dbo.Deposits', 'U') IS NOT NULL
    DROP TABLE dbo.Deposits
GO

CREATE TABLE Deposits
(
    DepositID int identity(1, 1) NOT NULL,
    LocationID int Constraint FK_DepositLocation
        References Locations(LocationID) NOT NULL,
    EmployeeID int Constraint FK_Clerk
        References Employees(EmployeeID),
    CustomerID int Constraint FK_Depositor
        References Customers(CustomerID) NOT NULL,
    DepositDate smalldatetime NOT NULL,
    DepositAmount smallmoney NOT NULL,
    Notes text,
    CONSTRAINT PK_Deposits PRIMARY KEY (DepositID)
);
GO

-- =====
-- Database: YugoNationalBank
-- Table: Withdrawals
-- =====
IF OBJECT_ID('dbo.Withdrawals', 'U') IS NOT NULL
    DROP TABLE dbo.Withdrawals

```

```

GO

CREATE TABLE Withdrawals
(
    WithdrawalID int identity(1, 1) NOT NULL,
    LocationID int Constraint FK_WithdrawalLocation
        References Locations(LocationID) NOT NULL,
    EmployeeID int Constraint FK_ProcessedBy
        References Employees(EmployeeID),
    CustomerID int Constraint FK_CustomerAccount
        References Customers(CustomerID) NOT NULL,
    WithdrawalDate smalldatetime NOT NULL,
    WithdrawalAmount smallmoney NOT NULL,
    WithdrawalSuccessful bit NOT NULL,
    Notes text,
    CONSTRAINT PK_Withdrawas PRIMARY KEY (WithdrawalID)
);

-- =====
-- Database: YugoNationalBank
-- Table:      CheckCashing
-- =====
IF OBJECT_ID('dbo.CheckCashing', 'U') IS NOT NULL
    DROP TABLE dbo.CheckCashing
GO

CREATE TABLE CheckCashing
(
    CheckCashingID int identity(1, 1) NOT NULL,
    LocationID int Constraint FK_BranchLocations
        References Locations(LocationID) NOT NULL,
    EmployeeID int Constraint FK_Employees
        References Employees(EmployeeID),
    CustomerID int Constraint FK_Customers
        References Customers(CustomerID) NOT NULL,
    CheckCashingDate smalldatetime NOT NULL,
    CheckCashingAmount smallmoney NOT NULL,
    CheckCashingSuccessful bit NOT NULL,
    Notes text,
    CONSTRAINT PK_CheckCashing PRIMARY KEY(CheckCashingID)
);
GO

```

2- اضغط F5 لتنفيذ الاستعلام.

مخططات العلاقات:

مخطط العلاقات هو الواجهة التي تعرض العلاقات بين الجداول في قاعدة بيانات.

لإنشاء مخطط:

1- في متصفح الكائنات، من بند قاعدة البيانات يمكنك النقر على Database Diagrams

2- سيعرض مربع حوار ليحيطكم علما بأن قاعدة البيانات ليس لديها أي مخطط. أقرأ الرسالة وانقر Yes

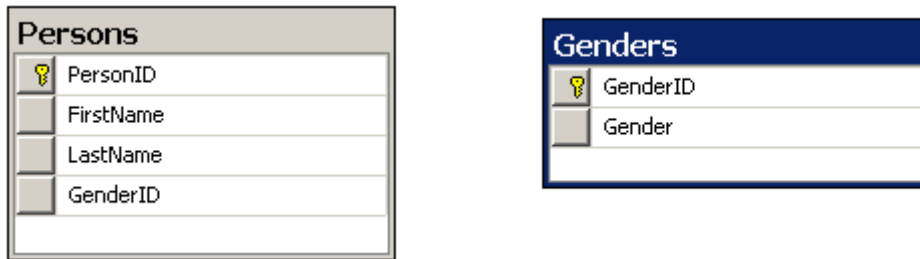
3- انقر بالزر الأيمن على Database Diagrams واختر New Database Diagram

4- في مربع الحوار Add Table انقر على كل جدول ثم انقر على Add، أو بدلا من ذلك يمكنك النقر نقرا مزدوجا على الجدول لإضافته

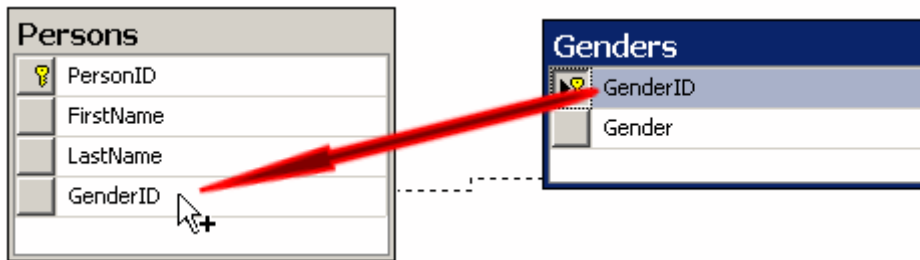
5- في مربع الحوار Add Table، انقر على Close

يمكنك النقر في شريط الأدوات على زر Zoom وتحديد قيمة أكبر أو أصغر.

لنقل مخطط أي جدول، يمكنك سحب شريط العنوان الخاص به. مثال:



6- لإنشاء علاقة بين الجدولين، يمكنك سحب المربع الرمادي من الجهة اليسرى لأي حقل من الجدول الأب وإسقاطه على أي حقل في الجدول الابن، والأفضل أن يتم الربط بين المفتاح الأساسي للجدول الأب مع الحقل الغريب من الجدول الابن. مثال:

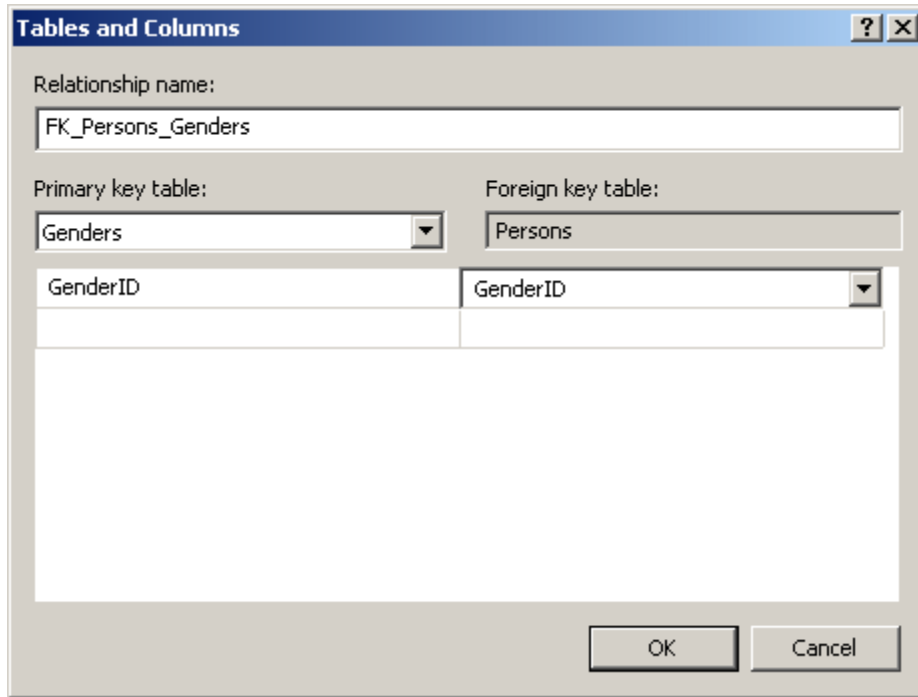


7- سيظهر مباشرة مربع الحوار Tables and Columns، ليعرض الحقل الذي تم سحبه والحقل الذي وقع الإفلات عليه.

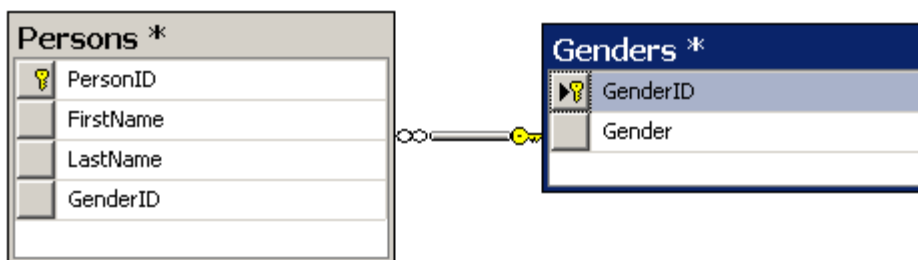
سيتم إظهار أي حقل تقوم باختياره، ولو لم يكن مناسباً للعلاقة بين الجدولين. ويجب تحديد الجدول الأب من القائمة Primary Key Table.

8- اختر من الجدول الأب، الحقل الأساسي.

9- حدد من الجدول الآخر الحقل الخارجي. مثال:



10- عند الانتهاء، انقر على OK، لتجد صلة ستنشأ بين الجدولين.



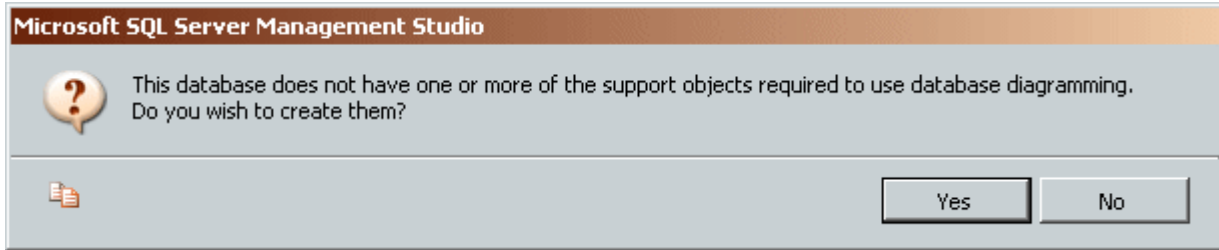
11- بنفس الطريقة، يمكنك إنشاء علاقات أخرى.

عند الانتهاء من ذلك، يمكنك حفظ المخطط وخلق قاعدة البيانات.

تطبيق عملي: إنشاء المخططات

1- في متصفح الكائنات، حدد قاعدة البيانات YugoNationalBank، ثم انقر على Database Diagrams

2- انقر على Yes من مربع الحوار الذي يعلمك بأن قاعدة البيانات هذه ليس لديها مخطط:



3- انقر بالزر الأيمن على قاعدة البيانات واختر New Database Diagram

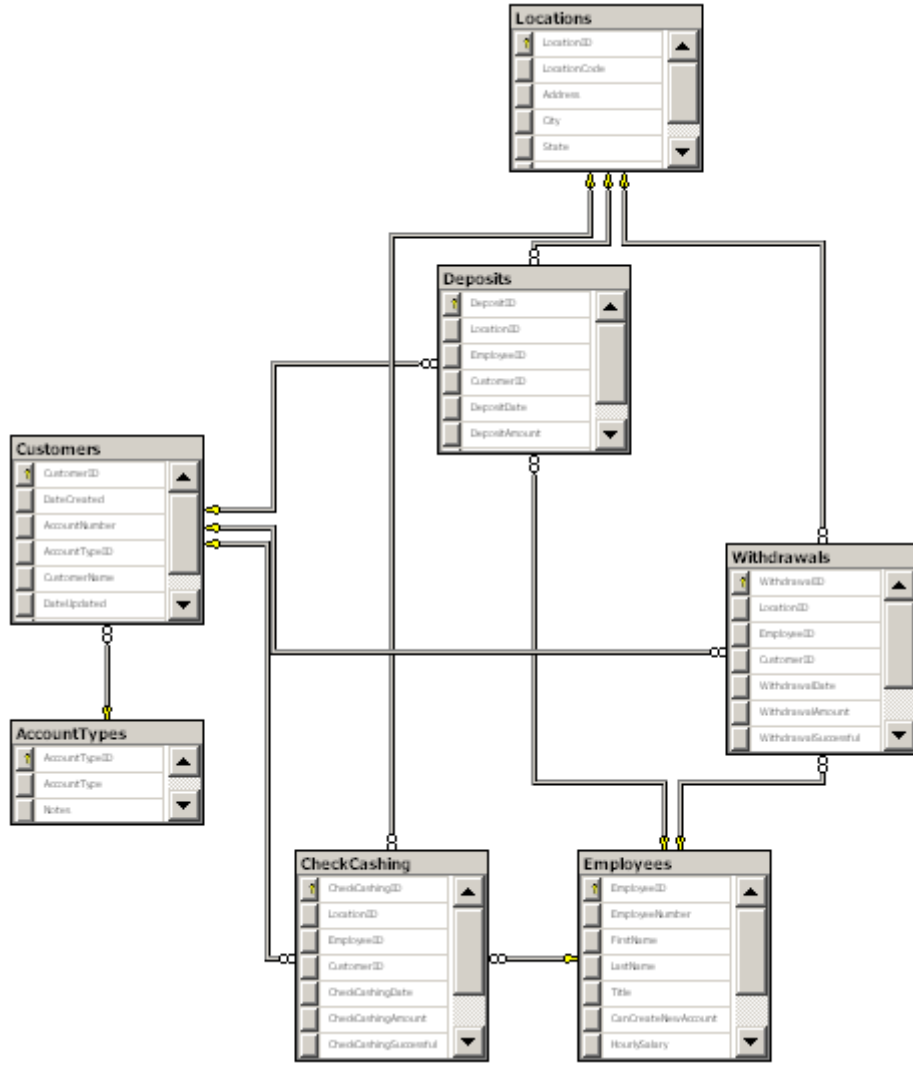
4- من مربع الحوار Add Table، حدد Customers وانقر على الزر > Add

5- انقر نقرا مزدوجا على CheckCashing لإضافته

6- بنفس الطريقة، قم بإضافة الجداول: AccountTypes، Deposits، Employees، Locations، و Withdrawals

7- من مربع الحوار Add Table، انقر على Close.

لاحظ أنه استنادا إلى الطريقة التي تم بها إنشاء قاعدة البيانات وكائناتها، تم إنشاء العلاقات:



8- لحفظ المخطط، انقر من شريط الأدوات على Save

9- أعط المخطط اسم dgmYugoNationalBank وانقر على OK، ثم أغلق واجهة التصميم.

التكامل المرجعي:

في قاعدة بيانات نموذجية يتم تدفق البيانات في اتجاهين إدخال وإخراج. في قاعدة بيانات البنك، يتم إنشاء حسابات العملاء أو حذفها على أساس منتظم. عندما يتم حذف حساب العميل، يحدث اضطراب بشأن المعاملات التي لها صلة بهذا الحساب. هنا يسمح التكامل المرجعي بإدارة العلاقات بين الجداول في قاعدة البيانات. يجب التحقق عندما يتم حذف بيانات من الجدول الأب، حيث يتم إخطار الجدول الابن بالسجلات المتعلقة بالبيانات المحذوفة لتحذف

هي الأخرى أيضا. عندما يتم تعديل بيانات على الجدول الأب، يتم تعديل البيانات المرتبطة بها في الجدول الابن.

لفرض التكامل المرجعي بين الجداول، يمكنك استخدام مربع الحوار Foreign Key Relationships. ويمكنك الوصول إلى هذا الأخير من واجهة تصميم الجدول أو من واجهة تصميم المخطط.

تطبيق عملي: فرض التكامل المرجعي

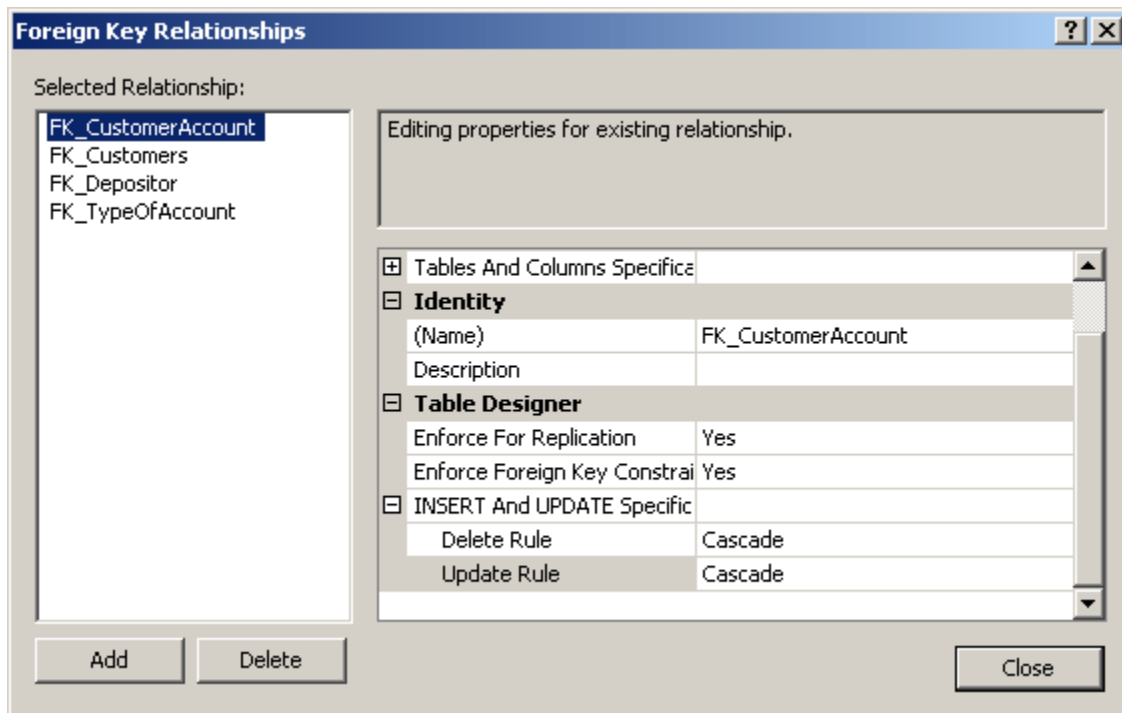
1- من متصفح الكائنات، قم بتوسيع البند YugoNationalBank. انقر بالزر الأيمن على dbo.Customers ثم اختر Design

2- انقر بالزر الأيمن في الجدول، واختر Relationships

3- حدد المفتاح الخارجي FK_CustomerAccount، ثم قم بتوسيع النطاق INSERT And UPDATE Specification في الجزء الأيمن

4- انقر على Delete Rule، ثم حدد في مربع الاختيارات Cascade

5- انقر على Update Rule، واختر من مربع الاختيارات Cascade:



6- بنفس الطريقة، اضبط الخيارات كما يلي:

Foreign Key	Delete Rule	Update Rule
FK_ChargeReasons	Cascade	Cascade
FK_Customers	Cascade	Cascade
FK_Depositor	Cascade	Cascade
FK_TypeOfAccount	Cascade	Cascade

7- انقر على Close

8- قم بحفظ وغلق الجدول

9- بنفس الطريقة، افتح الجدول Deposits في واجهة التصميم

10- افتح مربع الحوار Relationships

11- افتح المفتاح FK_Clerk لعرض خصائصه

اضبط Delete Rule و Update Rule على Cascade

12- قم بتنفيذ نفس الإجراءات للعلاقات الأخرى

13- أغلق مربع الحوار Relationships

14- احفظ الجدول وأغلقه.

ربط البيانات

عمليات الربط (Joins):

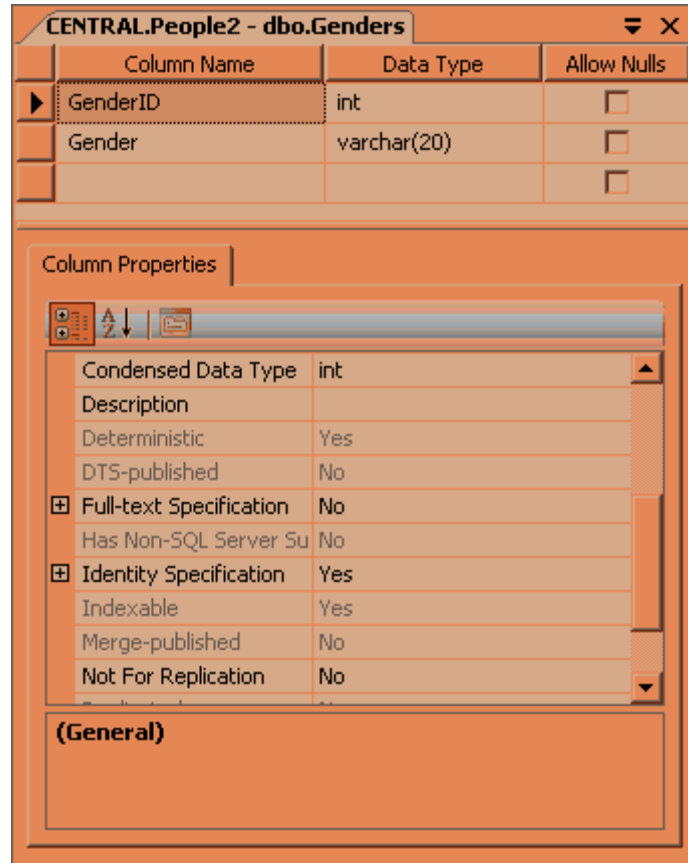
رأينا العلاقات بين الجداول وطريقة الربط بينها، ورأينا طريقة جعل البيانات في الجدول متاحة للجداول الأخرى، وهذا من شأنه أن يحد من الازدواجية في إدخال البيانات والأخطاء. هناك طريقة أخرى تتمثل في إنشاء مجموعة من الجداول بحيث تتركز السجلات في أكثر من جدول، وفي الأخير تجمع النتيجة في قائمة واحدة، وهذا هو أساس ربط البيانات.

ربط البيانات هو تقنية لإنشاء قائمة من السجلات تتكون من أكثر من جدول، وذلك باستخدام مجموعة من (أو جميع) الحقول من جداول مختلفة. نستنتج من ذلك أن ربط البيانات ينشأ أساساً عبر ثلاث خطوات:

- الجداول التي ستشارك في عملية الربط
- الحقول التي يتم بها إنشاء الارتباط في كل جدول
- الاستعلام الذي سينشئ السجلات.

جداول الربط:

قبل إنشاء الارتباط، لا بد من تحديد الجداول التي سيتم الربط بينها. سنحتاج لربط الجدول الأب إلى المفتاح الأساسي الذي سيستخدم كوصلة إلى جدول الابن. مثال لهذا النوع من الجداول:



يمكنك، إذا اقتضى الأمر، إنشاء بعض السجلات اللازمة لعملية الربط. مثال:

GenderID	Gender
1	Female
2	Male
3	Unknown
*	NULL

عند إنشائك للجدول الابن، لا تنسى إنشاء الحقل الذي يستخدم بمثابة صلة مع الجدول الأب. من الأحسن كما رأينا عند دراسة العلاقات، أن يستخدم نوع البيانات لهذا الحقل نفس نوع البيانات للمفتاح الأساسي من الجدول الأب. في هذا المثال الجدول الابن الذي سيتم ربطه إلى الجدول الأب أعلاه:

Column Name	Data Type	Allow Nulls
PersonID	int	<input type="checkbox"/>
FirstName	varchar(20)	<input checked="" type="checkbox"/>
LastName	varchar(20)	<input checked="" type="checkbox"/>
GenderID	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
Full-text Specification	No
Has Non-SQL Server Su	No
Identity Specification	Yes
Indexable	Yes
Merge-published	No
Not For Replication	No

(Name)

مرة أخرى، إذا لزم الأمر، يمكنك إضافة بعض السجلات اللازمة إلى هذا الجدول. مثال:

PersonID	FirstName	LastName	GenderID
1	Gertrude	Larson	1
2	Raymond	Kouma	NULL
3	Peter	Mukoko	2
4	Wally	Baston	2
5	Sylvia	Nguyen	NULL
6	Donald	Wallace	2
7	Hermine	Kana	1
8	Charlotte	Thomas	NULL
9	Paula	Barbers	NULL
10	Chrissie	Dentd	1
11	Ernestine	Essien	1
12	Barbara	Randt	NULL
13	Helene	Cranston	1
14	Robert	Palau	3
15	Paulette	Krazucky	1
16	Frank	Cranston	NULL
*	NULL	NULL	NULL

1 of 16

تطبيق عملي: التمهيد لربط البيانات

1- قم بتشغيل SQL Server Management Studio واتصل بالخادم.

2- من القائمة الرئيسية، انقر File -> New -> Query باستخدام الاتصال الحالي.

3- لإعداد قاعدة البيانات للتطبيقات المقبلة، اكتب الاستعلام التالي:

```
-- =====
-- Author:    FunctionX
-- Database:  RealEstate2
-- =====
IF EXISTS (
    SELECT *
        FROM sys.databases
        WHERE name = N'RealEstate2'
)
    DROP DATABASE RealEstate2
GO
CREATE DATABASE RealEstate2;
GO

-- =====
-- Author:    FunctionX
-- Database:  RealEstate2
-- Table:     PropertyTypes
-- =====
USE RealEstate2;
GO
CREATE TABLE PropertyTypes
(
    PropertyTypeID int identity(1,1) NOT NULL,
    PropertyType varchar(20)
);
GO
INSERT INTO PropertyTypes(PropertyType)
VALUES('Condominium');
GO
INSERT INTO PropertyTypes(PropertyType)
VALUES('Single Family');
GO
INSERT INTO PropertyTypes(PropertyType)
VALUES('Townhouse');
GO
INSERT INTO PropertyTypes(PropertyType)
VALUES('Unknown');
GO

-- =====
-- Author:    FunctionX
-- Database:  RealEstate2
-- Table:     Conditions
-- =====
USE RealEstate2;
GO
CREATE TABLE Conditions
```

```

(
    ConditionID int identity(1,1) NOT NULL,
    Condition varchar(20)
);
GO
INSERT INTO Conditions(Condition)
VALUES('Excellent');
GO
INSERT INTO Conditions(Condition)
VALUES('Good');
GO
INSERT INTO Conditions(Condition)
VALUES('Bad Shape');
GO
INSERT INTO Conditions(Condition)
VALUES('Mostly Damaged');
GO

-- =====
-- Author:    FunctionX
-- Database:  RealEstate2
-- Table:     Properties
-- =====
CREATE TABLE Properties
(
    PropertyID int identity(1,1) NOT NULL,
    PropertyNumber char(6),
    Address varchar(100),
    City varchar(50),
    State char(2),
    ZIPCode varchar(12),
    PropertyTypeID int,
    ConditionID int,
    Bedrooms smallint,
    Bathrooms float,
    FinishedBasement bit,
    IndoorGarage bit,
    Stories smallint,
    YearBuilt smallint,
    MarketValue money
);
GO

INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('524880', '1640 Lombardo Ave', 'Silver Spring', 'MD',
    '20904', 2, 2, 4, 2.5, 3, 1, 3, 1995, 495880.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('688364', '10315 North Hacht Rd', 'College Park', 'MD',
    '20747', 2, 1, 4, 3.5, 3,
    1, 2, 2000, 620724.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, FinishedBasement,
    Stories, MarketValue)
VALUES('611464', '6366 Lolita Drive', 'Laurel', 'MD',
    '20707', 2, 2, 1, 2, 422625.00);

```

```

GO
INSERT INTO Properties(Address, City, PropertyTypeID,
    Bedrooms, MarketValue)
VALUES('9002 Palasko Hwy', 'Tysons Corner',
    1, 2, 422895.00);
GO
INSERT INTO Properties(PropertyNumber, State,
    ZIPCode, Bedrooms, YearBuilt, MarketValue)
VALUES('420115', 'DC',
    '20011', 2, 1982, 312555);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
    PropertyTypeID, Bedrooms, YearBuilt, MarketValue)
VALUES('917203', 'Alexandria', '22024',
    2, 3, 1965, 345660.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    PropertyTypeID, ConditionID, Bedrooms, Bathrooms, MarketValue)
VALUES('200417', '4140 Holisto Crt', 'Germantown', 'MD',
    1, 1, 2, 1, 215495.00);
GO
INSERT INTO Properties(City, State, PropertyTypeID, ConditionID,
    Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES('Rockville', 'MD', 1, 2, 2, 2, 1996, 436885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('927474', '9522 Lockwood Rd', 'Chevy Chase', 'MD',
    '20852', 3, 3, 3, 2.5, 3, 0, 3,
    1992, 415665.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('207850', '14250 Parkdoll Rd', 'Rockville', 'MD',
    '20854', 3, 2, 3, 2.5, 2, 1, 2,
    1988, 325995.00);
GO
INSERT INTO Properties(City, PropertyTypeID, Bedrooms,
    YearBuilt, MarketValue)
VALUES('Washington', 3, 4, 1975, 366775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    YearBuilt, MarketValue)
VALUES('288540', '10340 Helmes Street #408', 'Silver Spring', 'MD',
    '20906', 1, 2, 1, 1, 2000, 242775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('247472', '1008 Copen Street', 'Silver Spring', 'MD',
    '20906', 2, 1,
    3, 3, 3, 1, 3, 1996, 625450.00);
GO
INSERT INTO Properties(City, ZIPCode, PropertyTypeID,
    Stories, YearBuilt, MarketValue)
VALUES('Chevy Chase', '20956', 2,
    3, 2001, 525450.00);
GO

```

```

INSERT INTO Properties(Address, City, State,
    PropertyTypeID, ConditionID, Bedrooms, MarketValue)
VALUES('686 Herod Ave #D04', 'Takoma Park', 'MD',
    1, 1, 2, 360885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297446', '14005 Sniders Blvd', 'Laurel', 'MD',
    '20707', 3, 4,
    4, 1.5, 3, 1, 2, 2002, 412885.00);
GO
INSERT INTO Properties(City, ZIPCode, ConditionID, Bedrooms,
    Stories, YearBuilt)
VALUES('Silver Spring', '20905', 2,
    4, 2, 1965);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('924792', '680 Prushia Rd', 'Washington', 'DC',
    '20008', 2, 2,
    5, 3.5, 3, 0, 3, 2000, 555885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('294796', '14688 Parrison Street', 'College Park', 'MD',
    '20742', 2, 1,
    5, 2.5, 2, 1, 2, 1995, 485995.00);
GO
INSERT INTO Properties(City, State, PropertyTypeID, ConditionID,
    Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES('Rockville', 'MD', 1, 2, 1, 1, 1996, 418885.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    YearBuilt, MarketValue)
VALUES('811155', '10340 Helmes Street #1012', 'Silver Spring',
    'MD', '20906', 1, 2,
    1, 1, 2000, 252775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('447597', '4201 Vilamar Ave', 'Hyattsville', 'MD',
    '20782', 3, 1,
    3, 2, 2, 1, 3, 1992, 365880.00);
GO
INSERT INTO Properties(Address, ZIPCode, Bathrooms)
VALUES('1622 Rombard Str', 20904, 2.5);
GO
INSERT INTO Properties(City, State, PropertyTypeID, ConditionID,
    Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES('Rockville', 'MD', 1, 2, 1, 1, 1996, 420555.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297415', '980 Phorwick Street', 'Washington', 'DC',
    '20004', 2, 2,

```

```

        4, 3.5, 3, 3, 1, 2004, 735475.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('475974', '9015 Marvin Crow Ave', 'Gaithersburg', 'MD',
    '20872', 2, 4,
    4, 2.5, 3, 1, 1, 1965, 615775.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('836642', '3016 Feldman Court', 'Rockville', 'MD',
    '20954', 2, 3,
    5, 3, 3, 1, 3, 1960, 528555.00);
GO
INSERT INTO Properties(Address, City, ZIPCode, PropertyTypeID,
    Bedrooms, Bathrooms, MarketValue)
VALUES('2444 Arielson Rd', 'Rockville', '20854', 1, 2, 1, 1996,
    475555.00);
GO

INSERT INTO Properties(City, State, PropertyTypeID, Stories)
VALUES('Rockville', 'MD',
    3, 1);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('208304', '7307 Everett Hwy', 'Washington', 'DC',
    '20012', 3, 1,
    2, 2.5, 2, 0, 4, 2006, 420550.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms,
    Bathrooms, YearBuilt, MarketValue)
VALUES('644114', '10340 Helmes Street#1006', 'Silver Spring',
    'MD', '20906', 1, 2,
    2, 2, 2000, 258445.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('937966', '7303 Warfield Court', 'Tysons Corner', 'VA',
    '22131', 2, 2,
    3, 2.5, 3, 1, 4, 2006, 825775.00);
GO
INSERT INTO Properties(City, ZIPCode, ConditionID, Bedrooms,
    Stories, YearBuilt)
VALUES('Fairfax', '22232', 2, 3, 3, 1985);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
    ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('297497', '12401 Conniard Ave', 'Takoma Park', 'MD',
    '20910', 3, 2,
    3, 2.5, 3, 1, 3, 2004, 280775.00);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
    PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
    YearBuilt, Stories, MarketValue)

```

```

VALUES('855255', 'Laurel', '20707', 2,
      4, 3, 2, 1962, 2, 342805.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('469750', '6124 Falk Rd', 'Arlington', 'VA',
      '22031', 2, 4,
      4, 3.5, 3, 1, 1, 1982, 635995.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('826927', '5121 Riehl Ace', 'Fairfax', 'VA',
      '22232', 3, 1,
      3, 1.5, 2, 0, 1, 2002, 325620.00);
GO
INSERT INTO Properties(City, ZIPCode, PropertyTypeID, Bedrooms,
      Bathrooms, MarketValue)
VALUES('Silver Spring', '20906', 1, 2, 2, 335655.00);
GO

INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('287064 ', '9533 Pensulian Rd', 'Silver Spring', 'MD',
      '20904', 2, 3,
      3, 1.5, 3, 1, 2, 1992, 485775.00);
GO
INSERT INTO Properties(PropertyNumber, City, ZIPCode,
      PropertyTypeID, ConditionID, Bedrooms, YearBuilt, Stories)
VALUES('724001 ', '705 Helios Ave', '20004',
      3, 3, 3, 1974, 4);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('209275', '944 Fryer Ave', 'Chevy Chase', 'MD',
      '20852', 2, 1,
      5, 2.5, 3, 0, 2, 2002, 625665.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('204759', '1950 Galego Street', 'Germantown', 'MD',
      '20874', 2, 1,
      4, 3.5, 2, 1, 4, 2007, 428665.00);
GO
INSERT INTO Properties(PropertyNumber, Address, City, State,
      ZIPCode, PropertyTypeID, ConditionID, Bedrooms, Bathrooms,
      FinishedBasement, IndoorGarage, Stories, YearBuilt, MarketValue)
VALUES('937259', '12366 Fowler Ave', 'Alexandria', 'VA',
      '22031', 3, 2,
      3, 1.5, 3, 1, 3, 2007, 402815.00);
GO


```

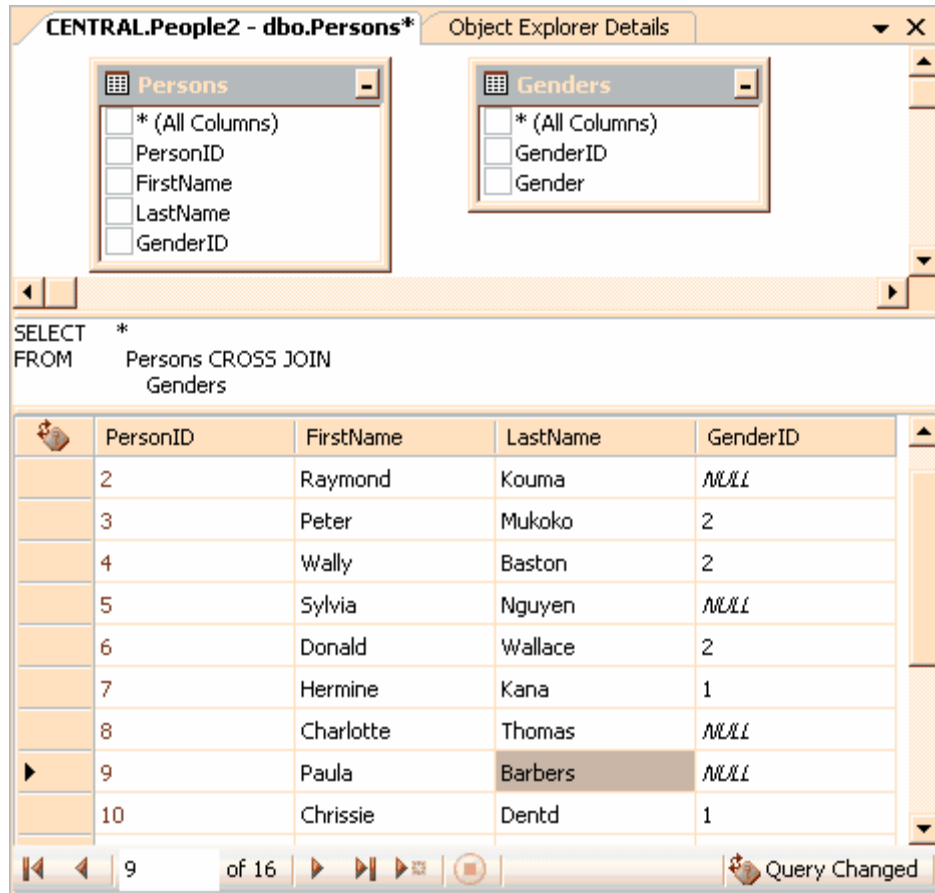
3- اضغط F5 لتنفيذ الاستعلام.

4- إحتفظ الملف باسم RealEstate2 وأغلق نافذة الاستعلام.

إنشاء الربط:

بعد تحديد الجداول والحقول اللازمة يمكنك إنشاء الارتباط. للقيام بذلك في SQL Server Management Studio يمكنك النقر بزر الأيمن على أحد الجداول المعنية بعملية الربط واختار Open Table. هذا الأخير من شأنه أن يعرض واجهة تصميم الاستعلام، قم بعرض نافذة المخطط والاستعلام. لإنشاء استعلام الربط لا بد من اختيار جدولين على الأقل، للقيام بذلك:

- من القائمة الرئيسية، يمكنك النقر على Query Designer -> Add Table
 - من شريط الأدوات Query Designer، يمكنك النقر على الزر  Add Table
 - يمكنك النقر بالزر الأيمن على المخطط، وانقر على Add table...
- بعد ذلك سيعرض مربع الحوار Add Table. لتحديد أي جدول يمكنك النقر مرتين أو تحديده ثم النقر على Add. في هذا المثال تم إضافة جدولين:



The screenshot shows the Query Designer window for a query named 'CENTRAL.People2 - dbo.Persons*'. The query is a cross join between the 'Persons' and 'Genders' tables. The result table displays the following data:

PersonID	FirstName	LastName	GenderID
2	Raymond	Kouma	NULL
3	Peter	Mukoko	2
4	Wally	Baston	2
5	Sylvia	Nguyen	NULL
6	Donald	Wallace	2
7	Hermine	Kana	1
8	Charlotte	Thomas	NULL
9	Paula	Barbers	NULL
10	Chrissie	Dentd	1

ستشاهد خطأ يربط الجدولين إذا تم إنشاء علاقة بينهما.

يتم إنشاء استعلام ربط جدولين باستخدام SQL وفقا للصيغة الأساسية التالية:

```
SELECT WhatColumn(s)
FROM ChildTable
TypeOfJoin ParentTable
ON Condition
```

يمثل ChildTable الجدول الذي يحتوي على السجلات التي سيتم استحضارها، ويمثل ParentTable الجدول الذي يحمل المفتاح الأساسي ويرتبط بالسجلات ذات الصلة بالجدول الابن.

عامل الشرط Condition هو تعبير منطقي يستخدم للتحقق من السجلات التي سيتم عزلها. لكتابة شرط ينبغي تعيين المفتاح الأساسي من الجدول الأب إلى المفتاح الخارجي من الجدول الابن. ولأن كلا هذين الحقليين يحمل نفس الاسم، يجب التمييز بينها بإضافة اسم الجدول قبل اسم الحقل متبوعا بنقطة.

أخيرا WhatColumn(s) يمثل الحقل (أو الحقول) التي يشملها الاستعلام، ويمكنك هنا استخدام الرمز * لتحديد كافة الحقول من كل الجداول، ويمكنك بدلا من استحضار كل الحقول إنشاء قائمة بأسماء من الحقول حيث تفصل بينها بفواصل. يمكنك كتابة اسم الحقل مباشرة إذا كان هذا الاسم لا يتكرر في جداول أخرى، مثال:

```
SELECT LastName, FirstName, Gender
FROM Persons
TypeOfJoin Genders
ON Persons.GenderID = Genders.GenderID
```

أما إذا كان أحد الحقول يحمل نفس الاسم من جدول آخر، كما هو الحال بالنسبة للمفاتيح الخارجية، فينبغي التمييز في قائمة الحقول. مثال:

```
SELECT LastName, FirstName, Persons.GenderID,
Genders.GenderID, Gender
FROM Persons
TypeOfJoin Genders
ON Persons.GenderID = Genders.GenderID
```

من أجل تسهيل القراءة البرمجية للاستعلام، يفضل تمييز كل الحقول. مثال:

```
SELECT Persons.LastName, Persons.FirstName, Persons.GenderID,
Genders.GenderID, Genders.Gender
FROM Persons
TypeOfJoin Genders
```


ON Persons.GenderID = Genders.GenderID

TypeOfJoin يمثل نوع الارتباط، سيأتي شرحه.


تطبيق عملي: إعداد عملية الربط

1- من متصفح الكائنات، انقر بالزر الأيمن على Databases، واختر Refresh

2- قم بتوسيع Databases ثم RealEstate2

3- قم بتوسيع البند Tables

4- ثم بالزر الأيمن انقر على Properties ثم انقر على Open Table

5- من شريط الأدوات Query Designer، انقر على الزر  Show Diagram Pane

والزر Show SQL Pane

6- من شريط أدوات Query Designer، انقر على  Add Table

7- في مربع الحوار Add Table، انقر نقرا مزدوجا على PropertyTypes

8- انقر على Close.

الربط الداخلي والمتقاطع (Cross and Inner Joins):

عند دراستنا للعلاقات، رأينا أن دور المفاتيح الأساسية والخارجية يتمثل في الحفاظ على تبادل المعلومات بين جدولين. هذه التقنية تلعب دورا رئيسيا عند إنشاء الربط. فهي تسمح بتحديد السجلات التي ترغب بإدراجها أثناء إنشاء عملية الربط. من أجل احترام اتجاه العلاقة بين الجدولين يدعم SQL ثلاثة أنواع من الربط.

الربط المتقاطع (Cross Joins):

يقوم الربط المتقاطع بإنشاء قائمة لجميع السجلات من كلا الجدولين على النحو التالي:
يرتبط السجل الأول من الجدول الأب بكل سجل من الجدول الابن، ثم سجل الثاني من الجدول

الأب مع كل سجل من الجدول الابن، وهكذا.... في هذه الحالة أيضا، ليست هناك حاجة لحقول مشتركة بين كل الجداول. بعبارة أخرى لا تستخدم تعبير ON.

لإنشاء استعمال الربط المتقاطع، يمكنك استبدال العامل *TypeOfJoin* في الصيغة السابقة بإحدى العبارات CROSS JOIN أو CROSS OUTER JOIN. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
       Genders.GenderID, Genders.Gender
FROM Persons
CROSS JOIN Genders
GO
```

بشكل افتراضي، يقوم SQL Server Management Studio بعد إضافة جدولين (إذا لم تكن هناك علاقة بينهما) بإنشاء استعمال الربط المتقاطع تلقائيا. كل ما عليك فعله هو اختيار الحقول المطلوبة ثم تنفيذ الاستعلام لمشاهدة النتيجة:

PersonID	FirstName	LastName	GenderID	Gender ID	Gender
12	Barbara	Randt	NULL	1	Female
13	Helene	Cranston	1	1	Female
14	Robert	Palau	3	1	Female
15	Paulette	Krazucky	1	1	Female
16	Frank	Cranston	NULL	1	Female
1	Gertrude	Larson	1	2	Male
2	Raymond	Kouma	NULL	2	Male
3	Peter	Mukoko	2	2	Male
4	Wally	Baston	2	2	Male
5	Sylvia	Nguyen	NULL	2	Male

الربط الداخلي (Inner Join):

تخيل أن لديك جدولين يمكن ربطهما من خلال مفتاح أساسي في جدول ومفتاح خارجي من جدول آخر.

CENTRAL.People2 - dbo.Genders	
GenderID	Gender
1	Female
2	Male
3	Unknown
*	NULL

CENTRAL.People2 - dbo.Persons				
PersonID	FirstName	LastName	GenderID	
1	Gertrude	Larson	1	
2	Raymond	Kouma	NULL	
3	Peter	Mukoko	2	
4	Wally	Baston	2	
5	Sylvia	Nguyen	NULL	
6	Donald	Wallace	2	
7	Hermine	Kana	1	
8	Charlotte	Thomas	NULL	
9	Paula	Barbers	NULL	
10	Chrissie	Dentd	1	
11	Ernestine	Essien	1	
12	Barbara	Randt	NULL	
13	Helene	Cranston	1	
14	Robert	Palau	3	
15	Paulette	Krazucky	1	
16	Frank	Cranston	NULL	
*	NULL	NULL	NULL	NULL

لاحظ أن بعض السجلات في الجدول Persons ليس لها قيمة في الحقل GenderID وهي تحمل عبارة NULL من قبل خادم قاعدة البيانات. عند إنشاء استعلام الربط لسجلات الجدول Persons، وإذا كنت تريد أن ضم السجلات التي تحمل قيمة، يمكنك إنشاؤها باستخدام الربط الداخلي.

افتراضيا، يقوم SQL Server Management Studio عند إنشاء استعلام جديد أثناء تصميم الاستعلام وإضافة جدولين، وإذا وجدت علاقة بين الجدولين، بإنشاء استعلام الربط الداخلي. إذا لم تكن هناك علاقة بين الجدولين، فعليك تحرير استعلام الربط الداخلي يدويا. مثال:

The screenshot shows the SQL Server Management Studio interface. The query window contains the following SQL code:

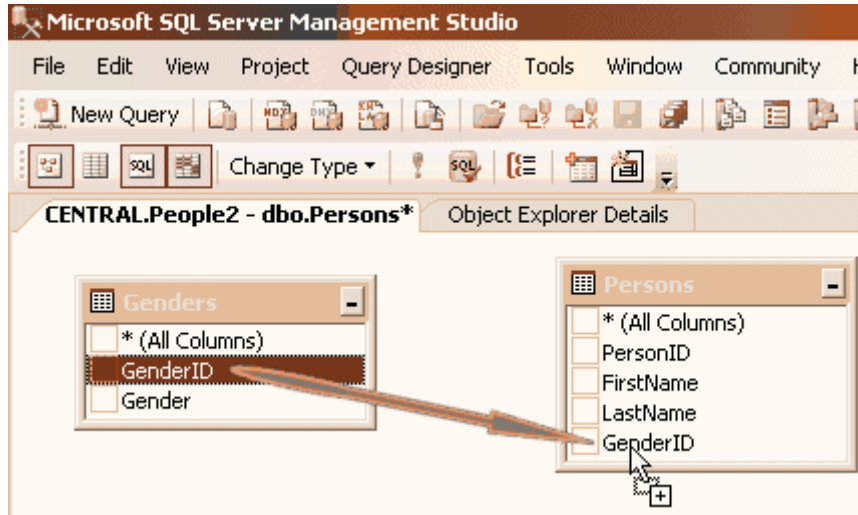
```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName, Persons.GenderID, Ge
FROM Persons CROSS JOIN
```

The results grid displays the following data:

	PersonID	FirstName	LastName	GenderID	Gender ID	Gender
	12	Barbara	Randt	NULL	1	Female
	13	Helene	Cranston	1	1	Female
	14	Robert	Palau	3	1	Female
	15	Paulette	Krazucky	1	1	Female
	16	Frank	Cranston	NULL	1	Female
	1	Gertrude	Larson	1	2	Male
	2	Raymond	Kouma	NULL	2	Male
	3	Peter	Mukoko	2	2	Male
	4	Wally	Baston	2	2	Male
	5	Sylvia	Nguyen	NULL	2	Male

لاحظ أن لا علاقة تربط الجدولين، لذلك تم إنشاء استعلام الربط المتداخل (المتقاطع).

هناك خيارين لإنشاء ربط داخلي، يمكنك سحب المفتاح الأساسي من الجدول الأب ثم إسقاطه على المفتاح الخارجي في الجدول الابن. مثال:



كبدل عن ذلك، يمكنك تحرير الاستعلام يدويا لإنشاء الربط الداخلي، حيث يتم تحديد عامل *TypeOfJoin* لصيغتنا العامة بعبارة INNER JOIN. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
Persons.GenderID,
Genders.GenderID AS [Gender ID], Genders.Gender
FROM Persons INNER JOIN Genders ON Persons.GenderID =
Genders.GenderID
```

بعد إنشاء الارتباط، سيظهر في الرسم التخطيطي خط يصل الجدولين. بعد تنفيذ الاستعلام

نشاهد هذه النتيجة:

CENTRAL.People2 - dbo.Persons* Object Explorer Details

Genders

- * (All Columns)
- GenderID
- Gender

Persons

- * (All Columns)
- PersonID
- FirstName
- LastName
- GenderID

```

SELECT Persons.PersonID, Persons.FirstName, Persons.LastName, Persons.GenderID,
Genders.GenderID AS [Gender ID], Genders.Gender
FROM Persons INNER JOIN Genders ON Persons.GenderID = Genders.GenderID

```

	PersonID	FirstName	LastName	GenderID	Gender ID	Gender
▶	1	Gertrude	Larson	1	1	Female
	3	Peter	Mukoko	2	2	Male
	4	Wally	Baston	2	2	Male
	6	Donald	Wallace	2	2	Male
	7	Hermine	Kana	1	1	Female
	10	Chrissie	Dentd	1	1	Female
	11	Ernestine	Essien	1	1	Female
	13	Helene	Cranston	1	1	Female
	14	Robert	Palau	3	3	Unknown
	15	Paulette	Krazucky	1	1	Female

1 of 10 Cell is Read Only.

ذكرنا سابقا أنه بإمكانك إدراج جميع الحقول من كلا الجدولين في الاستعلام. في مثالنا هذا لسنا بحاجة إلى الحقل GenderID من الجدول Genders. مثال:

The screenshot displays the Object Explorer Details window for the query. The query text is as follows:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName, Genders.Gender
FROM Persons INNER JOIN
      Genders ON Persons.GenderID = Genders.GenderID
```

PersonID	FirstName	LastName	Gender
1	Gertrude	Larson	Female
3	Peter	Mukoko	Male
4	Wally	Baston	Male
6	Donald	Wallace	Male
7	Hermine	Kana	Female
10	Chrissie	Dentd	Female
11	Ernestine	Essien	Female
13	Helene	Cranston	Female
14	Robert	Palau	Unknown
15	Paulette	Krazucky	Female

لاحظ أن الناتج يتضمن فقط السجلات التي تحمل بيانات (غير المعدومة) في الحقل GenderID مفتاح خارجي من جدول الأشخاص.

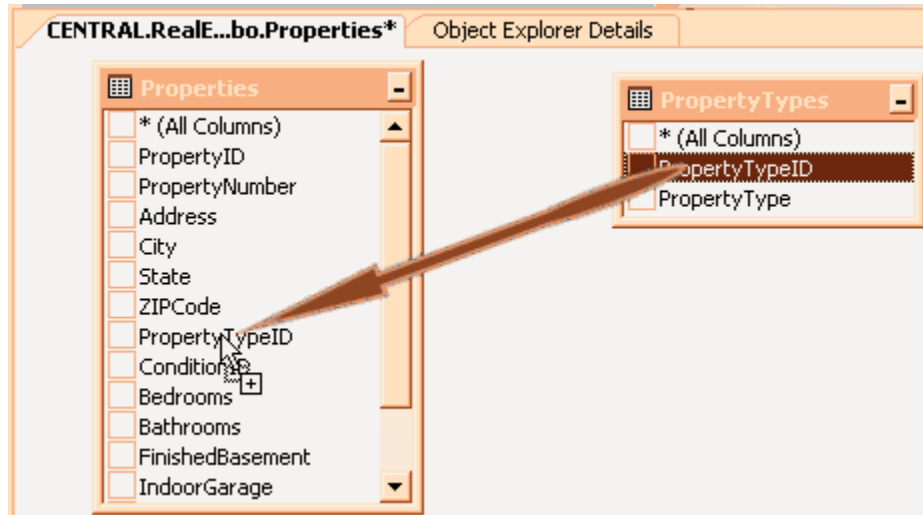
يمكن كبديل عن INNER JOIN الاكتفاء بكتابة JOIN فقط. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
      Genders.Gender
FROM Persons
JOIN Genders
ON Persons.GenderID = Genders.GenderID
GO
```

لحذف الربط بين الجدولين، إذا كنت تعمل في إطار تصميم الاستعلام، يمكنك النقر بزر الأيمن على الخط الذي يربط الجدولين في الرسم التخطيطي ثم انقر على Remove. باستخدام SQL، يجب التعديل على عبارات الربط JOIN والشرط ON.

تطبيق عملي: إنشاء الربط الداخلي

1- قم بإنشاء ربط داخلي بسحب الحقل PropertyTypeID من الجدول PropertyTypes، ثم إسقاطه على الحقل PropertyTypeID من الجدول Properties:



2- من الجداول حدد الحقول التالية: PropertyType, City, Bedrooms, Bathrooms, YearBuilt, and MarketValue

3- من شريط أدوات مصمم الاستعلام، انقر على زر التنفيذ لمشاهدة النتيجة.

The screenshot shows Microsoft SQL Server Management Studio with a query window titled 'CENTRAL.RealE...bo.Properties*'. The query is as follows:

```
SELECT PropertyTypes.PropertyType, Properties.City, Properties.Bedrooms, Properties.Bathrooms, Properties.Y
FROM Properties INNER JOIN
PropertyTypes ON Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
```

The results table shows the following data:

PropertyType	City	Bedrooms	Bathrooms	YearBuilt	MarketValue
Single Family	Silver Spring	4	2.5	1995	495880.0000
Single Family	College Park	4	3.5	2000	620724.0000
Single Family	Laurel	NULL	NULL	NULL	422625.0000
Condominium	Tysons Corner	2	NULL	NULL	422895.0000
Single Family	Alexandria	3	NULL	1965	345660.0000
Condominium	Germantown	2	1	NULL	215495.0000
Condominium	Rockville	2	2	1996	436885.0000
Townhouse	Chevy Chase	3	2.5	1992	415665.0000
Townhouse	Rockville	3	2.5	1988	325995.0000
Townhouse	Washington	4	NULL	1975	366775.0000

الربط الخارجي OUTER JOIN :

بدلاً من عرض فقط السجلات التي تحتوي قيم في الجدول الابن، قد ترغب في استعلام يعرض كافة السجلات، بما فيها السجلات التي لا تحمل أية قيمة. للحصول على هذه النتيجة قم بإنشاء ارتباط خارجي. وهناك ثلاث خيارات في الربط الخارجي:

الربط الخارجي من اليسار:

يضم الربط الخارجي من اليسار جميع السجلات من الجدول الابن. حيث تكتب العبارة *NULL* في سجلات المفتاح الخارجي للجدول الابن التي لا تحمل أي قيمة.

لإنشاء ارتباط خارجي من اليسار في قسم الرسم التخطيطي، انقر بالزر الأيمن على الخط الذي يربط الجدولين ثم انقر على الخيار *Select All Rows from ...* الذي من شأنه تحديد كافة السجلات من الجدول الابن (في هذا المثال، يتم تحديد كافة السجلات من الجدول *Persons*):

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer Details pane displays two tables: *Genders* and *Persons*. The *Persons* table is selected, and a context menu is open over it, with the option *Select All Rows from Persons* highlighted. The SQL query window shows the following query:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName, Genders.GenderID, Genders.Gender
FROM Persons INNER JOIN Genders ON Persons.GenderID = Genders.GenderID
```

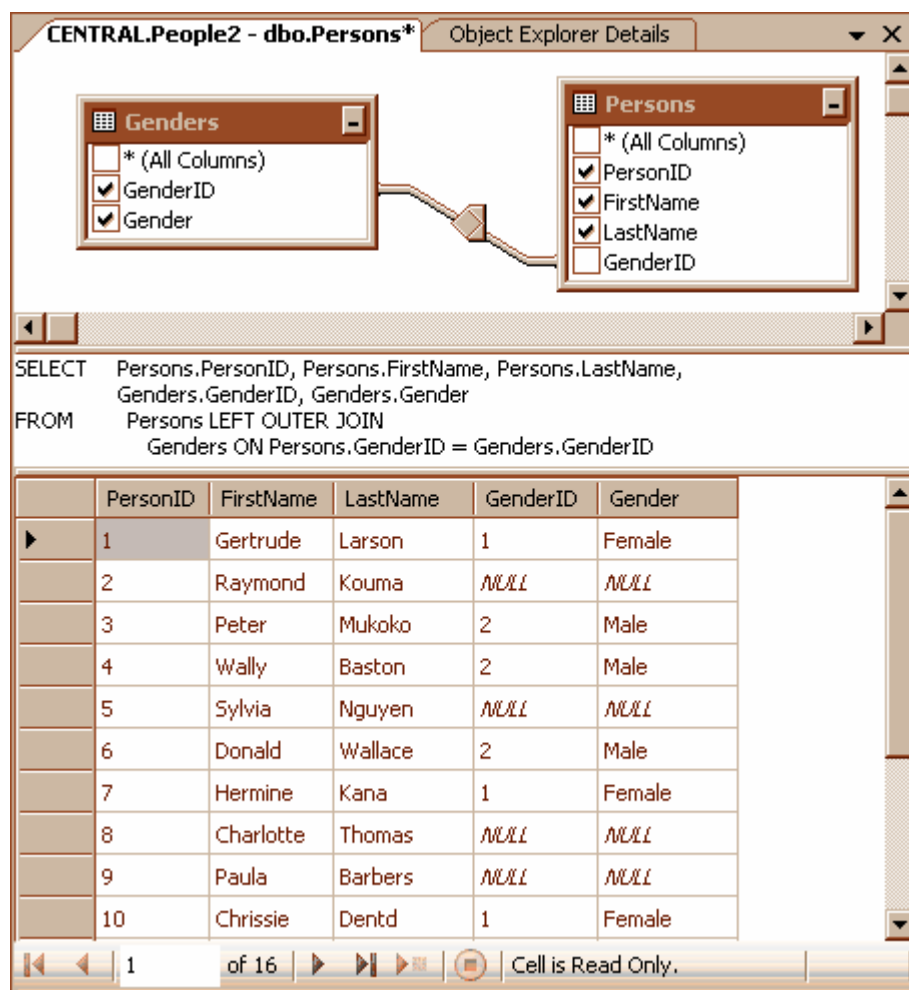
The Results pane displays the following data:

PersonID	FirstName	LastName	GenderID	Gender
1	Gertrude	Larson	1	Female
2	Raymond	Kouma	NULL	NULL
3	Peter	Mukoko	2	Male
4	Wally	Baston	2	Male
5	Sylvia	Nguyen	NULL	NULL
6	Donald	Wallace	2	Male
7	Hermine	Kana	1	Female
8	Charlotte	Thomas	NULL	NULL
9	Paula	Barbers	NULL	NULL
10	Chriccie	Dentrd	1	Female

بدلاً من ذلك، يمكنك استبدال العامل *TypeOfJoin* في صيغتنا إما بالعبارة LEFT JOIN أو LEFT OUTER JOIN. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
       Genders.GenderID, Genders.Gender
FROM Persons
LEFT OUTER JOIN Genders
ON Persons.GenderID = Genders.GenderID
GO
```

في كلتا الطريقتين، سيتم إضافة سهم إلى الزر الموجود في الخط الذي يربط الجدولين في الرسم التخطيطي، لاحظ المثال:



لاحظ أن النتيجة تشمل جميع السجلات من الجدول Persons (الجدول الأيمن)، والسجلات التي لا تحمل أية قيمة في الحقل GenderID من الجدول Persons توضع عليها عبارة NULL. الربط الخارجي من اليمين:

يضم الربط الخارجي من اليمين جميع السجلات من الجدول الأب، ويبحث عن السجلات المطابقة في الجدول الابن.

قبل إنشاء ارتباط خارجي من جهة اليمين، يجب إزالة الربط الخارجي من اليسار، إذا تم إنشاؤه مسبقاً، وذلك عن طريق النقر بالزر الأيمن على الخط الفاصل بين الجدولين واختيار الخيار الثاني تحت Remove (في مثالنا Select All Rows from Persons). بعد ذلك، يمكنك النقر بالزر الأيمن على الخط الذي يربط بينها وانقر على الخيار الذي من شأنه تحديد كافة السجلات من الجدول الأب. في هذا المثال نحدد Select All Rows From Genders.

لإنشاء ربط خارجي من جهة اليمين باستخدام SQL، يمكنك استبدال العامل *TypeOfJoin* في الصيغة السابقة بالعباراة RIGHT JOIN أو RIGHT OUTER JOIN. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName ,  
        Genders.GenderID, Genders.Gender  
FROM Persons  
RIGHT OUTER JOIN Genders  
ON Persons.GenderID = Genders.GenderID  
GO
```

سيظهر باستخدام أي من الطريقتين السابقتين، سهم على الخط الذي يربط الجدولين، حيث يشير هذا السهم إلى الجدول الابن. مثال:

The screenshot shows a SQL query window with the following SQL code:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
       Genders.GenderID, Genders.Gender
FROM Persons RIGHT OUTER JOIN
     Genders ON Persons.GenderID = Genders.GenderID
```


The results table is as follows:

PersonID	FirstName	LastName	GenderID	Gender
1	Gertrude	Larson	1	Female
7	Hermine	Kana	1	Female
10	Chrissie	Dentd	1	Female
11	Ernestine	Essien	1	Female
13	Helene	Cranston	1	Female
15	Paulette	Krazucky	1	Female
3	Peter	Mukoko	2	Male
4	Wally	Baston	2	Male
6	Donald	Wallace	2	Male
14	Robert	Palau	3	Unknown

لاحظ أن ناتج الاستعلام يبدأ بعرض السجل الأول من الجدول الأب (في هذه الحالة الجدول Genders)، ثم سجلات الجدول الابن، التي لديها مقابل للسجل الأول. ثم ينتقل إلى قيمة السجل الموالي للحقل GenderID وهكذا.... لاحظ أيضا أنه لا توجد سجلات فارغة في الجدول Genders.

تطبيق عملي: الحصول على السجلات غير المعدومة

1- للحصول على قائمة الممتلكات معروفة النوع، انقر بالزر الأيمن على الرابط بين الجدولين وقم بتحديد كافة السجلات من PropertyTypes.

2- من شريط الأدوات مصمم الاستعلام، انقر على زر التنفيذ  Execute

Object Explorer Details

Properties

- City
- State
- ZIPCode
- PropertyTypeID
- ConditionID
- Bedrooms
- Bathrooms
- FinishedBasement
- IndoorGarage
- Stories
- YearBuilt
- MarketValue

PropertyTypes

- *(All Columns)
- PropertyTypeID
- PropertyType

```
SELECT PropertyTypes.PropertyType, Properties.City, Properties.Bedrooms, Properties.Bathrooms, Properties.Y
FROM Properties RIGHT OUTER JOIN
PropertyTypes ON Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
```

PropertyType	City	Bedrooms	Bathrooms	YearBuilt	MarketValue
Condominium	Rockville	1	1	1996	420555.0000
Condominium	Rockville	2	1	NULL	475555.0000
Condominium	Silver Spring	2	2	2000	258445.0000
Condominium	Silver Spring	2	2	NULL	335655.0000
Single Family	Silver Spring	4	2.5	1995	495880.0000
Single Family	College Park	4	3.5	2000	620724.0000
Single Family	Laurel	NULL	NULL	NULL	422625.0000
Single Family	Alexandria	3	NULL	1965	345660.0000
Single Family	Silver Spring	3	3	1996	625450.0000
Single Family	Chev Chase	NULL	NULL	2001	525450.0000

1 of 40 Cell is Read Only.

3- لاحظ أن النتيجة هي قائمة الممتلكات مرتبة حسب الأنواع (condos, single families,

town homes

الربط الخارجي الكامل:

يستخدم استعمال الربط الخارجي الكامل لعرض كافة السجلات من كلا الجدولين، حيث إذا

لم يكن لسجل من أحد الجدولين قيمة في الجدول الآخر فسيتم فيه عرض عبارة NULL.

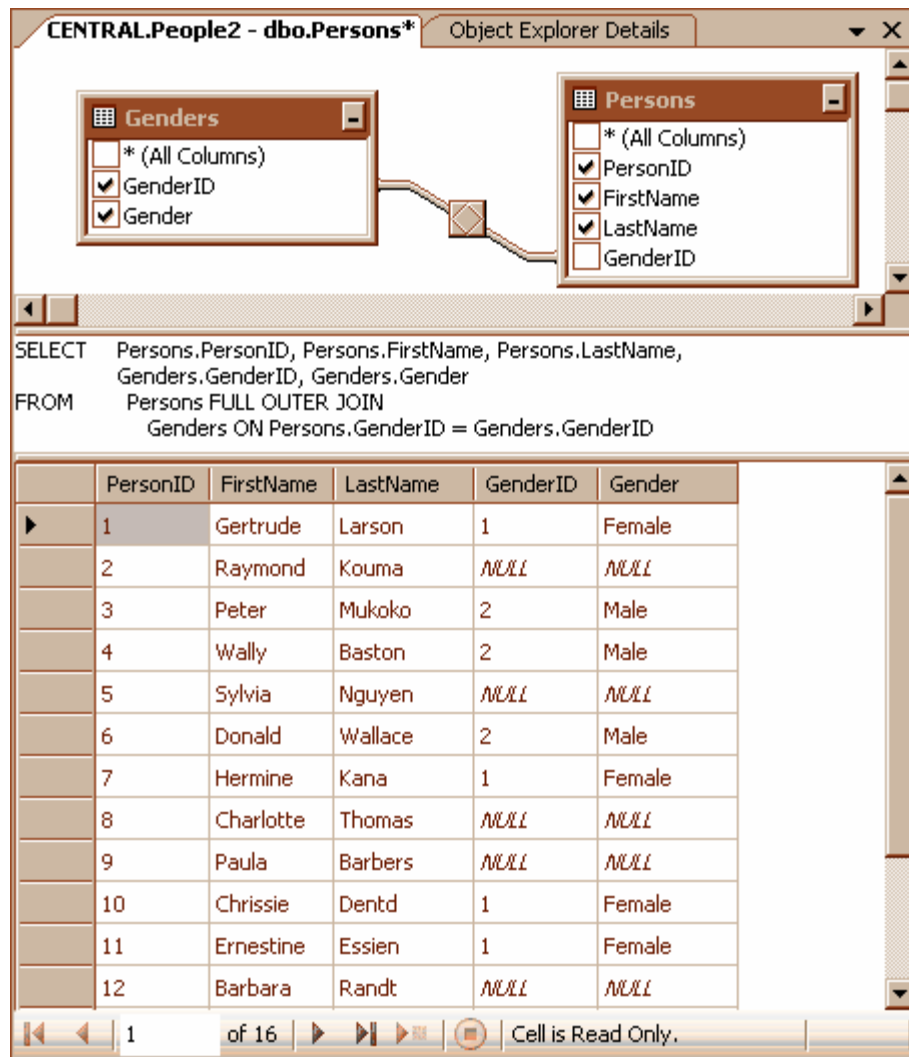
لإنشاء ارتباط خارجي كامل في واجهة تصميم الاستعلام، انقر بالزر الأيمن على الخط

الرابط بين الجدولين، وحدد كلا الخيارين تحت الخيار Remove.

لإنشاء ربط متكامل باستخدام SQL، قم باستبدال العامل *TypeOfJoin* من الصيغة بعبارة FULL JOIN أو FULL OUTER JOIN. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
       Genders.GenderID, Genders.Gender
FROM Persons
FULL OUTER JOIN Genders
ON Persons.GenderID = Genders.GenderID
GO
```

بعد ذلك، سيظهر مربع على الزر الرابط بين الجدولين. وبتنفيذ الاستعلام نشاهد:



The screenshot displays the Object Explorer Details window for the 'CENTRAL.People2 - dbo.Persons*' database. It shows a diagram of a FULL OUTER JOIN between the 'Persons' and 'Genders' tables. Below the diagram, the SQL query is shown, and the results of the query are displayed in a table.

PersonID	FirstName	LastName	GenderID	Gender
1	Gertrude	Larson	1	Female
2	Raymond	Kouma	NULL	NULL
3	Peter	Mukoko	2	Male
4	Wally	Baston	2	Male
5	Sylvia	Nguyen	NULL	NULL
6	Donald	Wallace	2	Male
7	Hermine	Kana	1	Female
8	Charlotte	Thomas	NULL	NULL
9	Paula	Barbers	NULL	NULL
10	Chrissie	Dentd	1	Female
11	Ernestine	Essien	1	Female
12	Barbara	Randt	NULL	NULL

تماما كما رأينا في طرق الربط بين جدولين حتى الآن، يمكنك إنشاء استعلام الربط الذي يشمل العديد من الجداول.

الارتباط وتحليل البيانات:

رأينا أن من أهم مميزات الاستعلام إمكانية عزل السجلات، ويتم ذلك باستخدام الشروط والمعايير. ويعزز الربط هذه الإمكانية لأنه يسمح بعرض السجلات من جداول مختلفة، وإدراجها في نفس الاستعلام.

في استعلامات الربط التي قمنا بها حتى الآن، عرضنا جميع السجلات وتركنا محرك قاعدة البيانات يتولى الباقي باستخدام قواعد الربط في SQL. للتحكم أكثر في جمع وعرض البيانات، يمكنك إدراج شروط (أو معايير) يتم وفقها عزل السجلات غير المرغوب فيها كما رأينا في الدروس السابقة، ليشمل الاستعلام SELECT شرطاً، يمكنك إدراج عبارة الشرط .WHERE

استخدام عبارة الشرط WHERE:

لإضافة معايير إلى الاستعلام اكتب الشرط في المربع المقابل للعمود Filter باستخدام أحد العبارات التي استعرضنا في الدروس السابقة. مثال:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
       Genders.GenderID, Genders.Gender
FROM   Persons LEFT OUTER JOIN
       Genders ON Persons.GenderID = Genders.GenderID
WHERE  Genders.Gender = 'female'
```

وهذا من شأنه أن ينتج:

The screenshot shows the SQL Server Enterprise Manager interface. At the top, there are two panes: 'Genders' and 'Persons'. The 'Genders' pane shows columns: * (All Columns), GenderID, and Gender. The 'Persons' pane shows columns: * (All Columns), PersonID, FirstName, LastName, and GenderID. Below these panes is a table with columns: Column, Table, Output, Filter, and Or... The table contains the following data:

Column	Table	Output	Filter	Or...
PersonID	Persons	<input checked="" type="checkbox"/>		
FirstName	Persons	<input checked="" type="checkbox"/>		
LastName	Persons	<input checked="" type="checkbox"/>		
GenderID	Genders	<input checked="" type="checkbox"/>		
Gender	Genders	<input checked="" type="checkbox"/>	= 'female'	

Below the table is the SQL query:

```
SELECT Persons.PersonID, Persons.FirstName, Persons.LastName,
Genders.GenderID, Genders.Gender
FROM Persons LEFT OUTER JOIN
Genders ON Persons.GenderID = Genders.GenderID
WHERE (Genders.Gender = 'female')
```

At the bottom, there is a result grid with the following data:


PersonID	FirstName	LastName	GenderID	Gender
1	Gertrude	Larson	1	Female
7	Hermine	Kana	1	Female
10	Chrissie	Dentd	1	Female
11	Ernestine	Essien	1	Female
13	Helene	Cranston	1	Female
15	Paulette	Krazucky	1	Female

The status bar at the bottom indicates '1 of 6' and 'Cell is Read Only'.

تطبيق عملي: تحليل وربط البيانات

1- للاطلاع على قائمة الممتلكات من النوع townhouses، غير الاستعلام على النحو:

```
SELECT PropertyTypes.PropertyType, Properties.City,
Properties.State, Properties.ZIPCode, Properties.Bedrooms,
Properties.Bathrooms, Properties.Stories,
Properties.MarketValue
FROM Properties RIGHT OUTER JOIN PropertyTypes
ON Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE (PropertyTypes.PropertyTypeID = 3)
```

2- من شريط الأدوات تصميم الاستعلام، انقر على زر التنفيذ 

3- للحصول على قائمة الممتلكات من النوعين townhouses و single families، غير

الاستعلام على النحو التالي:

```

SELECT PropertyTypes.PropertyType, Properties.MarketValue,
       Properties.City, Properties.State, Properties.Bedrooms,
       Properties.YearBuilt, Properties.Bathrooms,
       Properties.FinishedBasement, Properties.Stories
FROM   Properties RIGHT OUTER JOIN PropertyTypes
ON     Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE  (PropertyTypes.PropertyTypeID IN (2, 3))

```



4- من شريط الأدوات تصميم الاستعلام ، انقر على زر التنفيذ

5- للحصول على منازل العائلة مرتبة وفق ترتيب زمني بدءا من الأحدث، قم بتغيير الاستعلام على النحو التالي:

```

SELECT PropertyTypes.PropertyType,
       Properties.City, Properties.State, Properties.ZIPCode,
       Properties.Bedrooms, Properties.Bathrooms, Properties.Stories,
       Properties.YearBuilt, Properties.MarketValue
FROM   Properties RIGHT OUTER JOIN PropertyTypes
ON     Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE  (PropertyTypes.PropertyTypeID = 2)
ORDER BY Properties.YearBuilt DESC

```

6- انقر بالزر الأيمن على الجدول واختر Execute SQL

7- للحصول على قائمة الممتلكات التي تكلف ما بين \$ 350,000 و \$ 475,000، أكتب الاستعلام التالي:

```

SELECT Properties.PropertyNumber, PropertyTypes.PropertyType,
       Properties.MarketValue, Properties.City, Properties.State,
       Properties.Bedrooms, Properties.YearBuilt
FROM   Properties RIGHT OUTER JOIN PropertyTypes
ON     Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE  (Properties.MarketValue BETWEEN 350000 AND 475000)

```

8- قم بتنفيذ الاستعلام.

9- للحصول على قائمة المنازل لعائلة واحدة في ولاية Virginia وفق الترتيب الزمني، حيث رقم العقار معروف، قم بتغيير الاستعلام على النحو التالي:

```

SELECT Properties.PropertyNumber,
       PropertyTypes.PropertyType, Properties.MarketValue,
       Properties.City,
       Properties.State, Properties.Bedrooms,
       Properties.FinishedBasement, Properties.YearBuilt
FROM   Properties RIGHT OUTER JOIN PropertyTypes
ON     Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE  (PropertyTypes.PropertyTypeID = 2) AND
       (Properties.PropertyNumber IS NOT NULL) AND
       (Properties.State = 'VA')
ORDER BY Properties.YearBuilt DESC

```

10- من شريط الأدوات تصميم الاستعلام، انقر على زر التنفيذ.

11- للحصول على قائمة العقارات في ولاية Maryland الجنوبية حيث مبلغ العقار أقل من 400,000 دولار، أكتب الاستعلام التالي:

```
SELECT Properties.PropertyNumber, PropertyTypes.PropertyType,
       Properties.MarketValue, Properties.City, Properties.State,
       Properties.Bedrooms, Properties.YearBuilt
FROM   Properties RIGHT OUTER JOIN PropertyTypes
ON     Properties.PropertyTypeID = PropertyTypes.PropertyTypeID
WHERE  (Properties.MarketValue < 400000) AND
       (Properties.ZIPCode BETWEEN '20500' AND '21000')
```

12- من شريط الأدوات تصميم الاستعلام، انقر على زر التنفيذ.

الإجراءات المخزنة

أساسيات حول الإجراءات المخزنة:

تحدثنا في درس سابق، عن بعض أنواع العمليات التي يمكن تنفيذها على قاعدة بيانات، وتسمى تحديدا الدوال أو التوابع، يدعم SQL نوع آخر من العمليات يسمى الإجراءات المخزنة. إذا كنت مبرمج تطبيقات لغات أخرى مثل Pascal أو Visual Basic، فأنت معتاد على استخدام الإجراءات. مثل الدالة؛ يستخدم الإجراء المخزن لتنفيذ عمليات على قاعدة بيانات.

تطبيق عملي: مقدمة إلى الإجراءات المخزنة

1- قم بتشغيل SQL Server Management Studio ثم بتسجيل الدخول إلى الخادم

2- من القائمة الرئيسية، انقر على Query -> New -> File باستخدام الاتصال الحالي

3- لإنشاء قاعدة بيانات جديدة، أدخل الاستعلام التالي:

```
-- =====
-- Database: WattsALoan
-- =====
USE master
GO
-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'WattsALoan'
)
DROP DATABASE WattsALoan
GO

CREATE DATABASE WattsALoan
GO
-- =====
-- Table: Employees
-- =====
USE WattsALoan
GO

IF OBJECT_ID('dbo.Employees', 'U') IS NOT NULL
    DROP TABLE dbo.Employees
GO

CREATE TABLE dbo.Employees
(
```

```

    EmployeeID int identity(1,1) NOT NULL,
    EmployeeNumber nchar(10) NULL,
    FirstName varchar(20) NULL,
    LastName varchar(10),
    FullName AS ((LastName+ ', ' ) + FirstName),
    Title varchar(100),
    HourlySalary money,
    Username varchar(20),
    Password varchar(20),
    CONSTRAINT PK_Employees PRIMARY KEY(EmployeeID)
)
GO
INSERT INTO dbo.Employees(EmployeeNumber, FirstName, LastName, Title,
HourlySalary)
VALUES('293747', 'Jeanne', 'Tryler', 'Accounts Manager', 22.24);
GO
INSERT INTO dbo.Employees(EmployeeNumber, FirstName, LastName, Title,
HourlySalary)
VALUES('492947', 'Helene', 'Gustman', 'Accounts Representative',
14.55);
GO
INSERT INTO dbo.Employees(EmployeeNumber, FirstName, LastName, Title,
HourlySalary)
VALUES('804685', 'Ernest', 'Thomas', 'Accounts Representative',
12.75);
GO
-- =====
-- Table: LoanTypes
-- =====
USE WattsALoan
GO

IF OBJECT_ID('dbo.LoanTypes', 'U') IS NOT NULL
    DROP TABLE dbo.LoanTypes
GO

CREATE TABLE dbo.LoanTypes
(
    LoanTypeID int identity(1,1) NOT NULL,
    LoanType varchar(50) NOT NULL,
    CONSTRAINT PK_LoanTypes PRIMARY KEY(LoanTypeID)
);
GO
INSERT INTO LoanTypes(LoanType) VALUES('Personal Loan');
GO
INSERT INTO LoanTypes(LoanType) VALUES('Car Financing');
GO
INSERT INTO LoanTypes(LoanType) VALUES('Credit Card');
GO
INSERT INTO LoanTypes(LoanType) VALUES('Furniture Loan');
GO
-- =====
-- Table: Customers
-- =====
USE WattsALoan
GO

IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
    DROP TABLE dbo.Customers
GO

```

```

CREATE TABLE dbo.Customers
(
    CustomerID int identity(1,1) NOT NULL,
    DateCreated datetime NULL,
    FullName varchar(50) NOT NULL,
    BillingAddress varchar(100),
    BillingCity varchar(50),
    BillingState varchar(50),
    BillingZIPCode varchar(10),
    EmailAddress varchar(100),
    CONSTRAINT PK_Customers PRIMARY KEY(CustomerID)
)
GO
INSERT INTO Customers(DateCreated, FullName,
    BillingAddress, BillingCity, BillingState,
    BillingZIPCode, EmailAddress)
VALUES('2/26/2004', 'Julius Ramse',
    '927 Feuler Ave', 'Silver Spring',
    'MD', '20904', 'ramses1990@netscape.net');
GO
INSERT INTO Customers(DateCreated, FullName,
    BillingAddress, BillingCity, BillingState,
    BillingZIPCode)
VALUES('06/22/2006', 'Gertrude Vaillant',
    '10055 Larsenic Rd', 'Takoma Park',
    'MD', '20910');
GO
INSERT INTO Customers(DateCreated, FullName,
    BillingAddress, BillingCity, BillingState,
    BillingZIPCode, EmailAddress)
VALUES('12/3/2004', 'James Barrouch',
    '4204 Fallon Drive', 'Silver Spring',
    'MD', '20906', 'barrouchj@hotmail.com');
GO
INSERT INTO Customers(DateCreated, FullName,
    BillingAddress, BillingCity, BillingState,
    BillingZIPCode)
VALUES('08/02/2006', 'Christine Rougher',
    '825 Manning Street', 'Alexandria',
    'VA', '22231');
GO
INSERT INTO Customers(DateCreated, FullName,
    BillingAddress, BillingCity, BillingState,
    BillingZIPCode, EmailAddress)
VALUES('10/08/2006', 'Patrick Heller',
    '2480 Clarington Drive NW', 'Washington',
    'DC', '20006', 'hellerp@yahooo.com');
GO
-- =====
-- Table: LoanAllocation
-- =====
USE WattsALoan
GO

IF OBJECT_ID('dbo.LoanAllocations', 'U') IS NOT NULL
    DROP TABLE dbo.LoanAllocations
GO

CREATE TABLE dbo.LoanAllocations
(
    LoanAllocationID int identity(1,1) NOT NULL,

```

```

DatePrepared datetime NOT NULL,
EmployeeID int NULL
        CONSTRAINT FK_LoanPreparer
        FOREIGN KEY REFERENCES Employees(EmployeeID),
CustomerID int NOT NULL
        CONSTRAINT FK_LoanReceiver
        FOREIGN KEY REFERENCES Customers(CustomerID),
AccountNumber char(10),
LoanTypeID int NOT NULL
        CONSTRAINT FK_LoanTypes
        FOREIGN KEY REFERENCES LoanTypes(LoanTypeID),
LoanAmount money NOT NULL,
InterestRate decimal(6,2) NOT NULL,
Periods decimal(6,2) NOT NULL,
InterestAmount AS
((LoanAmount*(InterestRate/(100)))*(Periods/(12))),
FutureValue AS
(LoanAmount+(LoanAmount*(InterestRate/(100)))*(Periods/(12))),
MonthlyPayment AS
((LoanAmount+(LoanAmount*(InterestRate/(100)))*(Periods/(12)))/Period
s),
Notes Text,
        CONSTRAINT PK_LoanAllocations PRIMARY KEY(LoanAllocationID)
)
GO
INSERT INTO LoanAllocations(DatePrepared, EmployeeID,
        CustomerID, AccountNumber, LoanTypeID, LoanAmount,
        InterestRate, Periods, Notes)
VALUES('2/26/2004', 2, 1, '9171394', 4, 6500.00, 12.65, 36,
        'The loan will be delivered by our furniture business partner
Helios Furnian');
GO
INSERT INTO LoanAllocations(DatePrepared, EmployeeID,
        CustomerID, AccountNumber, LoanTypeID, LoanAmount,
        InterestRate, Periods, Notes)
VALUES('06/22/2007', 2, 2, '8628064', 2, 16500.00, 10.20, 60,
        'For this car loan, our partner Arlington Honda will process
and deliver the car.');
```

```

GO
INSERT INTO LoanAllocations(DatePrepared, EmployeeID,
        CustomerID, AccountNumber, LoanTypeID, LoanAmount,
        InterestRate, Periods, Notes)
VALUES('12/3/2006', 1, 3, '8468364', 3, 500.00, 18.65, 48,
        'This is a regular credit card.');
```

```

GO
INSERT INTO LoanAllocations(DatePrepared, EmployeeID,
        CustomerID, AccountNumber, LoanTypeID, LoanAmount,
        InterestRate, Periods, Notes)
VALUES('08/02/2006', 3, 4, '2483047', 1, 3500.00, 12.74, 36,
        'This is personal/cash loan allocated to a customer who walked
in the store and requested it.');
```

```

GO
INSERT INTO LoanAllocations(DatePrepared, EmployeeID,
        CustomerID, AccountNumber, LoanTypeID, LoanAmount,
        InterestRate, Periods, Notes)
VALUES('10/08/2006', 2, 5, '1311804', 4, 22748.36, 12.28, 60,
        'This is a regular car financing loan');
```

```

GO
-- =====
-- Table: Payments
-- =====
```

```

USE WattsALoan
GO

IF OBJECT_ID('dbo.Payments', 'U') IS NOT NULL
    DROP TABLE dbo.Payments
GO

CREATE TABLE dbo.Payments
(
    PaymentID int identity(1, 1) NOT NULL,
    PaymentDate datetime NOT NULL,
    EmployeeID int NULL
        CONSTRAINT FK_Employees
        FOREIGN KEY REFERENCES Employees(EmployeeID),
    LoanAllocationID int NOT NULL
        CONSTRAINT FK_LoanAllocations
        FOREIGN KEY REFERENCES
LoanAllocations(LoanAllocationID),
    PaymentAmount money NOT NULL,
    Balance money,
    Notes Text,
    CONSTRAINT PK_Payments PRIMARY KEY(PaymentID)
)
GO

```

4- قم بتنفيذ التعليمات البرمجية بالضغط على F5

5- في متصفح الكائنات، قم بتوسيع البند Databases وإذا لزم الأمر WattsALoan

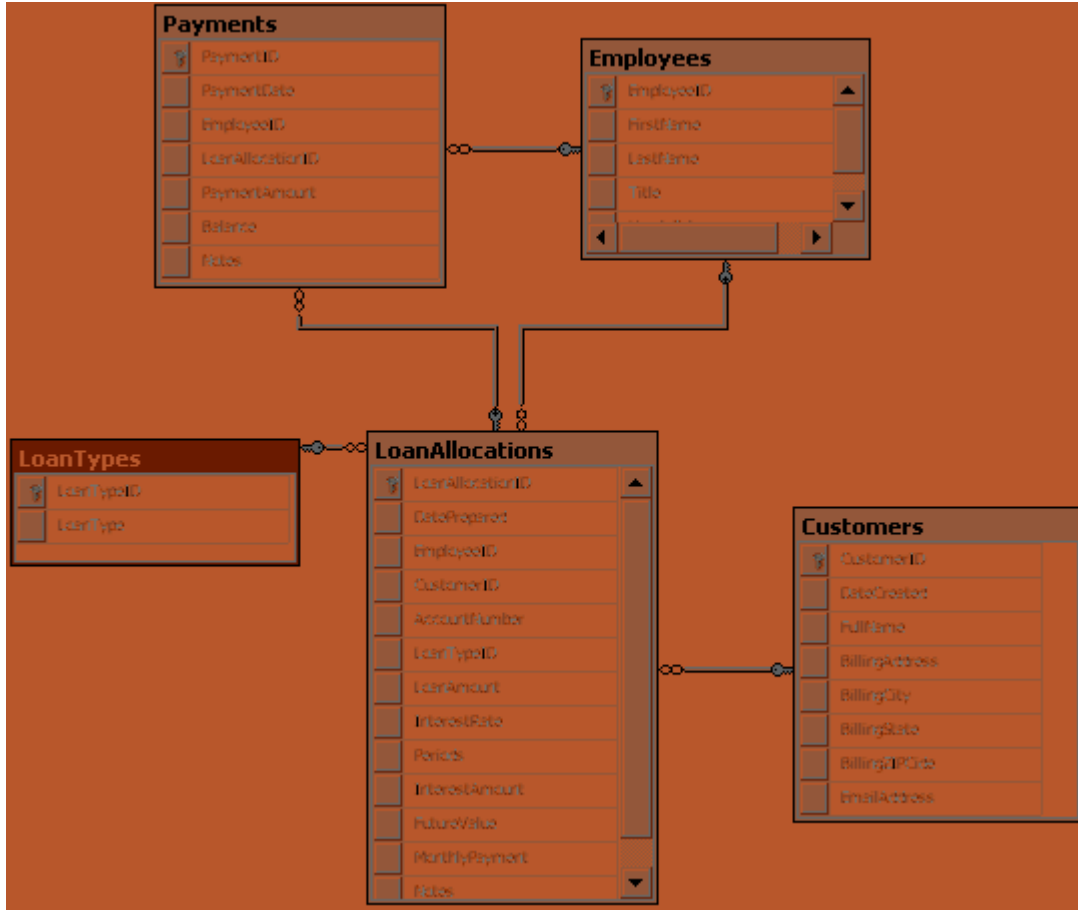
6- ثم انقر على Database Diagram

7- عندما يظهر مربع حوار رسالة التنبيه، انقر على Yes

8- انقر بزر الأيمن على Database Diagram ثم اختر New Database Diagram...

9- في مربع الحوار الحالي، انقر نقرا مزدوجا على كل جدول، وعند إضافة جميع

الجدول، انقر على Close



10- قم بحفظ الرسم التخطيطي باسم dgmWattsALoan ثم أغلقه.

إنشاء الإجراء المخزن:

لإنشاء إجراء مخزن قم بأحد الطرق التالية:

- من متصفح الكائنات، قم بتوسيع قاعدة البيانات حيث تريد إنشاء الإجراء، ثم قم بتوسيع البند Programmability، انقر بزر اليمين على Stored Procedures، واختر New Stored Procedure...، ستظهر نافذة الاستعلام مع الصيغة العامة لإنشاء إجراء مخزن.

- افتح نافذة الاستعلام جديدة على قاعدة البيانات التي تريد إنشاء الإجراء المخزن فيها، ثم قم بعرض Templates Explorer. ومن خلاله قم بتوسيع البند Store Procedure، ثم اسحب Create Stored Procedure وأفلته في إطار الاستعلام.

- افتح نافذة استعلام جديدة على قاعدة البيانات حيث تريد إنشاء الإجراء المخزن وأكتب استعلام إنشاء الإجراء المخزن.

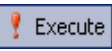
لإنشاء إجراء مخزن في SQL، عليك أن تبدأ الاستعلام بعبارة **CREATE PROCEDURE**، يمكنك أيضا استخدام **CREATE PROC** عوضا عن ذلك. ومثل كل غرض في قاعدة البيانات، يجب عليك تقديم اسم لهذا الإجراء:

- اسم الإجراء يمكن أن يكون بأي شكل وفق القواعد التي استعرضنا لتسمية الأغراض.
- تجنب ابتداء اسم الإجراء بالأحرف **sp_**، لتفادي تعارض يمكن أن يقع مع الإجراءات المدمجة في SQL Server.

بعد اسم الإجراء، اكتب الكلمة **AS** يليها جسم الإجراء، وهو مجموعة من الكلمات أو مجموعة من الأسطر التي تحدد ما سيقوم به الإجراء أو ما سينتجه. تظهر الصيغة بشكل عام:

```
CREATE PROCEDURE ProcedureName
AS
Body of the Procedure
```

من المهم التنبيه أن هناك الكثير من التفاصيل التي يمكن أخذها في الاعتبار لإنشاء الإجراء، حاليا نكتفي بهذه الصيغة، وسنأتي بالتفصيل على ما يمكن أن يحتويه جسم الإجراء.

بعد إنشاء هذا الإجراء، يجب تخزينه بصفته كائن في قاعدة البيانات. للقيام بذلك انقر من شريط الأدوات SQL Editor، على زر التنفيذ . إذا كانت التعليمات البرمجية خالية من الأخطاء، فسيتم إنشاء بند باسم الإجراء تحت البند الرئيسي Stored Procedures لقاعدة البيانات.

تعديل الإجراء:

كبقية كائنات قاعدة البيانات، يمكنك تعديل الإجراء المخزن كما يلي:

- من متصفح الكائنات، يمكنك النقر بالزر الأيمن على الإجراء واختر **Modify**
- من متصفح الكائنات، انقر بالزر الأيمن ثم اختر:

Stored Procedure As -> ALTER To -> New Query Editor Window

- افتح نافذة استعلام جديدة ومرتبطة بقاعدة البيانات التي تحتوي على الإجراء المخزن. ثم من Templates Explorer، قم بتوسيع Stored Procedure، ثم اسحب البند Stored Procedure وأقلته على نافذة الاستعلام.

في كل حالات السابقة، سيتم توليد الصيغة العامة للإجراء المخزن. يمكنك عندئذ تحرير استعلام لتعديل الإجراء المخزن، ثم تنفيذه لتطبيق التعديلات.

في لغة SQL، يتم تعديل الإجراء المخزن وفق الصيغة:

```
ALTER PROCEDURE ProcedureName
AS
Body of Procedure
```

حذف الإجراءات:

يتم التعامل مع الإجراءات في SQL Server على أنها كائنات مستقلة مخزنة في قاعدة البيانات. إذ كما يمكن إنشاؤها وتعديلها يمكن أيضا التخلص منها إذا لم تكن بحاجة إليها.

هناك أنواع مختلفة من الإجراءات المخزنة، البعض منها تعتبر مؤقتة، إذ يتم إنشاؤها لأغراض محددة ثم تحذف نفسها عندما لا تكون هناك حاجة إليها، مثل قطع الاتصال بقاعدة البيانات أو إيقاف تشغيل الكمبيوتر.

لحذف الإجراء، يمكنك استخدام متصفح الكائنات أو استعلام SQL، أي كانت طريقة إنشائه.

لحذف الإجراء باستخدام متصفح الكائنات، بعد توسيع بند قاعدة البيانات والبند Programmability ثم Stored Procedure، يمكنك النقر بالزر الأيمن على الإجراء المخزن ثم اختيار Delete، أو الضغط على Delete من لوح المفاتيح بعد تحديد الإجراء. وقبل الحذف سيظهر مربع الحوار Delete Object لتأكيد خيار الحذف.

لحذف إجراء مخزن باستخدام SQL، تستخدم الصيغة التالية:

```
DROP PROCEDURE ProcedureName
```

بالطبع، يجب عليك التأكد من أنك في قاعدة البيانات الصحيحة، وأن اسم إجراء ProcedureName موجود. وسيتم حذف الإجراء دون أية فرصة لتأكيد خيار الحذف.

إجراءات الاستكشاف:

أبسط استعمال للإجراء يتمثل في الاستعلام عن محتوى الجداول أو بعض الحقول. يتم ذلك باستخدام استعلام **SELECT** وتطبيق التقنيات التي استعرضنا لتحليل البيانات.

مثال لإنشاء الإجراء المخزن الذي يعرض قائمة الطلاب من الجدول Students، يمكن كتابته على النحو التالي:

```
CREATE PROCEDURE GetStudentIdentification
AS
BEGIN
    SELECT FirstName, LastName, DateOfBirth, Gender
    FROM Students
END
GO
```

إلى جانب الاستعلام **SELECT** في الإجراء السابق، يمكنك تنفيذ أي من العمليات التي رأيناها سابقا على قاعدة البيانات، بما في ذلك إنشاء وصيانة السجلات وغير ذلك.

تطبيق عملي: إنشاء إجراء مخزن

1- تأكد من وجود قاعدة البيانات YugoNationalBank التي أنشأناها سابقا، ثم من متصفح

الكائنات، انقر بالزر الأيمن YugoNationalBank وانقر على New Query

2- لإنشاء إجراء مخزن، اكتب ما يلي:

```
USE YugoNationalBank;
GO
-- =====
-- Author:      FunctionX
-- Create date: Friday, May 25, 2007
-- Description: This stored procedure assigns a
--              default password to each employee.
-- =====

CREATE PROCEDURE AssignDefaultPassword
AS
BEGIN
    UPDATE dbo.Employees
    SET Password = 'Password1' FROM dbo.Employees;
END
GO
```

3- قم بتنفيذ الاستعلام بالضغط على F5 لإنشاء الإجراء المخزن.

تنفيذ الإجراءات:

للحصول على نتائج الإجراء مخزن، يجب عليك تنفيذه (بعبارة أخرى: استدعاؤه). لأجل ذلك، تستخدم الكلمة EXECUTE يليها اسم الإجراء. هناك طرق أخرى لتنفيذ الإجراء، نقتصر الآن على أبسط صيغة بالشكل التالي:

```
EXECUTE ProcedureName
```

بنفس الطريقة يمكن استخدام الكلمة EXEC كما يلي:

```
EXEC ProcedureName
```

مثال، إذا كان لديك إجراء يسمى GetStudentIdentification ، لتنفيذه اكتب:

```
EXECUTE GetStudentIdentification
```

يمكنك أيضا أن تسبق اسم الإجراء باسم المخطط، مثل dbo. مثال:

```
EXECUTE dbo.GetStudentIdentification;
```

ويمكنك أيضا أن يسبق اسم مخطط قاعدة البيانات اسم قاعدة البيانات. مثال:

```
EXECUTE ROSH.dbo.GetStudentIdentification;
```

تطبيق عملي: تنفيذ إجراء مخزن

1- قم بحذف الاستعلام واكتب التالي:

```
EXECUTE AssignDefaultPassword;  
GO
```

2- لتنفيذ الإجراء المخزن، اضغط F5.

استخدام التعبيرات والتوابع:

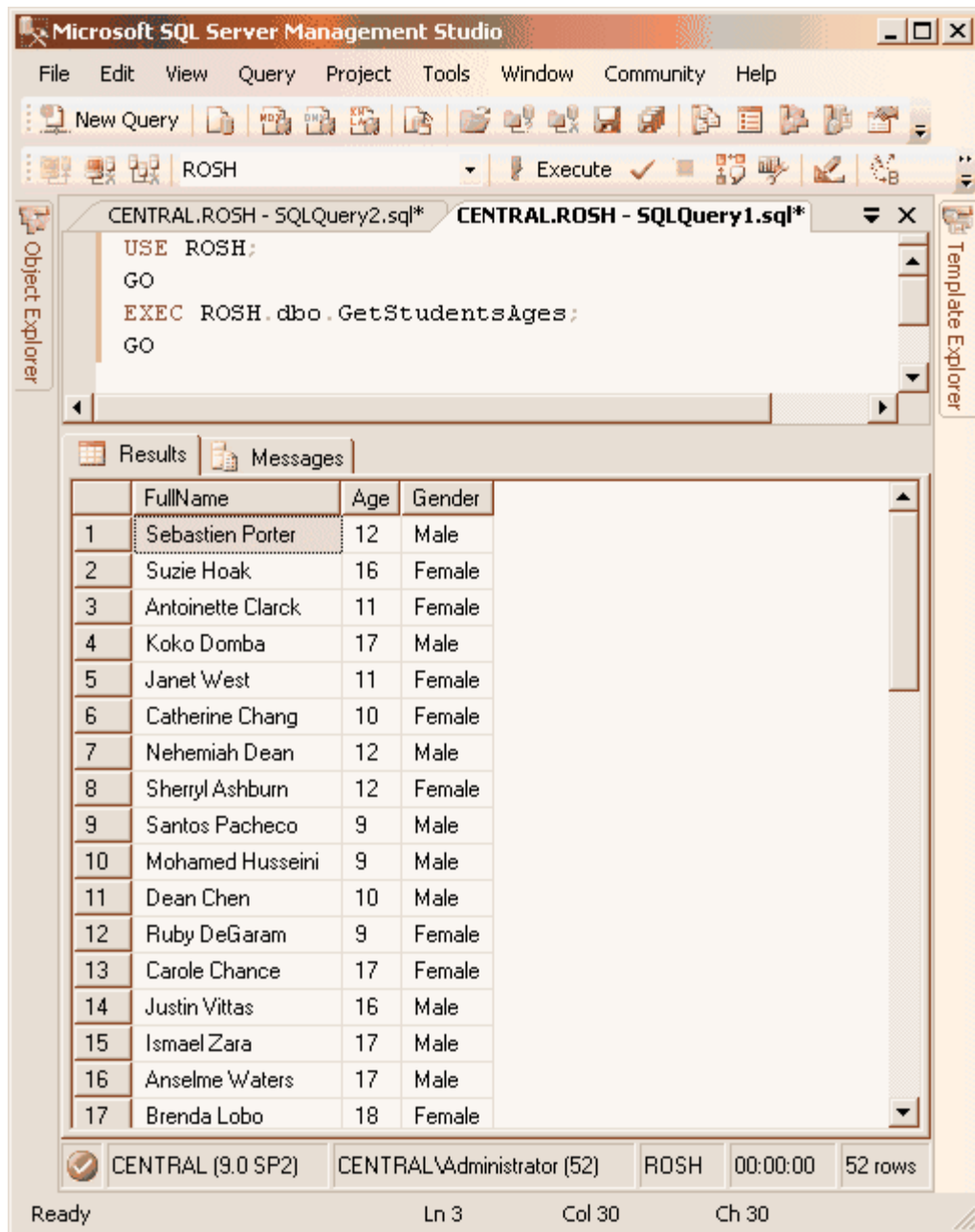
من مزايا استخدام الإجراء المخزن هو ليس فقط إمكانية إنشاء التعبيرات (التي رأيناها سابقا خلال تحليل البيانات)، ولكن أيضا إمكانية تخزين هذه التعبيرات. بناء على هذا، يمكنك إنشاء تعبير يجمع بين الاسم واللقب لإنتاج الاسم الكامل. مثال:

```
CREATE PROCEDURE GetStudentIdentification  
AS  
BEGIN  
    SELECT FullName = FirstName + ' ' + LastName,  
           DateOfBirth, Gender  
    FROM Students  
END  
GO
```

يمكن للإجراء المخزن أيضا استدعاء التوابع المدمجة، يتم ذلك وفقا للقواعد التي استعرضنا لاستدعاء الدوال. مثال إجراء مخزن يستدعي التوابع GETDATE و DATEDIFF:

```
USE ROSH;
GO
CREATE PROCEDURE GetStudentsAges
AS
BEGIN
    SELECT FullName = FirstName + ' ' + LastName,
           DATEDIFF(year, DateOfBirth, GETDATE()) AS Age,
           Gender
    FROM Students
END
GO
```

مثال لتنفيذ هذا الإجراء يظهر كما يلي:



تطبيق عملي: استدعاء توابع من خلال إجراء مخزن

1- قم بمسح الاستعلام السابق

2- أكتب استعلام إجراء المخزن كما يلي:

```
USE YugoNationalBank;
GO
```

```
-- =====
-- Author:          FunctionX
-- Create date:     Friday, May 25, 2007
-- Description:     This stored procedure creates a
--                 username for each employee.
```

```

--          It also assigns an email to the employee.
-- =====
CREATE PROCEDURE CreateUsername
AS
BEGIN
    UPDATE dbo.Employees
    SET Username = LOWER(LEFT(FirstName, 1) + LEFT(LastName, 5))
    FROM dbo.Employees;
    UPDATE dbo.Employees
    SET EmailAddress = LOWER(LEFT(FirstName, 1) + LEFT(LastName,
5)) +
                                '@yugonationalbank.com'
                                FROM dbo.Employees;
END
GO

```

3- لإنشاء الإجراء المخزن، اضغط F5

4- احذف الاستعلام وأكتب عوضا عنه التالي:

```

EXECUTE CreateUsername;
GO

```

5- اضغط F5 للتنفيذ.

معلومات الإجراءات والوسائط:

في الإجراءات المخزنة التي رأيناها حتى الآن، يفترض فيها أن القيم التي تستخدمها متوفرة في جداول من قاعدة البيانات. في بعض الحالات، قد تحتاج إلى إنشاء إجراء مخزن يستخدم قيمة غير موجودة في قاعدة البيانات. في مثل هذه الحالات نحتاج إلى تزويد الإجراء بقيمة أو بغير قيمة أخرى.

تسمى القيم الخارجية التي يتم تزويدها للإجراء المخزن بالمعاملات. يتم إنشاء المعلمات، إذا اقتضى الأمر، عند إنشاء الإجراء المخزن. يقال عن الإجراء المخزن المزود بمعاملة أنه يأخذ وسيط، ويمكن للإجراء المخزن أيضا أن يأخذ أكثر من وسيط واحد.

عند تنفيذ الإجراء المخزن الذي يأخذ أكثر من وسيط، يجب حينئذ توفير (تمرير) قيمة لكل وسيط. هناك حالات لا يتطلب عندما الإجراء تمرير قيمة للوسيط.

تمرير القيم إلى الوسائط:

لإنشاء إجراء مخزن يأخذ وسيط، اكتب الصيغة **CREATE PROCEDURE** أو **CREATE PROC** يليه اسم الإجراء، ثم اكتب اسم الوسيط بدءا بالرمز @. يتم إنشاء

المعلمة كما يتم إنشاء الحقل من الجدول، حيث يجب أن يكون له اسم ونوع البيانات والطول إذا لزم الأمر. الصيغة العامة لإنشاء إجراء مخزن يأخذ وسيط وحيد كما يلي:

```
CREATE PROCEDURE ProcedureName
@ParameterName DataType
AS
Body of the Procedure
```

عند إضافة وسيط إلى الإجراء المخزن، يمكنك أن تحدد ما الذي تريد فعله بالوسيط في جسم هذا الإجراء. يمكنك مثلاً استخدام المعلمة لتنفيذ استعلام متغير يمكن أن يوفره المستخدم. مثال: نريد إنشاء إجراء يتم تزويده بوسيط يمثل قيمة حقل الجنس، وبعد تنفيذه نحصل على قائمة من الطلاب لهذا الجنس، يمكن كتابة هذا الإجراء كما يلي:

```
CREATE PROC GetListOfStudentsByGender
@Gdr VARCHAR(12)
AS
SELECT FirstName, LastName,
DateOfBirth, HomePhone, Gender
FROM Students
WHERE Gender = @Gdr
```

تطبيق عملي: إنشاء الإجراء المخزن

1- قم بحذف الاستعلام السابق، ثم اكتب إجراء مخزن مزود بمعلومات كما ما يلي:

```
USE WattsALoan;
GO
```

```
CREATE PROCEDURE SpecifyCurrentBalance
@PmtDate datetime,
@EmplID int,
@LaID int,
@PmtAmt money
AS
BEGIN
-- Get the amount that was lent to the customer
DECLARE @AmountOfLoan money;
SET @AmountOfLoan = (SELECT las.FutureValue
FROM LoanAllocations las
WHERE (las.LoanAllocationID = @LaID));

-- If the customer had already made at least one payment,
-- get the current balance of the customer's account
DECLARE @CurrentBalance money;
SET @CurrentBalance = (SELECT MIN(pay.Balance)
FROM Payments pay
WHERE (pay.LoanAllocationID = @LaID));

-- If the customer has never made a payment (yet),
-- to specify the balance, subtract the current payment
-- from the original amount of the loan
IF @CurrentBalance IS NULL
BEGIN
```

```

INSERT INTO Payments(PaymentDate, EmployeeID,
                    LoanAllocationID, PaymentAmount, Balance)
VALUES(@PmtDate, @EmplID, @LaID, @PmtAmt,
      @AmountOfLoan - @PmtAmt);
END
-- If the customer had already at least one payment,
-- subtract the current payment from the previous balance
ELSE
BEGIN
INSERT INTO Payments(PaymentDate, EmployeeID,
                    LoanAllocationID, PaymentAmount,
Balance)
VALUES(@PmtDate, @EmplID, @LaID,
      @PmtAmt, @CurrentBalance - @PmtAmt);
END
END
GO

```

2- لإنشاء الإجراء المخزن، اضغط F5

تنفيذ الإجراءات المخزنة:

كما ذكر سابقاً، عند تنفيذ الإجراءات المخزنة التي تأخذ معلمة، تأكد من توفير قيمة لكل معلمة. وفقاً للصيغة التالية:

```
EXEC ProcedureName ParameterValue
```

تأكد من تمرير قيمة مناسبة لمعلمة الإجراء ومطابقة لنوعها. إذا كان نوع المعلمة حرف أو سلسلة نصية، اكتب قيمتها بين علامتي اقتباس. مثال:

```
EXEC ROSH.dbo.GetListOfStudentsByGender 'Male';
```

مثال لتنفيذ الإجراء:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
USE ROSH;
GO
EXEC ROSH.dbo.GetListOfStudentsByGender 'Male';
GO
```

The Results pane displays the following data:

	FirstName	LastName	DateOfBirth	HomePhone	Gender
1	Sebastien	Porter	1995-02-12 00:00:00.000	(301) 591-6236	Male
2	Koko	Domba	1990-02-05 00:00:00.000	(703) 363-1066	Male
3	Nehemiah	Dean	1995-12-02 00:00:00.000	(301) 938-2763	Male
4	Santos	Pacheco	1998-05-05 00:00:00.000	(703) 652-4412	Male
5	Mohamed	Husseini	1998-01-05 00:00:00.000	(202) 556-4766	Male
6	Dean	Chen	1997-05-02 00:00:00.000	(703) 518-3372	Male
7	Justin	Vittas	1991-05-04 00:00:00.000	(301) 549-4004	Male
8	Ismael	Zara	1990-05-25 00:00:00.000	(301) 356-1826	Male
9	Anselme	Waters	1990-07-23 00:00:00.000	(202) 525-0160	Male
10	Danilo	Chico	1997-06-20 00:00:00.000	(301) 364-9212	Male
11	Paul	Farms	1996-04-30 00:00:00.000	(301) 695-7727	Male
12	Lester	Bell	1995-12-06 00:00:00.000	(301) 979-0032	Male
13	Christian	Liss	1988-10-08 00:00:00.000	(202) 389-7487	Male
14	Clint	Fuller	1996-04-19 00:00:00.000	(301) 490-6390	Male
15	Thomas	Moore	1988-04-22 00:00:00.000	(410) 730-8100	Male
16	Tim	Amorros	1989-02-14 00:00:00.000	(301) 262-1717	Male
17	Maurice	Walken	1990-10-08 00:00:00.000	(202) 583-4228	Male

The status bar at the bottom indicates: Ready, Ln 4, Col 3, Ch 3, I. The message pane shows: CENTRAL (9.0 SP2) | CENTRAL\Administrator (52) | ROSH | 00:00:00 | 25 rows.

لاحظ أنه ينبغي عدم إدراج حقل الجنس في الاستعلام لأنه سيكون من خيار المستخدم.

يمكن إنشاء نوع آخر من الإجراءات المخزنة التي تأخذ أكثر من معلمة واحدة. في هذه الحالة، يتم إنشاء المعلمات قبل الكلمة AS، مع الفصل بين المعلمات برمز الفاصلة كما يلي:

```
CREATE PROCEDURE ProcedureName
@ParameterName1 DataType, @ParameterName2 DataType, @ParameterName_n
DataType
AS
```

مثال:

```
USE ROSH;
```

```

GO
CREATE PROCEDURE IdentifyStudentsByState
    @Gdr varchar(20),
    @StateOrProvince char(2)
AS
BEGIN
    SELECT FullName = LastName + ', ' + FirstName,
           DATEDIFF(year, DateOfBirth, GETDATE()) AS Age,
           Gender
    FROM Students
    WHERE (Gender = @Gdr) AND (State = @StateOrProvince)
END
GO

```

عند استدعاء الإجراء المخزن الذي يأخذ أكثر من معلمة، يجب أيضا توفير قيمة لكل معلمة وهنا لديك خيارين. إما أن تقوم بتوفير قيمة لكل معلمة وفق الترتيب الذي يظهر في الإجراء المخزن. مثال:

```

USE ROSH;
GO
EXEC ROSH.dbo.IdentifyStudentsByState 'Female', 'MD';
GO

```

ونائج الاستعلام:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```
USE ROSH;
GO
EXEC ROSH.dbo.IdentifyStudentsByState 'Female', 'MD';
GO
```

The Results pane shows the following data:

	FullName	Age	Gender
1	Hoak, Suzie	16	Female
2	Clarck, Antoinette	11	Female
3	West, Janet	11	Female
4	Ashburn, Sheryl	12	Female
5	DeGaram, Ruby	9	Female
6	Chance, Carole	17	Female
7	Verde, Suzanna	19	Female
8	Franse, Mincy	16	Female
9	Ledoux, Gabrielle	18	Female
10	Duma, Arlette	12	Female
11	Howerson, Bernadette	11	Female
12	Steinberg, Judith	18	Female
13	Napolis, Ella	16	Female
14	Miller, Ann	10	Female
15	Broadskey, Millicent	18	Female
16	Milchen, Victoria	19	Female
17	Laurens, Julie	19	Female

The status bar at the bottom indicates: Ready, Ln 3, Col 20, Ch 20, INS. The message pane shows: Q... CENTRAL (9.0 SP2) CENTRAL\Administrator (52) ROSH 00:00:00 21 rows.

أو يمكنك بطريقة أخرى، توفير قيمة لكل معلمة في ترتيب من اختيارك مع تحديد اسم المعلمة. مثال: الإجراء التالي الذي يأخذ 3 معلمات:

```
USE ROSH;
GO
CREATE PROCEDURE IdentifySomeStudents
    @Gdr varchar(20),
    @StateOrProvince char(2),
    @HomeStatus bit
AS
BEGIN
    SELECT FullName = LastName + ', ' + FirstName,
           DATEDIFF(year, DateOfBirth, GETDATE()) AS Age,
           Gender
    FROM Students
```

```

WHERE (Gender = @Gdr) AND
      (State = @StateOrProvince) AND
      (SPHome = @HomeStatus)
END
GO

```

عند استدعاء هذا النوع من الإجراءات، يمكنك كتابة اسم كل معلمة وتعيين القيمة المقابلة.
مثال:

```

EXEC IdentifySomeStudents @HomeStatus=1, @StateOrProvince='MD',
@Gdr='Female';

```

مثال لتنفيذ الإجراء:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```

USE ROSH;
GO
EXEC IdentifySomeStudents @HomeStatus=1,
@StateOrProvince='MD',
@Gdr='Female';
GO

```

The Results pane shows the following data:

	FullName	Age	Gender
1	West, Janet	11	Female
2	Duma, Arlette	12	Female
3	Milchen, Victoria	19	Female
4	Marlly, Paul	17	Female
5	Mart, Donnie	17	Female

The status bar at the bottom indicates: Ready, Ln 5, Col 26, Ch 26. The execution details show: CENTRAL (9.0 SP2), CENTRAL\Administrator (52), ROSH, 00:00:00, 5 rows.

تطبيق عملي: تنفيذ إجراء متعدد المعلمات

1- قم بحذف الاستعلام السابق، ثم إنشاء إجراء مخزن بما يلي:

```

USE WattsALoan;
GO
EXECUTE SpecifyCurrentBalance '03/25/2004', 2, 1, 249.08;
GO
EXECUTE SpecifyCurrentBalance '01/30/2006', 2, 5, 611.93;
GO
EXECUTE SpecifyCurrentBalance '04/20/2004', 1, 1, 249.08;
GO

```

```
EXECUTE SpecifyCurrentBalance '10/28/2006', 2, 4, 134.38;
GO
```

2- نفذ استعلام بضغط F5.

الوسيط الافتراضي:

تخيل أنك تريد إنشاء قاعدة بيانات لتخزين وإدارة سلع المتجر:

Column Name	Data Type	Allow Nulls
ItemCategoryID	int	<input type="checkbox"/>
ItemCategory	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
StoreItemID	int	<input type="checkbox"/>
ItemNumber	char(8)	<input type="checkbox"/>
ItemCategoryID	int	<input checked="" type="checkbox"/>
ItemName	varchar(50)	<input type="checkbox"/>
ItemSide	varchar(30)	<input checked="" type="checkbox"/>
UnitPrice	smallmoney	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

بعد ملء الجدول بعدد من البنود على النحو التالي:

ItemCategoryID	ItemCategory
1	Men
2	Women
3	Boys
4	Girls
5	Miscellaneous
*	NULL

ItemNumber	ItemCategoryID	ItemName	ItemSize	UnitPrice
264850	2	Long-Sleeve Jersey Dress	Petite	39.95
930405	4	Solid Crewneck Tee	Medium	12.95
293004	1	Cotton Comfort Open Bottom Pant	XLarge	17.85
924515	1	Hooded Full-Zip Sweatshirt	S	69.95
405945	3	Plaid Pinpoint Dress Shirt	22 35-36	35.85

294936	2	Cool-Dry Soft Cup Bra	36D	15.55
294545	2	Ladies Hooded Sweatshirt	Medium	45.75
820465	2	Cotton Knit Blazer	M	295.95
294694	2	Denim Blazer - Natural Brown	Large	75.85
924094	3	Texture-Striped Pleated Dress Pants	44x30	32.85
359405	3	Iron-Free Pleated Khaki Pants	32x32	39.95
192004	3	Sunglasses		15.85

تخيل أنك تريد إنشاء طريقة آلية تحسب سعر كل عنصر بعد تطبيق معدل خصم عليه.

مثل هذا الإجراء يمكن أن يكتب على النحو التالي:

```
CREATE PROC CalculateNetPrice
@discount Decimal
AS
SELECT ItemName, UnitPrice - (UnitPrice * @discount / 100)
FROM StoreItems
```

تنفيذ هذا الاستعلام يكون بهذا الشكل:

	ItemName	(No column name)
1	Long-Sleeve Jersey Dress	31.96000000
2	Solid Crewneck Tee	10.36000000
3	Cotton Comfort Open Bottom Pant	14.28000000
4	Hooded Full-Zip Sweatshirt	55.96000000
5	Plaid Pinpoint Dress Shirt	28.68000000
6	Cool-Dri® Soft Cup Bra	12.44000000
7	Ladies Hoody Sweatshirt	36.60000000
8	Cotton Knit Blazer	236.76000000
9	Denim Blazer - Natural Brown	60.68000000
10	Texture-Striped Pleated Dress Pants	26.28000000
11	Iron-Free Pleated Khaki Pants	31.96000000
12	Sunglasses	12.68000000

إذا كنت تريد إنشاء إجراء مخزن يأخذ وسيط من المرجح أن يأخذ نفس القيمة في معظم

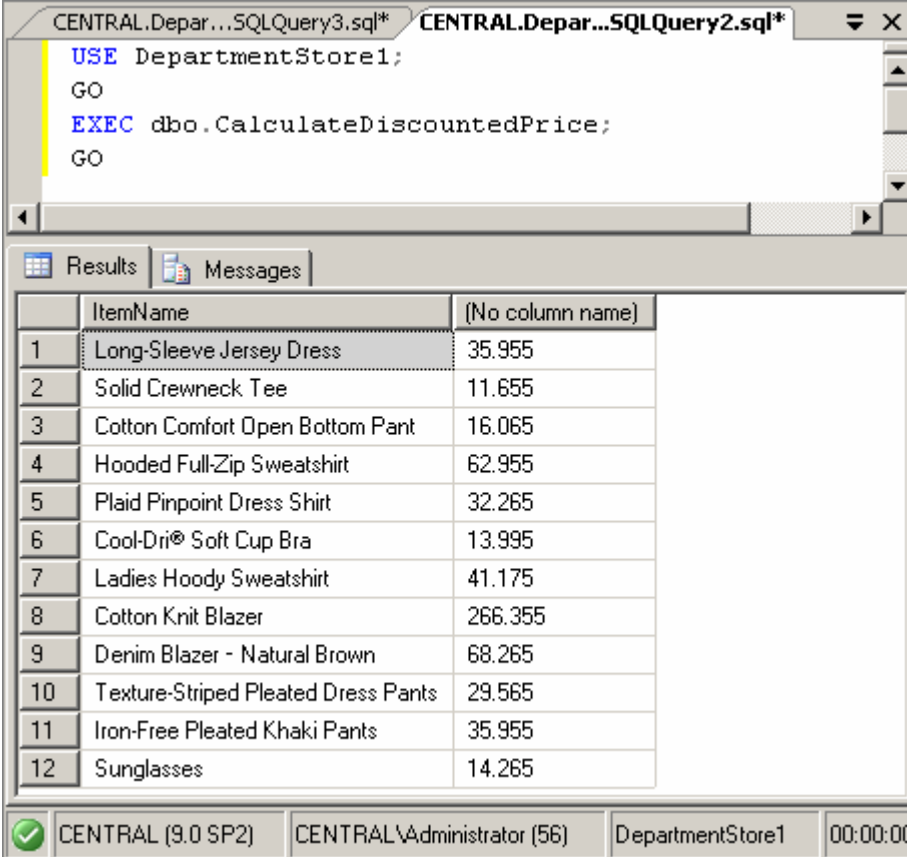
الحالات، يمكنك توفير هذه القيمة على أساس قيمة افتراضية، يعني ذلك أن المستخدم عندما

يقوم باستدعاء الإجراء المخزن دون توفير قيمة للوسيط، فإن القيمة الافتراضية هي التي تحل محل قيمة الوسيط.

لإنشاء إجراء مخزن يأخذ وسيط يحمل قيمة افتراضية، أكتب بعد الإعلان عن الوسيط على يمينه العلامة = تليها القيمة المطلوبة. مثال مطبق على قاعدة البيانات السابقة:

```
CREATE PROC CalculateDiscountedPrice
@discount decimal = 10.00
AS
    SELECT ItemName, UnitPrice - (UnitPrice * @discount / 100)
FROM StoreItems;
GO
```

عند تنفيذ الإجراء المخزن الذي يأخذ وسيط بقيمة افتراضية، لا يجب توفير قيمة لهذا الوسيط إذا كان القيمة مناسبة. على هذا الأساس يمكن استدعاء الإجراء أعلاه على النحو التالي:



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'CENTRAL.Depar...SQLQuery3.sql*' and 'CENTRAL.Depar...SQLQuery2.sql*'. The active tab contains the following SQL code:

```
USE DepartmentStore1;
GO
EXEC dbo.CalculateDiscountedPrice;
GO
```

Below the code, the 'Results' tab is active, displaying a table with 12 rows. The table has two columns: 'ItemName' and '(No column name)'. The data is as follows:

	ItemName	(No column name)
1	Long-Sleeve Jersey Dress	35.955
2	Solid Crewneck Tee	11.655
3	Cotton Comfort Open Bottom Pant	16.065
4	Hooded Full-Zip Sweatshirt	62.955
5	Plaid Pinpoint Dress Shirt	32.265
6	Cool-Dri® Soft Cup Bra	13.995
7	Ladies Hoody Sweatshirt	41.175
8	Cotton Knit Blazer	266.355
9	Denim Blazer - Natural Brown	68.265
10	Texture-Striped Pleated Dress Pants	29.565
11	Iron-Free Pleated Khaki Pants	35.955
12	Sunglasses	14.265

At the bottom of the window, the status bar shows: 'CENTRAL (9.0 SP2) | CENTRAL\Administrator (56) | DepartmentStore1 | 00:00:00'.

إذا كانت القيمة الافتراضية غير مناسبة للحساب الحالي، يمكنك توفير قيمة أخرى. مثال:

CENTRAL.Depar...SQLQuery3.sql* CENTRAL.Depar...SQLQuery2.sql*

```
USE DepartmentStore1;
GO
EXEC dbo.CalculateDiscountedPrice 40;
GO
```

Results Messages

	ItemName	(No column name)
1	Long-Sleeve Jersey Dress	23.97
2	Solid Crewneck Tee	7.77
3	Cotton Comfort Open Bottom Pant	10.71
4	Hooded Full-Zip Sweatshirt	41.97
5	Plaid Pinpoint Dress Shirt	21.51
6	Cool-Dri® Soft Cup Bra	9.33
7	Ladies Hoody Sweatshirt	27.45
8	Cotton Knit Blazer	177.57
9	Denim Blazer - Natural Brown	45.51
10	Texture-Striped Pleated Dress Pants	19.71
11	Iron-Free Pleated Khaki Pants	23.97
12	Sunglasses	9.51

CENTRAL (9.0 SP2) CENTRAL\Administrator (56) DepartmentStore1 00:00:00

باستخدام نفس الطريقة، يمكنك إنشاء إجراء مخزن يأخذ أكثر من وسيط مع القيم الافتراضية. مثال إجراء مخزن يأخذ وسيطين لكل منها قيمة افتراضية:

```
CREATE PROC CalculateSalePrice
@Discount decimal = 20.00,
@TaxRate decimal = 7.75
AS
SELECT ItemName As [Item Description],
UnitPrice As [Marked Price],
UnitPrice * @Discount / 100 As [Discount Amt],
UnitPrice - (UnitPrice * @Discount / 100) As [After Discount],
UnitPrice * @TaxRate / 100 As [Tax Amount],
(UnitPrice * @TaxRate / 100) + UnitPrice -
(UnitPrice * @Discount / 100) + (@TaxRate / 100) As [Net
Price]
FROM StoreItems;
GO
```

مثال لتنفيذ الإجراء:

Item Description	Marked...	Discount ...	After Discount	Tax Amount	Net Price
1 Long-Sleeve Jersey ...	39.95	7.99000000	31.96000000	3.19600000	35.23600000
2 Solid Crewneck Tee	12.95	2.59000000	10.36000000	1.03600000	11.47600000
3 Cotton Comfort Open ...	17.85	3.57000000	14.28000000	1.42800000	15.78800000
4 Hooded Full-Zip Swe...	69.95	13.99000...	55.96000000	5.59600000	61.63600000
5 Plaid Pinpoint Dress ...	35.85	7.17000000	28.68000000	2.86800000	31.62800000
6 Cool-Dri® Soft Cup Bra	15.55	3.11000000	12.44000000	1.24400000	13.76400000
7 Ladies Hoody Sweats...	45.75	9.15000000	36.60000000	3.66000000	40.34000000
8 Cotton Knit Blazer	295.95	59.19000...	236.76000000	23.67600000	260.51600000
9 Denim Blazer - Natur...	75.85	15.17000...	60.68000000	6.06800000	66.82800000
10 Texture-Striped Pleat...	32.85	6.57000000	26.28000000	2.62800000	28.98800000
11 Iron-Free Pleated Kha...	39.95	7.99000000	31.96000000	3.19600000	35.23600000
12 Sunglasses	15.85	3.17000000	12.68000000	1.26800000	14.02800000

عند استدعاء إجراء مخزن يأخذ أكثر من وسيط، حيث لكل وسيط قيمة افتراضية، لا يجب توفير قيمة لكل وسيطة، ويمكنك توفير قيمة لأحد الوسائط فقط أو لبعض منها. الإجراء أعلاه يمكن أن يستدعى بوسيط واحد على النحو التالي:

```
EXEC CalculateSalePrice2 55.00
```

في هذه الحالة، فإن الوسائط الأخرى ستستخدم القيمة الافتراضية.

يمكن أيضا استدعاء الإجراء المخزن أعلاه مع تزويده بقيمة الوسيط الثاني، كما يلي:

```
EXEC CalculateSalePrice2 @TaxRate = 8.55
```

في هذه الحالة، يأخذ الوسيط الأول قيمته الافتراضية.

تطبيق عملي: استخدام القيمة الافتراضية للوسيط

1- قم بحذف الاستعلام السابق

2- لإنشاء نسخة جديدة من الإجراء المخزن السابق، اكتب ما يلي:

```
USE WattsALoan;
GO
```

```

DROP PROCEDURE SpecifyCurrentBalance;
GO
CREATE PROCEDURE SpecifyCurrentBalance
    @PmtDate datetime,
    @EmplID int,
    @LaID int,
    @PmtAmt money,
    @Comments Text = ''
AS
BEGIN
    -- Get the amount that was lent to the customer
    DECLARE @AmountOfLoan money;
    SET      @AmountOfLoan = (SELECT las.FutureValue
                              FROM LoanAllocations las
                              WHERE (las.LoanAllocationID =
@LaID));

    -- If the customer had already made at least one payment,
    -- get the current balance of the customer's account
    DECLARE @CurrentBalance money;
    SET      @CurrentBalance = (SELECT MIN(pay.Balance)
                              FROM Payments pay
                              WHERE (pay.LoanAllocationID =
@LaID));

    -- If the customer has never made a payment (yet),
    -- to specify the balance, subtract the current payment
    -- from the original amount of the loan
    IF      @CurrentBalance IS NULL
    BEGIN
        INSERT INTO Payments(PaymentDate, EmployeeID,
                              LoanAllocationID, PaymentAmount,
                              Balance, Notes)
        VALUES(@PmtDate, @EmplID, @LaID, @PmtAmt,
                @AmountOfLoan - @PmtAmt, @Comments);
    END
    -- If the customer had already at least one payment,
    -- subtract the current payment from the previous balance
    ELSE
    BEGIN
        INSERT INTO Payments(PaymentDate, EmployeeID,
                              LoanAllocationID, PaymentAmount,
                              Balance, Notes)
        VALUES(@PmtDate, @EmplID, @LaID,
                @PmtAmt, @CurrentBalance - @PmtAmt,
@Comments);
    END
END
GO

```

3- لإنشاء الإجراء المخزن اضغط F5

4- احذف محتويات إطار الاستعلام، ثم اكتب الاستعلام الآتي:

```

USE WattsALoan;
GO

EXECUTE SpecifyCurrentBalance '07/15/2004', 3, 1, 498.16,
'The customer sent a double-payment to cover this and last month';
GO

```

```

EXECUTE SpecifyCurrentBalance '01/26/2007', 1, 3, 50;
GO
EXECUTE SpecifyCurrentBalance '08/26/2004', 2, 1, 249.08;
GO
EXECUTE SpecifyCurrentBalance '02/28/2006', 3, 5, 611.93;
GO
EXECUTE SpecifyCurrentBalance '10/24/2007', 2, 2, 415.25;
GO
EXECUTE SpecifyCurrentBalance '05/30/2004', 1, 1, 249.08;
GO
EXECUTE SpecifyCurrentBalance '02/22/2007', 2, 3, 20;
GO
EXECUTE SpecifyCurrentBalance '03/23/2006', 2, 5, 611.93;
GO
EXECUTE SpecifyCurrentBalance '07/22/2007', 2, 2, 415.25,
    'First regular payment';
GO
EXECUTE SpecifyCurrentBalance '12/24/2006', 1, 5, 611.93;
GO
EXECUTE SpecifyCurrentBalance '04/25/2006', 3, 5, 611.93;
GO
EXECUTE SpecifyCurrentBalance '09/26/2007', 2, 2, 415.25;
GO
EXECUTE SpecifyCurrentBalance '09/24/2006', 3, 4, 134.38;
GO
EXECUTE SpecifyCurrentBalance '03/25/2007', 2, 3, 25;
GO
EXECUTE SpecifyCurrentBalance '11/28/2006', 2, 5, 611.93,
    'First Car Payment';
GO
EXECUTE SpecifyCurrentBalance '08/28/2007', 1, 2, 415.25,
    'Second payment';
GO

```

5- نفذ الاستعلام بالضغط على F5.

معلومات الإخراج:

يقوم Transact-SQL بتمرير قيم المتغيرات على أساس قيمة مرجعية، يمكن لهذا النوع من التمرير أن يقوم بإرجاع قيمة. بعبارة أخرى، يمكنك إنشاء معلمة في الإجراء المخزن الغرض منها إعادة قيمة جديدة عند نهاية الإجراء، ثم استخدام هذه القيمة على النحو الذي تراه مناسباً.

لإنشاء المعلمة التي تعيد قيمة من الإجراء المخزن، أضف بعد اسم الإجراء الكلمة

OUTPUT، يمكنك استخدامها وفقاً للصيغة التالية:

```

CREATE PROCEDURE ProcedureName
@ParameterName DataType OUTPUT
AS
Body of the Procedure

```

يمكنك في جسم هذا الإجراء تنفيذ المهام المناسبة. ويجب قبل نهاية هذا الإجراء تحديد قيمة للمعلمة التي تعيد قيمة. ستأخذ هذه المعلمة تلك القيمة حال الخروج من الإجراء. مثال:

```
CREATE PROCEDURE dbo.CreateFullName
    @FName varchar(20),
    @LName varchar(20),
    @FullName varchar(42) OUTPUT
AS
    SELECT @FullName = @LName + ', ' + @FName
GO
```

عند استدعاء الإجراء المخزن الذي يعيد قيمة في أحد المعلمات، يجب تمرير متغير لهذه المعلمة، ثم فيما بعد يمكنك استخدام هذه القيمة على النحو الذي تراه مناسباً. مثال:

```
DECLARE @FirstName varchar(20),
        @LastName varchar(20),
        @Full varchar(42)
SET @FirstName = 'Melanie';
SET @LastName = 'Johanssen';

EXECUTE dbo.CreateFullName @FirstName, @LastName, @Full OUTPUT

SELECT @Full;
GO
```

من مزايا استخدام التوابع أو الإجراءات المخزنة هو أن لديها إمكانية الوصول إلى الجداول والسجلات من قاعدة البيانات. هذا يعني أنه يمكنك الوصول إلى الحقول والسجلات طالما قمت بتحديد الجدول أو View، وهو ما يحدث باستخدام الاستعلام SELECT، مثال:

```
USE ROSH;
GO

CREATE PROCEDURE ShowStudentsFullNames
    @FullName varchar(42) OUTPUT
AS
    SELECT @FullName = LastName + ', ' + FirstName FROM Students;
GO
```

سيقوم هذا الإجراء المخزن بعد تنفيذه بإرجاع قيمة واحدة فقط. المثال التالي ينفذ الإجراء

أعلاه:

```

CENTRAL.ROSH - SQLQuery8.sql*  CENTRAL.Peopl...QLQuery7.sql*  CENTRAL.Depar...SQLQuery6.sq
USE ROSH;
GO
/*
CREATE PROCEDURE ShowStudentsFullNames
    @FullName varchar(42) OUTPUT
AS
    SELECT @FullName = LastName + ', ' + FirstName FROM Students;
GO
*/
DECLARE @fName varchar(20);
EXECUTE ShowStudentsFullNames @fName OUTPUT;
SELECT @fName;
GO

```

Results	
	[No column name]
1	Laurel, Charles

عند استدعاء مثل هذا الإجراء، وعدم تحديد شرط لإخراج نتيجة معينة، فسيقوم مترجم SQL في هذه الحالة بتحديد السجل السابق. هذا يعني أنه يجب عليك التأكد دوماً من أن الإجراء المخزن الخاص الذي يأخذ معلمة إخراج قادر على عزل النتيجة.

يمكن إضافة شرط الاستعلام **WHERE** إذا كان الإجراء المخزن يحتوي الاستعلام **SELECT**. مثال:

```

USE ROSH;
GO

CREATE PROCEDURE ShowStudentsFullNames
    @FullName varchar(42) OUTPUT
AS
    SELECT @FullName = LastName + ', ' + FirstName FROM Students
    WHERE StudentID = 8;
GO

```

يعرض هذا الإجراء قيمة السجل الثامن المخزن في الجدول.

ملخص الدرس (تطبيقات):

قم بإنشاء إجراء مخزن باسم ProcessPayroll بعدة معلمات، وظيفته:

1- إيجاد عدد ساعات العمل خلال الأسبوع

2- حساب أجرة أسبوع لكل العامل

3- عدد ساعات العمل الإجمالية (لكل العمال) خلال أسبوع.

Views

نظرة عامة:

رأينا عند دراسة وتحليل البيانات، أن الاستعلام تقنية تسمح بعزل سلسلة من الحقول و/أو السجلات من الجداول، يتم ذلك عادة لغرض تحليل البيانات. ويمكن أيضا القيام بذلك لأجل إنشاء قائمة جديدة من البيانات لأغراض معينة.

يتم إنشاء الاستعلام غالبا بشكل مؤقت، مثلا خلال تحليل البيانات، باستخدام الجداول، النماذج، صفحات على شبكة الإنترنت. يتم التخلص من الاستعلام مباشرة بعد الحصول على القائمة المطلوبة.

تسمح العديد من تطبيقات قواعد البيانات، بما في ذلك SQL Server، بإنشاء استعلامات مع إمكانية تخزينها لاستخدامها لاحقا، أو حتى لاستخدامها كما لو كانت جداول من قاعدة البيانات. هذه هي الفكرة الأساسية حول طريقة عمل كائنات تدعى **Views**.

View هي لائحة من الحقول أو مجموعة من السجلات التي تم استردادها من جدول أو أكثر، أو مزيج من View واحدة أو أكثر، أو Views وجداول أخرى. بناء على هذا، قبل إنشاء View يجب أولا تحديد الحقول والسجلات التي سيتم التعامل معها.

تطبيق عملي: التمهيدي لإنشاء View

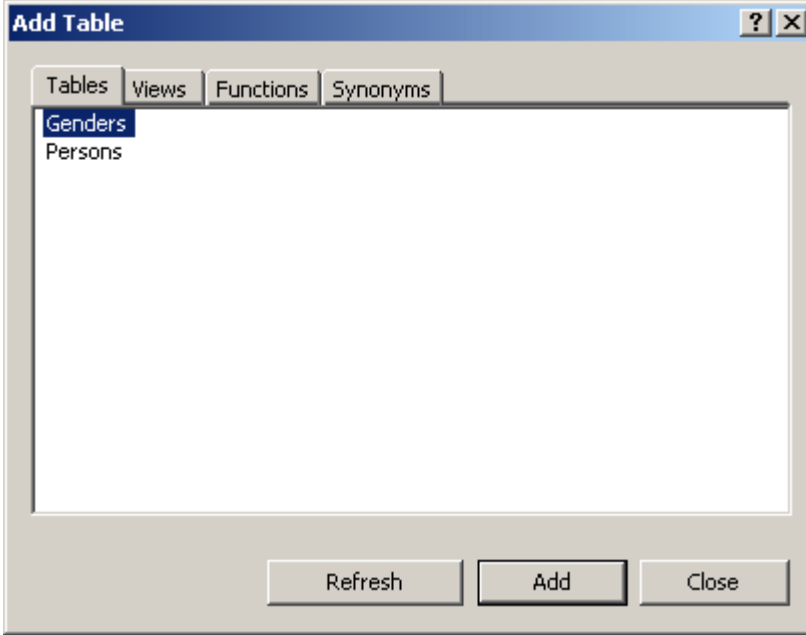
1- قم بتشغيل SQL Server Management Studio واتصل بالخادم

2- تأكد من وجود قاعدة بيانات RealEstate2 التي تم إنشاؤها في درس سابق.

أساسيات قبل إنشاء View:

من أجل إنشاء View، يمكنك استخدام متصفح الكائنات أو إطار الاستعلام. وقبل بدء إنشاء View ينبغي تحديد الجدول (أو الجداول) التي سيتم إدراجها في View.

لإنشاء View من متصفح الكائنات، يمكنك توسيع البند Databases ثم انقر بزر الأيمن على Views ثم اختر New View. وسيفتح مربع الحوار لإضافة الجداول:



يتم العمل على علبة الحوار هذه، بنفس الطريقة التي استعرضناها في درس سابق

- لتحديد الجدول الذي سيتم استخدامه كمصدر للبيانات، انقر على الصفحة Tables.

- إذا أردت إضافة View

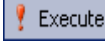
آخر، حدد الصفحة Views من الأعلى، ثم اختر كائن View الذي تود إدراجه.

- إذا أردت استخدام إحدى الدوال لتوليد السجلات، يمكنك العثور عليها في الصفحة Functions.

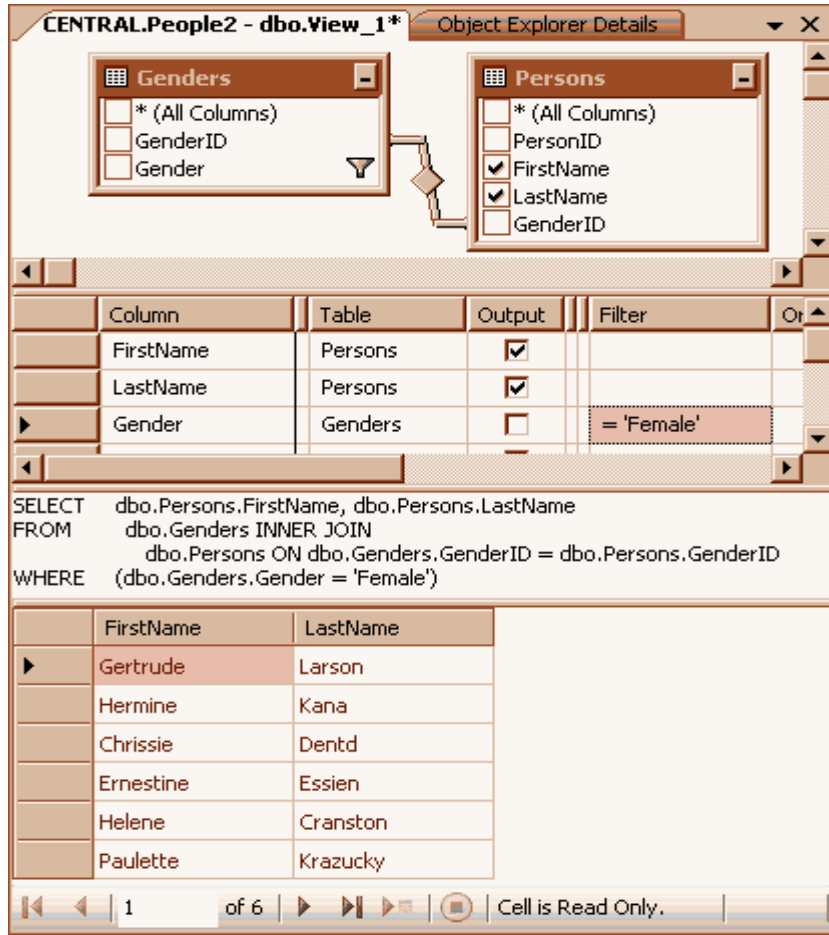
بعد تحديد كائنات المصدر التي تود استخدامها لإنشاء View، يمكنك النقر نقرا مزدوجا عليها أو قم بتحديدتها ثم انقر على Add.

انقر على Close من مربع الحوار بعد إضافة جميع الكائنات التي تحتاجها لبناء View.

كما رأينا في درس بناء الاستعلام، يمكنك إنشاء صلة بين الجداول بسحب أي حقل من جدول وإسقاطه على أي حقل من جدول آخر، كذلك كتابة الاستعلام مباشرة وغير ذلك....

يمكنك في أي وقت اختبار ناتج View بتنفيذ الاستعلام بالنقر على زر التنفيذ .

وبذلك تظهر في القسم السفلي من واجهة تصميم View نتائج الاستعلام. مثال:



من أهم مميزات كائنات Views هو إمكانية استخدامها مرارا وتكرارا. لتحقيق ذلك، يجب حفظ View في قاعدة البيانات. مثل باقي الكائنات في SQL Server يجب أن تمتلك View اسما خاصا بها، وأن يتم حفظها في قاعدة البيانات. لحفظ View في واجهة التصميم يمكن النقر على زر الحفظ Save من شريط الأدوات. يمكنك أيضا غلق واجهة التصميم ثم حفظ View بعد أن تتلقى علبة حوار تأكيد خيار الحفظ، يتبع اسم View نفس القواعد التي رأيناها لتسمية الكائنات في SQL Server.

تطبيق عملي: إنشاء View في واجهة التصميم

1- في متصفح الكائنات، قم بتوسيع البند Databases ثم RealEstate2

2- بالزر الأيمن انقر على Views ثم اختر New View

3- في مربع الحوار Add Table انقر نقرا مزدوجا على PropertyTypes، Properties و Conditions

4- انقر على Close

5- من الجدول PropertyTypes، اسحب الحقل PropertyTypeID وأسقطه على الحقل
PropertyTypeID من الجدول Properties

6- من الجدول Conditions، اسحب الحقل ConditionID وأسقطه على الحقل
ConditionID من الجدول Properties

7- قم بتحديد الحقول التالية:

PropertyNumber, PropertyType, Condition, City, ZIPCode, Bedrooms,
Bathrooms, FinishedBasement, Stories, YearBuilt, MarketValue

8- في قسم المعايير (Criteria)، انقر على PropertyType واضغط Tab 6 مرات، لتصل
إلى عمود التصفية (Filter) واكتب Single Family

9- اضغط Tab، واكتب Townhouse في العمود Or...

سننتج في تسمية كائنات View في هذه الدروس نفس طريقة تسمية الكائنات الأخرى.

تصبح View بعد حفظها في قاعدة البيانات جزءا منها، وسيتم إضافتها تحت البند
الرئيسي Views بالاسم الذي تم حفظها به.

تطبيق عملي: حفظ كائن View

1- لحفظ View، انقر من شريط الأدوات Standard، على الزر Save

2- اكتب اسم View مثلا: Homes، ثم انقر على OK

Object Explorer Details

PropertyTypes

- * (All Columns)
- PropertyTypeID
- PropertyType

Properties

- * (All Columns)
- PropertyID
- PropertyNumber
- Address
- City
- State
- ZIPCode
- PropertyTypeID
- ConditionID
- Bedrooms
- Bathrooms
- FinishedBasement
- IndoorGarage
- Stories
- YearBuilt
- MarketValue

Conditions

- * (All Columns)
- ConditionID
- Condition

Column	Table	Output	Sort Type	Filter	Or...
PropertyNumber	Properties	<input checked="" type="checkbox"/>			
PropertyType	PropertyTy...	<input checked="" type="checkbox"/>		= 'Single Family'	= 'Townhouse'
Condition	Conditions	<input checked="" type="checkbox"/>			

```

SELECT dbo.Properties.PropertyNumber, dbo.PropertyTypes.PropertyType, dbo.Conditions.Condition,
dbo.Properties.City, dbo.Properties.ZIPCode, dbo.Properties.Bedrooms, dbo.Properties.Bathrooms,
dbo.Properties.FinishedBasement, dbo.Properties.Stories, dbo.Properties.YearBuilt,
dbo.Properties.MarketValue
FROM dbo.Conditions INNER JOIN
dbo.Properties ON dbo.Conditions.ConditionID = dbo.Properties.ConditionID INNER JOIN
dbo.PropertyTypes ON dbo.Properties.PropertyTypeID = dbo.PropertyTypes.PropertyTypeID
WHERE (dbo.PropertyTypes.PropertyType = 'Single Family') OR
(dbo.PropertyTypes.PropertyType = 'Townhouse')

```

3- أغلق واجهة تصميم View

4- من متصفح الكائنات، تحت البند RealEstate2، انقر بالزر الأيمن على Views ثم

اختر New View

5- من مربع الحوار Add Table، اختر الجداول PropertyTypes، Properties، Conditions

6- انقر على Close

7- من الجدول PropertyTypes، اسحب الحقل PropertyTypeID وأسقطه على الحقل

PropertyTypeID من الجدول Properties

8- من الجدول Conditions، اسحب ConditionID وأسقطه على الحقل ConditionID من الجدول Properties

9- حدد من كل الجداول الحقول التالية:

PropertyNumber, PropertyType, Condition, City, ZIPCode, Bedrooms, Bathrooms, MarketValue

10- في قسم المعايير، انقر على PropertyType، اضغط على الزر Tab 3 مرات واضغط على الزر Space لإزالة علامة الاختيار من العمود Output

11- اضغط على 3 مرات على Tab، لتصل إلى عمود التصفية، واثم اكتب Condominiums واضغط Enter

12- من القائمة الرئيسية، انقر على Save -> File

13- اكتب اسم View مثلا Condos وانقر على OK

14- أغلق نافذة التصميم.

إنشاء View باستخدام SQL:

لإنشاء View برمجيا، يمكنك استخدام SQL وفقا للجملة التالية:

```
CREATE VIEW ViewName  
AS  
SELECT Statement
```

يمكن أن يقوم SQL Server Management Studio بتوليد هيكل SQL لإنشاء View. للقيام بذلك، أولا أنشئ إطار استعمال جديد، ثم من Template Explorer اسحب وأفلت البند Create View تحت البند الرئيسي View.

SQL إنشاء View يبدأ بعبارة CREATE VIEW متبوعا باسم مناسب، استخدم الكلمة AS بعد ذلك للدلالة على أن ما سيأتي من الاستعلام لتعريف محتوى View.

يتم تعريف View باستخدام عبارة SELECT، بإتباع نفس القواعد التي رأيناها خلال تحليل البيانات. مثال:

```

CREATE VIEW dbo.ListOfMen
AS
SELECT dbo.Genders.Gender ,
        dbo.Persons.FirstName, dbo.Persons.LastName
FROM    dbo.Genders INNER JOIN dbo.Persons
ON      dbo.Genders.GenderID = dbo.Persons.GenderID
WHERE   (dbo.Genders.Gender = 'Male');
GO

```

بعد كتابة استعلام إنشاء View ثم تنفيذه بضغط F5 أو الزر Execute، أي بمجرد تنفيذ عبارة SQL، سيضاف اسم الكائن View بنفس الاسم الذي ورد في الاستعلام، وستتم إضافة إلى قائمة Views من قاعدة البيانات الحالية حتى لو لم تقم بحفظ الاستعلام.

تطبيق عملي: إنشاء View

1- تأكد من وجود قاعدة البيانات Yugo National Bank التي تم إنشاؤها في درس سابق.

2- من متصفح الكائنات، انقر بالزر الأيمن على YugoNationalBank وانقر على New Query

3- لإنشاء كائن View جديد، اكتب ما يلي:

```

Use YugoNationalBank;
GO
CREATE VIEW PayrollPreparation
AS
SELECT  dbo.Employees.EmployeeNumber ,
        dbo.Employees.LastName + ', ' +
        dbo.Employees.FirstName AS [Full Name],
        dbo.Employees.HourlySalary,
        dbo.TimeSheets.TimeSheetCode, dbo.TimeSheets.Week1Monday,
        dbo.TimeSheets.Week1Tuesday, dbo.TimeSheets.Week1Wednesday,
        dbo.TimeSheets.Week1Thursday, dbo.TimeSheets.Week1Friday,
        dbo.TimeSheets.Week1Saturday, dbo.TimeSheets.Week1Sunday,
        dbo.TimeSheets.Week2Monday, dbo.TimeSheets.Week2Tuesday,
        dbo.TimeSheets.Week2Wednesday, dbo.TimeSheets.Week2Thursday,
        dbo.TimeSheets.Week2Friday, dbo.TimeSheets.Week2Saturday,
        dbo.TimeSheets.Week2Sunday
FROM    dbo.Employees INNER JOIN dbo.TimeSheets
ON      dbo.Employees.EmployeeNumber = dbo.TimeSheets.EmplNumber;
GO

```

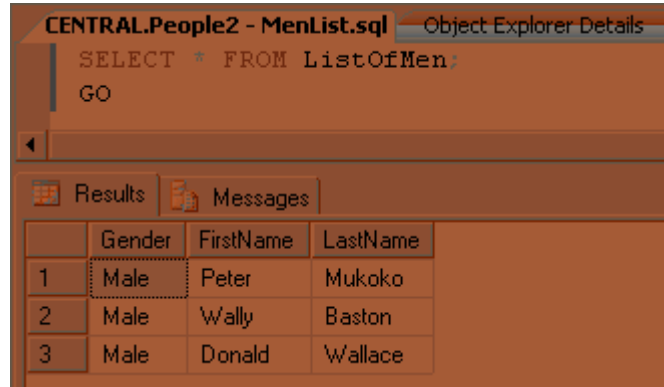
4- قم بتنفيذ الاستعلام بالضغط F5

5- احذف محتوى نافذة الاستعلام.

فتح أو تنفيذ View

بعد إنشاء كائن View، يمكنك عرضه بطريقتين:

- لمشاهدة هيكل View (الذي يمثل الجداول والعلاقات...)، انقر بالزر الأيمن من متصفح الكائنات على الكائن View ثم اختر Design
 - لمشاهدة استعلام SQL الذي يمثل View، انقر بالزر الأيمن من متصفح الكائنات على الكائن View ثم اختر Edit.
- تنفيذ كائن View يعني مشاهدة نتائجه، للقيام بذلك يمكنك:
- فتح نافذة استعلام مرتبطة بقاعدة البيانات التي تحتوي على View. ثم كتابة استعلام SELECT وفق نفس الصيغ والقواعد التي رأينا مع الجداول. مثال:



- من متصفح الكائنات، بالزر الأيمن انقر على الكائن View ثم اختر Open View.

تطبيق عملي: تنفيذ View

1- من متصفح الكائنات، تحت البند RealEste2، قم بتوسيع Views، ثم انقر بالزر الأيمن

على Condos ثم اختر Open View

CENTRAL.RealEs...2 - dbo.Condos		Object Explorer Details					
	PropertyNumber	Condition	City	ZIPCode	Bedrooms	Bathrooms	MarketValue
▶	200417	Excellent	Germantown	NULL	2	1	215495.0000
	NULL	Good	Rockville	NULL	2	2	436885.0000
	288540	Good	Silver Spring	20906	1	1	242775.0000
	NULL	Excellent	Takoma Park	NULL	2	NULL	360885.0000
	NULL	Good	Rockville	NULL	1	1	418885.0000
	811155	Good	Silver Spring	20906	1	1	252775.0000
	NULL	Good	Rockville	NULL	1	1	420555.0000
	644114	Good	Silver Spring	20906	2	2	258445.0000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2- ألق واجهة ناتج View

3- من متصفح الكائنات، انقر بالزر الأيمن على YugoNationalBank ثم اختر New Query

4- لتنفيذ كائن View اكتب استعلام كما يلي:

```
USE YugoNationalBank;
GO
SELECT PayrollPreparation.* FROM PayrollPreparation;
GO
```

5- لتنفيذ الكائن View، اضغط على F5

6- احذف محتوى نافذة الاستعلام.

صيانة كائنات View:

خصائص الكائن View:

يعتبر كائن View في SQL Server غرض (Object) كبقية الأغراض، للأجل ذلك يمكن مشاهدة محتواه، تعديله أو حذفه، ويملك كائن View خصائصه الخاصة. لمشاهدتها يمكنك النقر بالزر الأيمن على كائن View ثم وانقر على Properties، لتشاهد نافذة الخصائص حاملة معلومات مختلفة، مثل اسم قاعدة البيانات وكائن View الذي ينتمي إليها، تاريخ إنشاء View، وغير ذلك...

تطبيق عملي: خصائص View

1- لإنشاء كائن View آخر، اكتب ما يلي:

```
-- =====  
-- Database: YugoNationalBank  
-- View:      TimeSheet  
-- =====  
USE YugoNationalBank  
GO  
  
IF object_id(N'TimeSheet', 'V') IS NOT NULL  
    DROP VIEW dbo.TimeSheet  
GO  
  
CREATE VIEW dbo.TimeSheet  
AS  
SELECT EmplNumber, StartDate,  
       Week1Monday, Week1Tuesday, Week1Wednesday,  
       Week1Thursday, Week1Friday,  
       Week1Saturday, Week1Sunday,  
       Week2Monday, Week2Tuesday, Week2Wednesday,  
       Week2Thursday, Week2Friday, Week2Saturday,  
       Week2Sunday, Notes  
FROM   TimeSheets;
```

2- اضغط F5 لتنفيذ الاستعلام

3- قم بحذف الاستعلام

4- من متصفح الكائنات، تحت البند YugoNationalBank، انقر بالزر الأيمن على

Views واختر Refresh

5- ثم قم بتوسيع Views. انقر بالزر الأيمن على TimeSheet واختر Properties

6- اضغط على Esc لغلق نافذة الخصائص.

تعديل كائن View:

- بعد إنشاء كائن View، يمكنك تغييره أو تعديله حسب الحاجة، لأجل ذلك هناك خياران:
- من متصفح الكائنات، انقر بالزر الأيمن على View، ثم اختر Design، يمكنك حذف أو إضافة الحقول من واجهة التصميم كما يمكنك تغيير أي من الخيارات في أي قسم من واجهة التصميم. بعد التعديلات على الكائن View قم بحفظها وأغلق واجهة التصميم.

- من متصفح الكائنات، انقر بالزر الأيمن على كائن View، اختر Edit، ثم قم بإجراء التعديلات اللازمة ولا تنسى حفظها.

- بطريقة أخرى من متصفح الكائنات، انقر بالزر الأيمن على الكائن View، ثم اختر:

Script View As -> ALTER To -> New Query Editor Window

الصيغة الأساسية برمجة لتعديل View هي:

```
ALTER VIEW ViewName
AS
SELECT Statement
```

تبدأ الاستعلام بعبارَة ALTER VIEW متبوعة باسم كائن View الذي تود تعديله، بعد ذلك أضف الكلمة AS مشيراً إلى أن ما سيأتي من الاستعلام هو الذي سيغير محتوى View، ثم اكتب الاستعلام على النحو الذي تراه مناسباً. مثلاً، يمكنك إنشاء عبارة SELECT التي تتضمن تعديلات أو استعلام جديد تماماً.

يعرض كائن View الذي قمنا بإنشائه قائمة الذكور فقط من جدول الأشخاص، أضفنا أيضاً الحقل Gender، هذا الحقل زائد عن الحاجة، لأننا نعلم مسبقاً أن القائمة لا تضم سوى الذكور. مثال على تغيير View لإزالة (أو بالأحرى إقصاء) الحقل Gender من جدول الأشخاص:

```
ALTER VIEW dbo.ListOfMen
AS
SELECT dbo.Persons.FirstName, dbo.Persons.LastName
FROM   dbo.Genders INNER JOIN dbo.Persons
ON     dbo.Genders.GenderID = dbo.Persons.GenderID
WHERE  (dbo.Genders.Gender = 'Male');
```

حذف كائن View:

يمكن حذف كائن View من قاعدة البيانات إذا لم تكن بحاجة إليه، يمكن ذلك بطرق مختلفة:

- من متصفح الكائنات ومن قاعدة بيانات، انقر بالزر الأيمن على اسم الكائن View الذي تود حذفه وانقر على Delete. وحينها ستتاح لك فرصة لتأكيد خيارك أو لتغيير رأيك.

- من متصفح الكائنات، انقر بالزر الأيمن الكائن View، ثم اختر:

Script View As -> DROP To New Query Editor Window

- بطريقة أخرى، افتح نافذة استعلام مرتبطة بقاعدة البيانات التي تحتوي على كائن View الذي تريد حذفه، من Template Explorer، من البند View، اسحب البند Drop View ثم أسقطه في إطار الاستعلام.

الصيغة العامة لحذف View هي:

```
DROP VIEW ViewName
```

في الجانب الأيمن لعبارة DROP VIEW أضف اسم الكائن View الذي تود حذفه، ثم قم بتنفيذ العبارة لحذف View. ولن تتلقى في هذه الحال أي تحذير من قبل SQL Server.

استخدام الكائن View:

إدخال البيانات باستخدام View:

تخيل أن لديك جدول الموظفين وتريد إنشاء قائمة بأسمائهم فقط. لأجل ذلك، يمكنك إنشاء View على النحو التالي:

```
CREATE VIEW dbo.EmployeesNames
AS
SELECT FirstName,
        LastName,
        LastName + ', ' + FirstName AS FullName FROM Persons;
GO
```

يستند كائن View السابق إلى جدول، حيث يمكنك القيام بإدخال البيانات إلى هذا الجدول من خلال كائن View، عوضا عن إدخال البيانات مباشرة إلى الجدول. للقيام بذلك، يمكنك إتباع نفس القواعد التي استعرضنا في درس سابق. مثال:

```
INSERT INTO dbo.EmployeesNames(FirstName, LastName)
VALUES('Peter', 'Justice');
```

إدخال البيانات باستخدام View يعني إدخال البيانات في الجدول الأصل، وهذا يعني أن الجدول سيتم تحديثه تلقائيا. استنادا إلى هذه الميزة، يمكنك إنشاء View خصيصا لتحديث البيانات في الجدول، بحيث تقوم بعرض الحقول التي يسمح بتحديثها فقط.

تطبيق عملي: إدخال البيانات باستخدام View

1- لإنشاء دالة سنستخدمها لاحقا، أدخل الاستعلام التالي:

```

=====
-- Database: YugoNationalBank
-- Function: CreateTimeSheetCode
-- Purpose: This function takes an employee number
--           and the start date of a time sheet.
--           Then it creates a unique number
--           in the format 0000000000000
--           The first 5 digits represent the
--           employee number,
--           the second 4 digits represent the year,
--           the 2 digits represent the month,
--           that last 2 digits represent the day
=====
USE YugoNationalBank
GO

IF OBJECT_ID (N'dbo.CreateTimeSheetCode') IS NOT NULL
    DROP FUNCTION dbo.CreateTimeSheetCode
GO

CREATE FUNCTION dbo.CreateTimeSheetCode(@EmplNbr varchar(6),
                                         @dteStart datetime)
RETURNS varchar(15)
AS
BEGIN
    DECLARE @strMonth AS varchar(20);
    DECLARE @strDay AS varchar(20);
    DECLARE @iMonth AS int;
    DECLARE @iDay AS int;
    DECLARE @strTimeSheetCode varchar(20);

    SET @iMonth = CONVERT(varchar(20), MONTH(@dteStart));
    SET @iDay = CONVERT(varchar(20), DAY(@dteStart));

    IF @iMonth < 10
        SET @strMonth = CONVERT(varchar(20), YEAR(@dteStart)) +
            '0' + CONVERT(varchar(20), @iMonth);
    ELSE
        SET @strMonth = CONVERT(varchar(20), YEAR(@dteStart)) +
            CONVERT(varchar(20), @iMonth);

    IF @iDay < 10
        SET @strDay = @strMonth + '0' +
            CONVERT(varchar(20), @iDay);
    ELSE
        SET @strDay = @strMonth + CONVERT(varchar(2), @iDay);

    SET @strTimeSheetCode = @EmplNbr + @strDay;
    RETURN @strTimeSheetCode;
END
GO

```

3- اضغط F5 لتنفيذ الاستعلام

4- لإدخال البيانات باستخدام View، أكتب الاستعلام التالي:

```

USE YugoNationalBank
GO
-- The following code performs data entry using a view
INSERT INTO dbo.TimeSheet

```

```

VALUES('46288', '1/1/2007',
       0.00, 8.50, 9.50, 8.50, 9.00, 0.00, 0.00,
       10.00, 9.50, 8.50, 10.50, 9.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('66286', '1/1/2007',
       0.00, 8.50, 6.50, 5.50, 6.50, 0.00, 0.00,
       4.00, 6.00, 6.50, 6.00, 5.50, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('92493', '1/1/2007',
       0.00, 8.00, 9.00, 8.50, 9.50, 0.00, 0.00,
       5.50, 6.50, 4.50, 6.00, 4.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('27199', '1/15/2007',
       6.00, 8.50, 0.00, 4.00, 6.50, 0.00, 0.00,
       4.00, 0.00, 6.00, 4.00, 0.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('39538', '1/15/2007',
       8.00, 8.00, 6.00, 8.50, 6.00, 0.00, 0.00,
       9.50, 10.50, 8.00, 8.00, 8.50, 0.00, 0.00,
       'There were a few missing times in the time sheet. ' +
       'They have been recorded.');
```

```

GO
INSERT INTO dbo.TimeSheet
VALUES('40550', '1/15/2007',
       8.50, 8.00, 0.00, 8.50, 0.00, 0.00, 0.00,
       6.00, 6.50, 6.50, 0.00, 4.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('66286', '1/29/2007',
       8.00, 6.50, 9.50, 8.00, 7.50, 0.00, 0.00,
       10.50, 9.50, 8.50, 8.00, 10.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('90026', '2/12/2007',
       8.50, 6.50, 8.00, 8.00, 9.50, 0.00, 0.00,
       9.50, 8.00, 8.50, 8.00, 8.00, 0.00, 0.00, '');
GO
INSERT INTO dbo.TimeSheet
VALUES('92493', '2/12/2007',
       4.00, 6.50, 5.50, 8.00, 6.50, 0.00, 0.00,
       8.00, 8.00, 8.00, 6.00, 8.00, 0.00, 0.00, '');
GO
-- The following code updates a table using a function
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('46288', '1/1/2007')
WHERE (EmplNumber = '46288') AND (StartDate = '1/1/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('66286', '1/1/2007')
WHERE (EmplNumber = '66286') AND (StartDate = '1/1/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('92493', '1/1/2007')
WHERE (EmplNumber = '92493') AND (StartDate = '1/1/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('27199', '1/15/2007')
```

```

WHERE (EmplNumber = '27199') AND (StartDate = '1/15/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('39538', '1/15/2007')
WHERE (EmplNumber = '39538') AND (StartDate = '1/15/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('40550', '1/15/2007')
WHERE (EmplNumber = '40550') AND (StartDate = '1/15/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('66286', '1/29/2007')
WHERE (EmplNumber = '66286') AND (StartDate = '1/29/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('90026', '2/12/2007')
WHERE (EmplNumber = '90026') AND (StartDate = '2/12/2007');
GO
UPDATE dbo.TimeSheets
SET TimeSheetCode = dbo.CreateTimeSheetCode('92493', '2/12/2007')
WHERE (EmplNumber = '92493') AND (StartDate = '2/12/2007');
GO

```

5- اضغط F5 لتنفيذ الاستعلام

6- قم بحذف محتوى نافذة الاستعلام.

استخدام View باسم مستعار:

إن كائن View لا يمثل دالة SQL Server لكنه يمكن أن يستخدم كدالة. والمعلومات هنا لا تعني قيم يمكن تمريرها إلى View لكن هذه القيم ستكون محددة فقط عند إنشاء الكائن View، فهي ليست معلومات حقيقية.

عند إنشاء View، يمكنك إنشاء خانة لتمرير قيمة الحقول بين قوسين. وفق التالي:

```

CREATE VIEW CarIdentifier([Tag #], Manufacturer, [Type of Car],
Available)
. . .

```

هذه التقنية تسمح لك بتحديد أسماء الحقول وفق اختيارك. ويمكنك أيضا استخدام الاسم الفعلي للحقول، إذا كان اسم كلمة واحدة يمكنك كتابته مباشرة، وإذا كان يتضمن عدة كلمات يجب ضمها بين معقوفتين [].

بعد كتابة أسماء الحقول، يجب استخدام عدد المعلومات بنفس عدد الحقول، حيث تمثل كل معلمة أحد الحقول على الترتيب، مثال:

```

CREATE VIEW dbo.MenAndWomen([First Name], [Last Name], Gender)
AS

```

```

SELECT dbo.Persons.FirstName,
       dbo.Persons.LastName,
       dbo.Genders.Gender
FROM   dbo.Genders INNER JOIN dbo.Persons
ON     dbo.Genders.GenderID = dbo.Persons.GenderID;
GO

```

ولأن (كما ذكرنا سابقاً) كائن View لا يمثل دالة SQL Server حيث لا يمكن تمرير قيم مباشرة إلى View، فإن تنفيذ View لا يتطلب تحديد قيم المعلمات، ويتم تنفيذها فقط بكتابة استعلام **SELECT**، مثال:

	First Name	Last Name	Gender
1	Gertrude	Larson	Female
2	Peter	Mukoko	Male
3	Wally	Baston	Male
4	Donald	Wallace	Male
5	Hermine	Kana	Female
6	Chrissie	Dentd	Female
7	Ernestine	Essien	Female
8	Helene	Cranston	Female
9	Robert	Palau	Unknown
10	Paulette	Krazucky	Female

كائن View والمعايير:

من أهم خصائص الاستعلام أنه يتميز بالمرونة، حيث يمكنك الوصول إلى أية نتيجة من خلال بيانات معقدة عن طريق معالجة البيانات باستخدام الشروط. يمكن استخدام الشروط لعرض محتوى View بدلاً من الجداول التي من شأنها تعقيد الاستعلام أكثر. يمكنك عند إنشاء View ضمن عبارة **SELECT**، تحديد الحقول وفق ترتيب معين، وإضافة معايير لاستبعاد بعض السجلات. مثال:

CENTRAL.Peopl...jects\maw.sql Object Explorer Details

```

SELECT * FROM dbo.MenAndWomen
WHERE (Gender = 'Male') OR (Gender = 'Female');
GO

```

Results Messages

	First Name	Last Name	Gender
1	Gertrude	Larson	Female
2	Peter	Mukoko	Male
3	Wally	Baston	Male
4	Donald	Wallace	Male
5	Hermine	Kana	Female
6	Chrissie	Dentd	Female
7	Ernestine	Essien	Female
8	Helene	Cranston	Female
9	Paulette	Krazucky	Female

كائن View والتوابيع:

يمكن إنشاء View أكثر تعقيدا أو ذات وظائف متقدمة، ويمكن إشراك التوابيع المبرمجة كما يمكن استخدام التوابيع المضمنة مع SQL Server.

إذا لم تكن التوابيع المضمنة تفي بالغرض، يمكنك إنشاء تابع خاص بك. مثال:

```

USE People;
GO
CREATE FUNCTION dbo.GetFullName
(
    @FName varchar(20),
    @LName varchar(20)
)
RETURNS varchar(41)
AS
BEGIN
    RETURN @LName + ', ' + @FName;
END
GO

```

بعد إنشاء التابع الذي تريد، يمكنك إدراجه في جسم استعلام View إذا لزم الأمر. مثال:

```

CREATE VIEW dbo.MyPeople
AS
SELECT dbo.GetFullName(FirstName, LastName) AS [Full Name],
       dbo.Genders.Gender
FROM   Genders INNER JOIN dbo.Persons
ON     dbo.Genders.GenderID = dbo.Persons.GenderID;

```

CENTRAL.Peopl... dbo.MyPeople		Object Explorer Details
	Full Name	Gender
▶	Larson, Gertrude	Female
	Mukoko, Peter	Male
	Baston, Wally	Male
	Wallace, Donald	Male
	Kana, Hermine	Female
	Dentd, Chrissie	Female
	Essien, Ernestine	Female
	Cranston, Helene	Female
	Palau, Robert	Unknown
	Krazucky, Paulette	Female
*	NULL	NULL

تطبيق عملي: استخدام تابع ضمن View

1- من متصفح الكائنات، تحت YugoNationalBank قم بتوسيع Views إذا لزم الأمر، ثم انقر بالزر الأيمن PayrollPreparation واختر:

Script View As -> ALTER To -> New Query Editor Window

2- اكتب الاستعلام التالي:

```
USE [YugoNationalBank]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER VIEW [dbo].[PayrollPreparation]
AS
SELECT  dbo.Employees.EmployeeNumber,
        dbo.Employees.LastName + ', ' +
        dbo.Employees.FirstName AS [Full Name],
        dbo.Employees.HourlySalary,
        dbo.TimeSheets.TimeSheetCode, dbo.TimeSheets.Week1Monday,
        dbo.TimeSheets.Week1Tuesday, dbo.TimeSheets.Week1Wednesday,
        dbo.TimeSheets.Week1Thursday, dbo.TimeSheets.Week1Friday,
        dbo.TimeSheets.Week1Saturday, dbo.TimeSheets.Week1Sunday,
        dbo.TimeSheets.Week2Monday, dbo.TimeSheets.Week2Tuesday,
        dbo.TimeSheets.Week2Wednesday, dbo.TimeSheets.Week2Thursday,
        dbo.TimeSheets.Week2Friday, dbo.TimeSheets.Week2Saturday,
        dbo.TimeSheets.Week2Sunday,

        CAST(dbo.TimeSheets.Week1Monday AS SmallMoney) +
        CAST(dbo.TimeSheets.Week1Tuesday AS SmallMoney) +
        CAST(dbo.TimeSheets.Week1Wednesday AS SmallMoney) +
        CAST(dbo.TimeSheets.Week1Thursday AS SmallMoney) +
        CAST(dbo.TimeSheets.Week1Friday AS SmallMoney) +
        CAST(dbo.TimeSheets.Week1Saturday AS SmallMoney) +
```

```

CAST(dbo.TimeSheets.Week1Sunday AS SmallMoney)
AS [Total Week1],
CAST(dbo.TimeSheets.Week2Monday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Tuesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Wednesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Thursday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Friday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Saturday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Sunday AS SmallMoney)
AS [Total Week2],
CAST(dbo.TimeSheets.Week1Monday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Tuesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Wednesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Thursday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Friday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Saturday AS SmallMoney) +
CAST(dbo.TimeSheets.Week1Sunday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Monday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Tuesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Wednesday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Thursday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Friday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Saturday AS SmallMoney) +
CAST(dbo.TimeSheets.Week2Sunday AS SmallMoney)
AS [Total Time Sheet]
FROM dbo.Employees INNER JOIN dbo.TimeSheets
ON dbo.Employees.EmployeeNumber = dbo.TimeSheets.EmplNumber;

```

3- لتحديث الاستعلام، من القائمة الرئيسية انقر على Query ثم Execute

4- احذف محتوى نافذة الاستعلام

5- لمشاهدة نتيجة الاستعلام، اكتب ما يلي:

```

Use YugoNationalBank;
GO
USE YugoNationalBank;
GO
SELECT PayrollPreparation.* FROM PayrollPreparation;
GO

```

6- احذف الاستعلام.

CENTRAL.YugoN...LQuery15.sql* Object Explorer Details

```

Use YugoNationalBank;
GO
USE YugoNationalBank;
GO
SELECT PayrollPreparation.* FROM PayrollPreparation;
GO

```

Results Messages

	Empl...	Full Name	HourlySalary	TimeSheetCode	Week1Mond...	Week1Tuesd...
1	46288	Marconi, Catherine	22.55	4628820070101	0.00	8.50
2	27199	Ankoma, Leonie	14.88	2719920070115	6.00	8.50
3	66286	Zbrnitz, Lienev	15.75	6628620070101	0.00	8.50
4	66286	Zbrnitz, Lienev	15.75	6628620070129	8.00	6.50
5	39538	Waste, Plant	16.75	3953820070115	8.00	8.00
6	92493	Roberts, Kirsten	18.05	9249320070101	0.00	8.00
7	92493	Roberts, Kirsten	18.05	9249320070212	4.00	6.50
8	90026	Lamy, Roger	10.24	9002620070212	8.50	6.50
9	40550	Kirkland, Lorraine	12.86	4055020070115	8.50	8.00

الفهارس Indexes

أساسيات:

في الصفحات الأخيرة من أي كتاب تجد سلسلة من العنواين يقابل كل عنصر منها رقم صفحة، تسمى هذه القائمة بفهرس الكتاب. يسمح فهرس الكتاب بالوصول إلى أي جزء (أو صفحة) منه بسرعة وسهولة، وهذا هو مبدأ عمل الفهارس Indexes.

تماما مثل الكتب؛ يمكن للجدول (أو View) استخدام آلية عمل الفهرس. الفهرس هو الحقل (أو مجموعة من الحقول) التي يمكن استخدامها لتحديد (أو تطبيق عمليات محددة على) السجلات استنادا إلى الصلاحيات المخولة على هذه السجلات.

تطبيق عملي: تمهيد إلى الفهارس

- 1- بعد تشغيل SQL Server Management Studio، قم بالاتصال بالخادم.
- 2- لإنشاء قاعدة بيانات جديدة، انقر بالزر الأيمن من متصفح الكائنات على Databases، ثم على New Database...
- 3- أدخل الاسم CollinsUniversity وانقر على OK
- 4- من متصفح الكائنات، قم بتوسيع البند CollinsUniversity
- 5- انقر بالزر الأيمن على Tables واختر New Table...
- 6- أنشئ الحقول على النحو التالي:

Column Name	Data Type	Allow Nulls
StudentNumber	int	Unchecked
FirstName	nvarchar(32)	Checked
MI	char(1)	Checked
LastName	nvarchar(32)	Unchecked
Username	nvarchar(20)	Unchecked
Password	nvarchar(20)	Checked

7- أغلق نافذة تصميم الجدول

8- انقر على Yes لتأكيد خيار حفظ الجدول

9- أكتب اسم الجدول StudentRegistration وانقر على OK

10- من متصفح الكائنات، انقر بالزر الأيمن على Databases، ثم New Query...

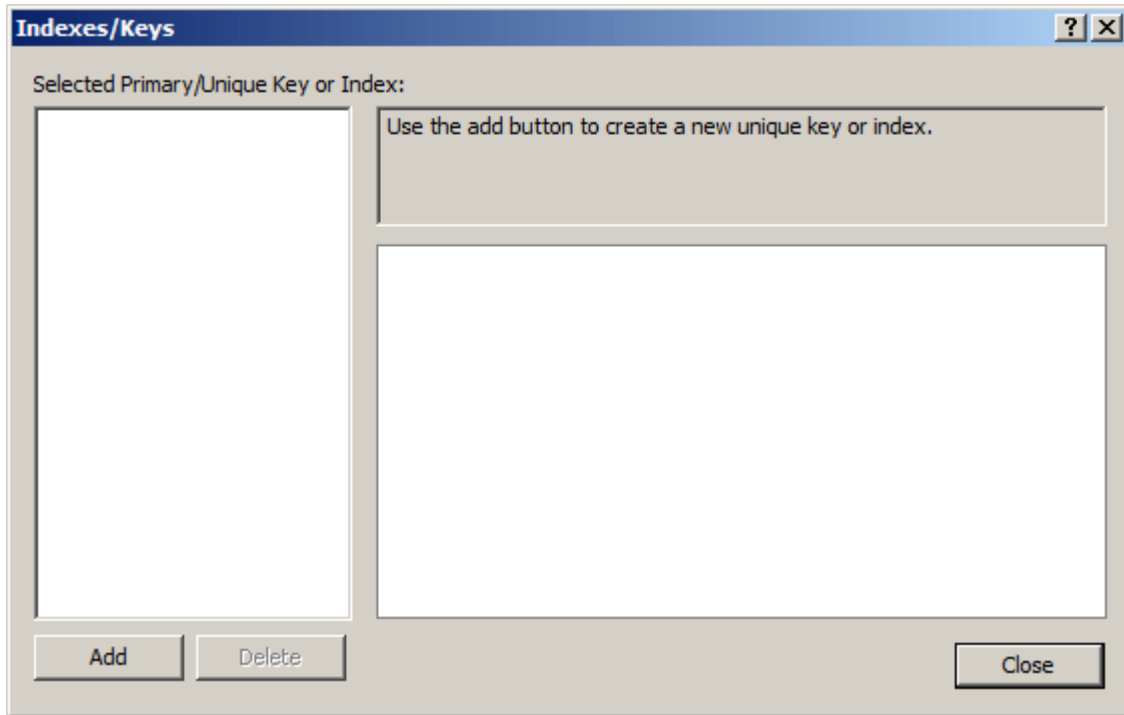
```
CREATE DATABASE CeilInn1;  
GO  
USE CeilInn1;  
CREATE TABLE Rooms  
(  
    RoomNumber nvarchar(10),  
    LocationCode nchar(10) not null,  
    LocationName nvarchar(50) null,  
    RoomType nvarchar(50)  
);  
GO
```


11- قم بتنفيذ الاستعلام ثم أغلق نافذة الاستعلام.

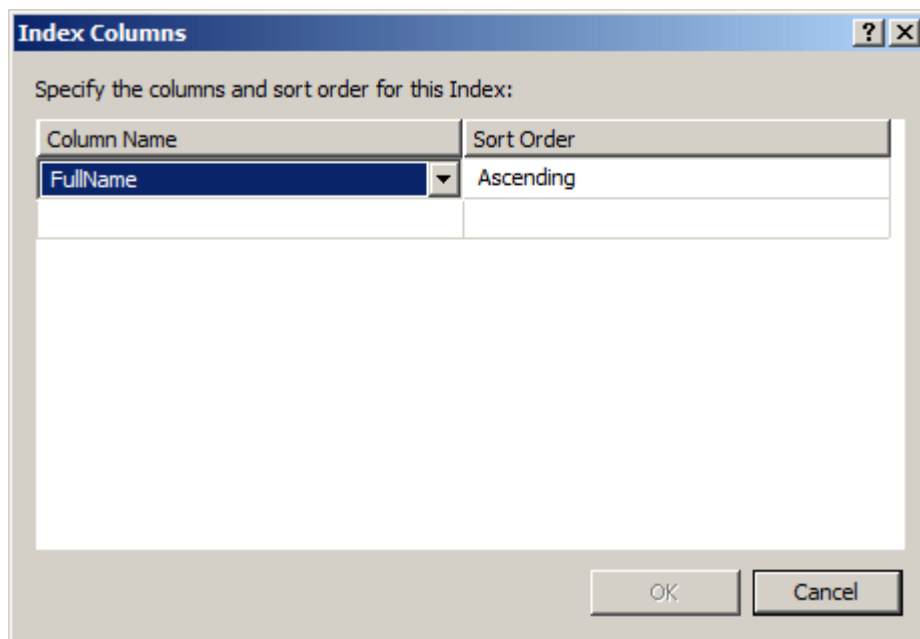
إنشاء فهرس أثناء التصميم:

محرك قاعدة البيانات SQL Server مجهز بآلية تقوم تلقائياً بإنشاء الفهرس على الجدول اعتماداً على طريقة إنشائه. مثال، إذا قمت بإنشاء مفتاح أساسي على جدول، فإن محرك قاعدة البيانات سيقوم تلقائياً بإنشاء فهرس لحقل المفتاح الأساسي. ويمكنك رغم ذلك، إنشاء فهرس وفق اختيارك أثناء التصميم أو باستخدام SQL.

لإنشاء فهرس أثناء التصميم، يمكنك استخدام نافذة Indexes/Keys. للقيام بذلك، قم بتوسيع بند قاعدة البيانات من متصفح الكائنات التي تحمل الجدول (أو View) المحدد، ثم قم بتوسيع البند Tables، ثم بالزر الأيمن انقر على الجدول الذي تريد إنشاء الفهرس عليه، ثم اختر Design، في النافذة، انقر بالزر الأيمن واختر Indexes/Keys... لتواجه مربع الحوار:



لإنشاء فهرس، انقر على زر الإضافة Add ثم أكتب اسم الفهرس، يقوم محرك قاعدة بيانات باقتراح اسم بشكل افتراضي، إذا كنت تريد تغييره انقر على (Name) ثم اكتب الاسم الذي تريده. بعد ذلك يجب تحديد حقل واحد على الأقل. افتراضياً، سيقوم محرك قاعدة البيانات بتحديد الحقل الأول من الجدول. إذا أردت تغييره انقر على Columns ثم على زر التفاصيل  لتظهر شاشة بهذا الشكل:



من تحت العمود Column Name، يمكنك تحديد الحقول. عند الانتهاء من ذلك انقر على OK.

تطبيق عملي: إنشاء فهرس أثناء التصميم

1- من متصفح الكائنات، تحت البند Tables لقاعدة CollinsUniversity، انقر بالزر

الأيمن على StudentRegistration واختر Design

2- بالزر الأيمن انقر على أي مكان في الفراغ الأبيض، واختر Indexes/Keys...

3- انقر على زر الإضافة Add

4- انقر على Columns ثم على زر التفاصيل 

5- انقر على قائمة الخيارات حيث StudentNumber خيار افتراضي وحدد LastName

6- انقر على قائمة الخيارات تحت LastName

7- انقر على السهم من مربع الخيارات وحدد Username

8- انقر على OK

9- انقر على (Name) ثم اكتب IX_Credentials

10- انقر على Close.

إنشاء الفهارس باستخدام SQL:

يتم إنشاء فهرس باستخدام SQL، وفق الصيغة الأساسية التالية:

```
CREATE INDEX IndexName ON Table/View(Column(s))
```

يمكنك أيضا، بدلا من ذلك، فتح إطار استعلام، ثم من Templates Explorer، قم بسحب

وإفلات البند Index Basis (أو بند فرعي مشابه). وسيتم توليد استعلام هيكلية لإنشاء فهرس:

```
-- =====  
-- Create index basic template  
-- =====  
USE <database_name, sysname, AdventureWorks>  
GO  
  
CREATE INDEX <index_name, sysname, ind_test>
```



```

ON <schema_name, sysname, Person>.<table_name, sysname, Address>
(
    <column_name1, sysname, PostalCode>
)
GO

```

يبتدئ استعلام إنشاء فهرس بعبارة **CREATE INDEX**، متبوعاً باسم مناسب للفهرس، ثم كلمة **ON** يليها اسم الجدول أو View التي تريد إنشاء الفهرس عليها. يتبع كل ذلك قوسين يكتب بداخلهما على الأقل اسم حقل واحد. مثال:

```

USE Exercisel;
GO
CREATE INDEX IX_Employees
ON Employees(EmployeeNumber);
GO

```

إذا كان الفهرس يشمل عدة حقول، يجب الفصل بينها بفاصلة. مثال:

```

USE Exercisel;
GO
CREATE INDEX IX_Employees
ON Employees(LastName, Username);
GO

```

تطبيق عملي: إنشاء فهرس باستخدام SQL

1- اكتب الاستعلام الآتي في نافذة الاستعلام:

```

USE CeilInn1;
GO
CREATE INDEX IX_Rooms
ON Rooms(RoomNumber, LocationCode);
GO

```

2- اضغط على F5 للتنفيذ، ثم أغلق نافذة الاستعلام.

صيانة الفهرس:

يتم التعامل مع الفهارس بصفحتها كائنات من قاعدة البيانات، إذ يمكن مراجعتها وتعديلها أو حذفها إذا اقتضى الأمر.

حذف الفهرس:

إذا لم تكن بحاجة إلى أي فهرس يمكنك حذفه، يمكنك بذلك من خلال واجهة التصميم أو باستخدام SQL.

لحذف الفهرس أثناء التصميم، افتح الجدول في واجهة التصميم ثم بالزر الأيمن، انقر على أي مكان في إطار الجدول واختر Indexes/Views. في الإطار الأيسر، حدد الفهرس ثم انقر على زر الحذف Delete، ولن تتلقى أي تحذير قبل الحذف، لذلك تأكد من الفهرس المحدد.

الصيغة العامة لحذف فهرس باستخدام SQL هي كما يلي:

```
DROP INDEX IndexName ON TableName;
```

حيث TableName هو اسم الجدول الذي يحتوي على الفهرس. IndexName هو اسم الفهرس الذي تريد التخلص منه. مثال:

```
USE Exercisel;  
GO  
DROP INDEX IX_Students  
ON Students;  
GO
```

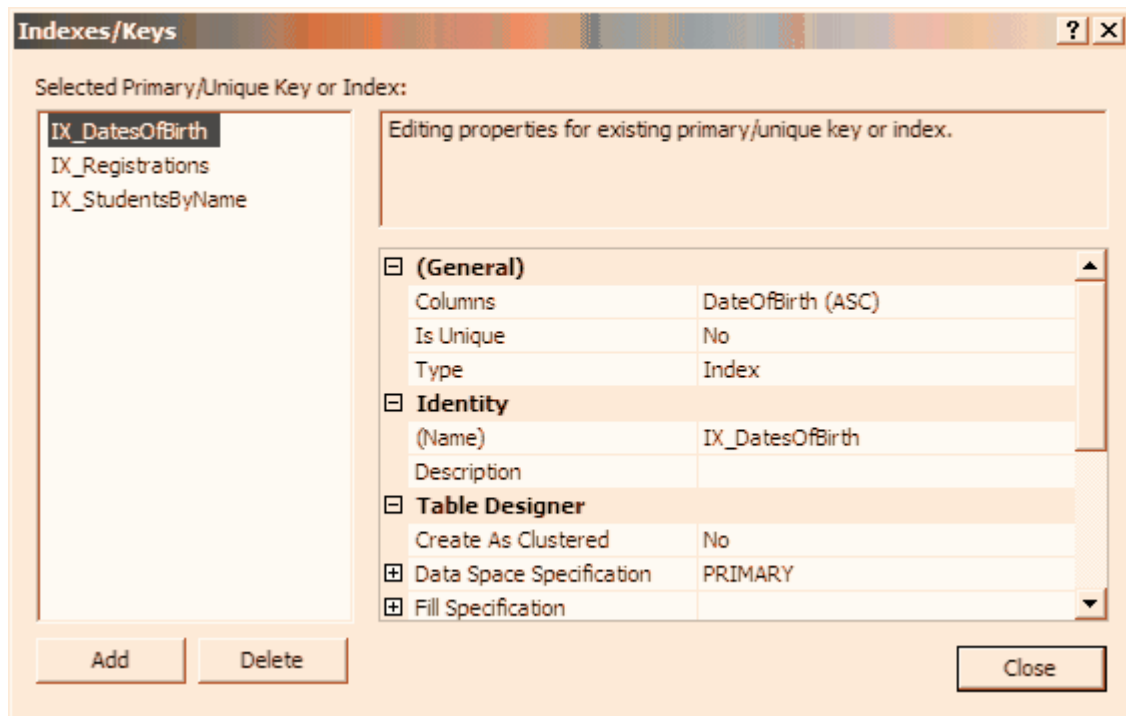
أنواع الفهارس:

يوفر SQL Server أنواعا مختلفة من الفهارس، أشهرها نوعان: الفهرس المتكثّل -أو المجمعّ - (Clustered Index) والفهرس غير المتكثّل (Nonclustered Index).

الفهرس المتكثّل (Clustered Index):

رأينا أن إنشاء الفهرس يتم باستخدام حقل أو أكثر من الجدول، يعني ذلك أنه عندما يتعلق الأمر باستخدام فهرس، فإن القيم التي ستستخدم للتأشير هي نفسها القيم المخزنة في هذا في الحقل (أو هذه الحقول)، لهذا يعد هذا الفهرس متكثّل أو متجمع (Clustered) في قيم هذا الحقل. وتسمى الحقول التي تشكل في مجموعها فهرسا بالمفاتيح.

يتم إنشاء الفهرس المتجمع أثناء التصميم، من خلال واجهة Indexes/Keys، قم بتحديد الفهرس أولاً، ثم انقر من القائمة اليمنى لتغيير الخيار Create As Clustered إلى القيمة Yes:



عند الانتهاء انقر على Close.

لإنشاء فهرس متجمع باستخدام SQL يمكن استخدام الصيغة التالية:

```
CREATE CLUSTERED INDEX IndexName ON Table/View(Column(s))
```

الكلمة الوحيدة الجديدة في هذه الصيغة هي CLUSTERED. مثال:

```
USE Exercisel;
GO
CREATE TABLE dbo.Students
(
    StudentID int identity(1,1) NOT NULL,
    FirstName nvarchar(40) NULL,
    LastName nvarchar(40) NOT NULL,
)
GO
```

```
CREATE CLUSTERED INDEX IX_Student
ON Students(LastName);
GO
```

يسمى الجدول الذي يحتوي على فهرس متكامل بجدول متكامل أيضا (Clustered Table).

من مميزات الفهرس المتكامل، ما يلي:

- جعل البحث عن السجلات والفرز أكثر سهولة، وهو ما يسهل على محرك قاعدة البيانات عملية الطواف حول البيانات للوصول إلى السجلات المطلوبة.

- ليس هذا شرطا، لكن ينبغي أن تكون السجلات فريدة (سبق وأن رأينا كيف يتم ذلك، باستخدام قيود التحقق؛ وسنرى لاحقا طريقة إنشاء سجلات فريدة).
- يجب ألا يكون في الجدول أكثر من فهرس متكامل واحد، وإذا قمت بإنشاء فهرس متكامل آخر، فسيتم حذف الفهرس المتكامل السابق.

الفهرس غير المتكامل (Nonclustered Index):

بينما يستخدم الفهرس المتكامل قائمة مرتبة من السجلات، يقوم نوع آخر من الفهارس باستخدام آلية مختلفة لا تستند إلى قائمة من السجلات المرتبة ولكن على بيانات مرجعية، وهو ما يعرف بالفهرس غير المتكامل (Nonclustered). على عكس النوع السابق، يمكن للجدول أن يحتوي على أكثر من فهرس غير متكامل.

لإنشاء فهرس غير متكامل أثناء التصميم، قم بعرض نافذة Indexes/Keys، ثم قم بتحديد الفهرس وغير خاصية Create As Clustered إلى القيمة No.

لإنشاء فهرس Nonclustered باستخدام SQL، يمكنك استخدام الصيغة التالية:

```
CREATE NONCLUSTERED INDEX IndexName ON Table/View(Column(s))
```

الكلمة الجديدة في هذه الصيغة هي NONCLUSTERED، وبقية العبارات هي نفسها التي تم شرحها سابقا، مثال:

```
CREATE NONCLUSTERED INDEX IX_Student
ON Students(LastName, StudentNumber);
GO
```

إدخال البيانات وتحليلها مع استخدام فهرس:

رأينا أن وجود الفهارس يمكن أن يفرض بعض الإجراءات لإدخال البيانات، مثل فرض وجود قيم فريدة في لكل سجل من الحقل المفهرس، إلى جانب ذلك، فإن الفائدة من الفهارس في الواقع تتضح عندما يتعلق الأمر بتحليل البيانات.

فهارس فريدة (Unique Indexes):

يتم اعتماد الفهرس بطريقتين، إما بفرز السجلات؛ وهنا يقوم الفهرس المتكفل بالتكفل بهذا الجانب، لأنه يقوم تلقائيا وبشكل داخلي بترتيب السجلات.

لكن ماذا لو لم تكن السجلات فريدة؟ مثال، إذا تم إدخال بيانات بشكل خاطئ في قائمة الطلاب، بحيث يكون لطلابين أو أكثر نفس البيانات، إذا قمت بإنشاء فهرس لمثل هذا الجدول، فسيقوم محرك قاعدة البيانات بإنشاء سجلات مكررة على الفهرس، وهذا سيء؛ لأنه يتطلب زمنا إضافيا لتحديد السجلات، ويصبح الأمر أسوأ مع عدد كبير جدا من السجلات.

عند إنشاء الجدول، يمكنك فهرسته وجعل الفهرس يطبق القواعد التي من شأنها أن تجعل السجلات فريدة في الجدول، لمراعاة ذلك، يمكنك تطبيق قاعدة لجعل بيانات الفهرس فريدة.

للقيام بذلك، عند إنشاء الفهارس أثناء تصميم الجدول في الواجهة Indexes/Keys، حدد الفهرس من الجانب الأيسر. ثم في اللائحة على اليمين غير قيمة الخاصية Is Unique إلى Yes، يمكنك في أي وقت إزالة هذه الخاصية بتغيير القيمة إلى No.

لإنشاء فهرس فريد باستخدام SQL، يمكنك إدراج الكلمة UNIQUE في الاستعلام بهذا الشكل:

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name ON  
Table/View(Column(s))
```

ابدأ بكلمة CREATE ثم اتبعها بعبارة UNIQUE، ثم تحديد فيما إذا كان الفهرس

Clustered أم لا. بقية العبارات رأيناها سابقا¹. مثال:

```
-- =====  
-- Table: Employees  
-- =====  
USE Exercisel  
GO
```

```
CREATE TABLE dbo.Employees  
(  
    EmployeeNumber nvarchar(10),  
    DateHired datetime NULL,  
    FirstName nvarchar(40) NULL,  
    LastName nvarchar(40) NOT NULL,  
    HourlySalary decimal(6, 2)  
)  
GO
```

¹ المعقوفتين الواردتين في هذه الصيغة للدلالة على العبارات الاختيارية، والرمز | يدل على إمكانية استخدام أحد العبارتين، هذه الصيغة النحوية (باستخدام هذه الرموز) هي الواردة في ملفات المساعدة في SQL Server.

```
CREATE UNIQUE INDEX IDX_EmployeesIDs
ON dbo.Employees(EmployeeNumber, LastName);
GO
```

إذا جعلت السجلات فريدة في الفهرس، فسيتلقى مستخدم البرنامج أثناء إدخال البيانات، إذا

قام بإدخال قيمة مكررة في الجدول، رسالة خطأ. مثال:

```
-- =====
-- Create index basic template
-- =====
USE Exercisel;
GO
```

```
CREATE TABLE dbo.Employees
(
    EmployeeNumber integer NOT NULL,
    FirstName varchar(50) NULL,
    LastName varchar(50) NOT NULL,
    HourlySalary decimal(6, 2)
)
GO
```

```
CREATE UNIQUE INDEX IDX_EmployeeNumber
ON dbo.Employees(EmployeeNumber);
GO
```

```
INSERT INTO Employees(EmployeeNumber, FirstName,
                        LastName, HourlySalary)
VALUES(92935, 'Joan', 'Hamilton', 22.50)
GO
```

```
INSERT INTO Employees(EmployeeNumber, FirstName,
                        LastName, HourlySalary)
VALUES(22940, 'Peter', 'Malley', 14.25)
GO
```

```
INSERT INTO Employees(EmployeeNumber, FirstName,
                        LastName, HourlySalary)
VALUES(27495, 'Christine', 'Fink', 32.05)
GO
```

```
INSERT INTO Employees(EmployeeNumber, FirstName,
                        LastName, HourlySalary)
VALUES(22940, 'Gertrude', 'Monay', 15.55)
GO
```

```
INSERT INTO Employees(EmployeeNumber, FirstName,
                        LastName, HourlySalary)
VALUES(20285, 'Helene', 'Mukoko', 26.65)
GO
```

عند تنفيذ هذا الاستعلام نحصل على:

```
(1 row(s) affected)
(1 row(s) affected)
(1 row(s) affected)
Msg 2601, Level 14, State 1, Line 1
Cannot insert duplicate key row in object 'dbo.Employees'
with unique index 'IDX_EmployeeNumber'.
The statement has been terminated.
(1 row(s) affected)
```

القوادر Triggers

عند القيام بإحدى عمليات الإضافة أو الحذف أو التعديل على سجل (أو سجلات) من الجدول، يقوم الجدول في SQL Server بإنتاج إخطار أو إشعار. ونقول حينها أن الجدول قام بإطلاق حدث، يمكنك استخدام هذا الحدث لاتخاذ بعض الإجراءات.

القادر (Trigger) هو عملية يتم تنفيذها وراء الكواليس عند تطبيق حدث على الجدول.

تطبيق عملي: تمهيد لإنشاء القوادر

1- قم بتشغيل SQL Server Management Studio

2- من شريط الأدوات قياسي، انقر على New Query

3- سنقوم بإنشاء قاعدة بيانات جديدة تحتوي على جدول واحد، اكتب ما يلي:

```
-- =====
-- Database:    CeilInn4
-- =====

IF EXISTS(SELECT name FROM sys.databases
          WHERE name = N'CeilInn4b')
DROP DATABASE CeilInn4;
GO
CREATE DATABASE CeilInn4;
GO

USE CeilInn4;
GO

IF OBJECT_ID('Rooms', 'U') IS NOT NULL
    DROP TABLE Rooms
GO

-- =====
-- Database:    CeilInn4
-- Table:      Rooms
-- Description: This table is used to hold information
--              about his room rented for the hotel
-- =====
CREATE TABLE Rooms
(
    RoomNumber nvarchar(10),
    LocationCode nchar(10) default N'Silver Spring',
    RoomType nvarchar(20) default N'Bedroom',
    BedType nvarchar(40) default N'Queen',
    Rate money default 85.95,
    Available bit default 1
)
```

```

);
GO

-- =====
-- Database: CeilInn4
-- Table:      DatabaseOperations
-- Description: This table is used to hold information
--              about operations performed on any table
--              of the database. It specifies:
--              a. The type of object on which the action
--                  was performed. The types of object can
--                  be a table
--              b. The name of the table
--              c. The name of the employee who
--                  performed the action.
--              d. The action that was performed. This
--                  can be an insert, an update, or a
--                  delete operation
--              e. The date/time the action was performed
-- =====
CREATE TABLE DatabaseOperations (
    ObjectType nchar(20),
    ObjectName nvarchar(40),
    EmployeeName nvarchar(50),
    ActionPerformed nvarchar(50),
    TimePerformed datetime2
);
GO

```

4- اضغط F5 لتنفيذ الاستعلام.

إنشاء قاذح (Trigger):

يمكنك إنشاء القوالب باستخدام SQL، سنتحدث عن ذلك لاحقاً. لتتعرف على الهيكل الأساسي لاستعلام إنشاء قاذح، افتح نافذة الاستعلام، ثم من Template Explorer قم بتوسيع البند Triggers، ثم اسحب البند Create T-SQL Trigger وأفلته على إطار الاستعلام لتحصل على ما يلي:

```

-- =====
-- Template generated from Template Explorer using:
-- Create Trigger (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- See additional Create Trigger templates for more
-- examples of different Trigger statements.
--
-- This block of comments will not be included in
-- the definition of the function.
-- =====
SET ANSI_NULLS ON
GO

```



```

SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
CREATE TRIGGER <Schema_Name,
              sysname,
              Schema_Name>.<Trigger_Name,
              sysname,
              Trigger_Name>
ON <Schema_Name, sysname, Schema_Name>.<Table_Name, sysname,
Table_Name>
AFTER <Data_Modification_Statements, , INSERT,DELETE,UPDATE>
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here

END
GO

```

تنفيذ القوادح:

خلافًا للإجراءات المخزنة؛ لست بحاجة لتنفيذ القادح، لأن نظام التشغيل بنفسه (من خلال الحدث) ومشغل قاعدة البيانات يتوليان القيام بذلك. يتم تنفيذ القادح وراء الكواليس فور إرسال الجدول الحدث المناسب لإطلاق القادح. في الواقع، يتوقف إطلاق الحدث على وقوع تغيير (إضافة، حذف، تعديل) في الكائن من عدم وقوعه.

تسيير القوادح:

يعتبر القادح غرض (كائن) من قاعدة بيانات، لذلك فهو يحمل اسماً ويمكن التعديل عليه، كما يمكن حذفه.

التعديل على القادح:

إذا كان سلوك القادح غير مناسب، يمكنك تغييره، وصيغة التعديل على القوادح كما يلي:

```

ALTER TRIGGER schema_name.trigger_name
ON schema_name.table_name
AFTER , UPDATE>
AS
    statement

```

للحصول على هيكل استعلام لإنشاء قادح، قم بفتح نافذة الاستعلام، ثم من Templates Explorer، قم بتوسيع البند Triggers، واسحب الخيار Alter ثم اسقطه في إطار الاستعلام:

```
=====
-- Alter T-SQL Trigger Template
=====
USE <database_name, sysname, AdventureWorks>
GO

ALTER TRIGGER <schema_name, sysname, Sales>.<trigger_name, sysname,
uStore>
ON <schema_name, sysname, Sales>.<table_name, sysname, Store>
AFTER <data_modification_statements, , UPDATE>
AS <T-SQL_statement,
'
UPDATE Sales.Store
SET ModifiedDate = GETDATE()
FROM inserted WHERE inserted.CustomerID = Sales.Store.CustomerID>
GO
```

حذف القوادح:

يمكنك حذف قادح من الجدول باستخدام الصيغة التالية:

```
DROP TRIGGER TriggerName
```

حيث يمثل *TriggerName* اسم القادح الذي تود حذفه.

قوادح DML:

يدعم SQL Server ثلاثة أنواع من القوادح: DML، DDL، و Logon.

قادح DML هو إجراء يقوم بأداء أحد العمليات على بيانات الجدول. وهذا يعني أن هذا النوع من القوادح يجب أن يتم إنشاؤه استناداً إلى جدول من قاعدة البيانات.

قوادح DML للإضافة¹

قادح الإضافة هو قادح DML يتم تشغيله عند إضافة سجل إلى الجدول. الصيغة العامة

لإنشاء قادح DML كما يلي:

```
CREATE TRIGGER TriggerName
ON TableName
AFTER/FOR INSERT
AS
    TriggerCode
```

¹ DML هي اختصار Data Manipulation Language.

بعد عبارة **CREATE TRIGGER** يأتي اسم القادح بإتباع قواعد تسمية الكائنات في SQL Server، ثم كلمة ON يليها اسم الجدول الذي سيطبق عليه القادح، حيث يجب التأكد من وجود الجدول في قاعدة البيانات.

نتحدث في هذه الحالة عن إنشاء قادح بعد إضافة سجل، لأجل ذلك؛ يمكنك استخدام أحد العبارتين AFTER INSERT أو FOR INSERT.

لكتابه شفرة SQL التي تشكل موضوع عمل القادح، تضاف كلمة AS متبوعة بالتعليمات البرمجية المناسبة.

بعد إنشاء قادح الإضافة، وحالما يحين الوقت المناسب لإطلاق القادح (أي عند وقوع الحدث) سيتم تنفيذ القادح. وحينها يقوم محرك قاعدة البيانات تلقائياً وداخليا بإنشاء جدول مؤقت اسمه **inserted**. يحتوي هذا الجدول على نسخة من السجل (أو السجلات) التي تم إنشاؤها. يمكنك الوصول إلى تلك السجلات إذا لزم الأمر.

تطبيق عملي: إنشاء قادح DML

1- في نافذة استعلام جديدة اكتب الاستعلام الآتي لإنشاء القادح:

```
USE CeilInn4;
GO

-- =====
-- Database:      CeilInn4
-- DML Trigger: RecordInsertion
-- Description: This trigger updates the DatabaseOperations
--               by letting it know that a new record was
--               added to the Rooms table. The trigger
--               also specifies the name of the employee
--               who performed the operation and the time
--               this occurred
-- =====
CREATE TRIGGER RecordInsertion
ON Rooms
AFTER INSERT
AS
BEGIN
    INSERT INTO DatabaseOperations
    VALUES(N'Table', N'Rooms', SUSER_SNAME(),
           N'Created a new record', GETDATE())
END
GO
```

2- اضغط F5 للتنفيذ.

5- في مستكشف الكائنات، انقر بالزر الأيمن على Databases ثم اختر Refresh

6- قم بتوسيع البند Databases ثم CeilInn4

7- من قاعدة البيانات CeilInn4 ثم افتح Tables

8- من أجل إطلاق حدث "إدخال بيانات" سنقوم بإضافة بعض السجلات، بالزر الأيمن

انقر على Rooms ثم اختر Edit Top 200 Rows

RoomNumber	LocationCode	RoomType	BedType	Rate	Available
104	SLSP				
105	SLSP		King	95.50	True
106	SLSP		King	95.50	True
107	SLSP				True
108	SLSP		King	95.50	
109	SLSP				True
110	SLSP	Conference		450.00	True

9- أغلق الجدول

10- من مستكشف الكائنات، انقر بالزر الأيمن على الجدول DatabaseOperations وانقر

على Select Top 1000 rows لمشاهدة السجلات.

11- أغلق الجدول DatabaseOperations.

قواعد DML للحذف:

يمكنك إنشاء قوادح يتم تطبيقه عند حذف سجل من الجدول، باستخدام الصيغة التالية:

```
CREATE TRIGGER TriggerName
ON TableName
AFTER/FOR DELETE
AS
    TriggerCode
```

في هذه الحالة، تستخدم الصيغة AFTER DELETE أو FOR DELETE. بقية المعاملات

تطرقنا إليها في الحالة السابقة.

عند حذف أحد السجلات من الجدول، يقوم محرك قاعدة البيانات بإنشاء جدول مؤقت

يسمى **deleted**. يحتوي هذا الجدول على نسخ من السجلات التي تم حذفها. يمكنك، إذا لزم

الأمر، الوصول إلى هذا الجدول لمعرفة المزيد عن تلك السجلات.

تطبيق عملي: قادح DML للحذف

1- في نافذة استعلام جديدة، اكتب الآتي لإنشاء قادح حذف:

```
USE CeilInn4;
GO

-- =====
-- Database:      CeilInn4
-- DML Trigger: RecordDeletion
-- Description: This trigger adds a new record to the
--              DatabaseOperations when an existing record
--              of the Rooms table has been deleted.
-- =====
CREATE TRIGGER RecordDeletion
ON Rooms
AFTER DELETE
AS
BEGIN
    INSERT INTO DatabaseOperations
    VALUES(N'Table', N'Rooms', SUSER_SNAME(),
           N'Deleted a room', GETDATE())
END
GO
```

2- اضغط F5 لتنفيذ الاستعلام

3- لحذف سجل من الجدول Rooms، انقر عليه من متصفح الكائنات، ثم اختر Edit
Top 200 Rows

4- بالزر الأيمن انقر على طرف السجل 106 واختر Delete ثم Yes للتأكيد.

5- أغلق الجدول

6- من متصفح الكائنات، انقر بالزر الأيمن على الجدول DatabaseOperations واختر
Select Top 1000 rows لمعاينة سجلاته

7- إغلاق DatabaseOperations الجدول.

خصائص قوادح DML:

يمكنك الذهاب أبعد من ذلك في إنشاء قوادح، يمكنك إنشاء العديد من القوادح (طبعا، بأسماء مختلفة) التي تقوم بنفس الإجراءات على الجدول. فمثلا يمكنك إنشاء عدة قوادح إضافة (أو حذف أو تحديث) التي تعمل على نفس الجدول، و تستهدف نفس الإجراءات.

تشمل القواعد العديد من الخصائص الإضافية.

القواعد وقيود إدخال البيانات:

رأينا في درس سابق أن لمساعدة المستخدم لإدخال البيانات، يمكنك منع إدخال بيانات إلى أحد الحقول أو إجبار المستخدم على إدخال قيم إليها. وذلك بتفعيل خاصية **NOT NULL** للحقل المحدد. إذا لم يتم المستخدم أثناء إدخال البيانات بتوفير قيمة لهذا الحقل، فإن السجل لا يمكن إنشاؤه. عند إنشاء قادم **DML** لإضافة بيانات، لذلك يجب مراعاة إمكانية عدم إدخال أي قيمة، وإلا فلن يتم تشغيل القادم.

رأينا أيضا ضمن هذه السلسلة إمكانية إنشاء قيد لإدخال البيانات (**Check Constraint**) على الجدول، للتأكد من مطابقة السجل المضاف (أو المعدل) لشروط معينة، وإذا لم تتحقق الشروط المطبقة على الحقل، فلن يتم إنشاء (أو تعديل) السجل. إذا كان القادم يقوم بإضافة أو تعديل بيانات دون مراعاة قيود الإدخال، فسيفشل في ذلك.

من حدود قيود إدخال البيانات هو أنها لا تطبق إلا على الجدول الذي أنشأت عليه، بينما يمكن لقادم **DML** أن يقوم بأداء قيد لإدخال البيانات على أكثر من جدول واحد، وهذا ما يوفر ميزة زائدة على قيود الإدخال الاعتيادية.

رأينا عند دراسة البيانات وعلاقات التكامل المرجعي (**Referential Integrity**)، أنه عند تحرير سجل من الجدول الأصل، فسيحصل التغيير أيضا على الجدول الابن، وهذا يدل على أن تطبيق التكامل المرجعي يحصل على أكثر من جدول. يقوم قادم **DML** عند إطلاقه، بفحص قواعد التكامل المرجعي، ويتوقف حال مخالفته أحد هذه القواعد.

القواعد البديلة (**Instead of Trigger**):

ليكن الجدول التالي في قاعدة بيانات:

```
CREATE DATABASE SmallBusiness;  
GO
```

```
USE SmallBusiness;  
GO
```

```
CREATE TABLE Customers
```

```
(
    CustomerID int identity(1, 1) primary key not null,
    AccountNumber nchar(10),
    FullName nvarchar(50)
);
GO
```

```
CREATE TABLE DatabaseOperations (
    EmployeeName nvarchar(50),
    ActionPerformed nvarchar(50),
    TimePerformed datetime2
);
GO
```

من خلال ما رأيناه سابقاً، عندما يقوم المستخدم بإدخال بيانات إلى جدول (أو كائن View)، يقوم هذا الأخير بإطلاق حدث فور إنشاء السجل. رأينا أن قوادم DML تسمح بإنشاء إخطار لذلك الحدث، فمثلاً يمكنك استغلال ذلك لملء جدول (Log) لتتقب التغييرات.

افتراضياً، عند القيام بإضافة أو حذف أو تعديل سجل، فإن التغيير يحصل فوراً ويطبق على الجدول، وبدلاً من قبول التغييرات يمكنك إلغاؤها. يمكنك أيضاً استخدام قادم DML من إعداد مذكرة لحفظ التغييرات، ويتم ذلك بإنشاء نوع آخر من قوادم DML وهو القوادم البديلة (Instead of Trigger).

عند تنفيذ قادم AFTER/FOR على أي جدول بعد حدث طراً عليه، قد ترغب بفعل شيء ما قبل هذا الحدث. مثلاً، قد تحتاج إلى منع المستخدم من إضافة (أو حذف أو تغيير) بيانات في الجدول، وطبعاً، ينبغي الاهتمام بـ طبيعة عمل القادم، طريقة وحيدة تسمح لك القيام بذلك هي استخدام القوادم البديلة.

إنشاء القوادم البديلة:

عند تطبيق قادم AFTER، يمكن تطبيق قادم بديل على جدول أو كائن View. الصيغة:

```
CREATE TRIGGER TriggerName
ON TableOrViewName
INSTEAD OF INSERT/UPDATE/DELETE
AS
    TriggerCode
```

يبدأ الاستعلام بعبارة CREATE TRIGGER متبوعة باسم القادم، بعد ذلك اكتب ON يليها اسم جدول أو الكائن View التي سيؤثر عليها القادم.

الجديد هنا عبارة INSTEAD OF، يليها نوع العمليات الحاصلة. إذا كنت تريد:

- التقاط حدث إنشاء سجل، استخدم المعامل **INSERT**
- التقاط حدث تعديل سجل، استخدم المعامل **UPDATE**
- التقاط حدث حذف سجل، استخدم المعامل **DELETE**

لتحرير استعلام SQL مضمون عمل القادح، أضف كلمة AS ثم اكتب التعليمات البرمجية.

عند استخدام التعبير **INSTEAD OF**، يتم تطبيق القادح عندما يفتح الجدول (أو كائن View) ولكن قبل حصول تغيرات الحدث. والفرق بينه وبين قادح **AFTER**، أنه يمكنك تنفيذ بعض الإجراءات قبل أن يحصل التغيير في الجدول أو الكائن View، هذا يعني أيضا أنه إذا كانت وظيفة القادح إنشاء سجل جديد، فالسجل في هذه الحالة غير موجود، وهو ما يعني أنك لا تستطيع التقاط هذا السجل. وهنا يمكنك أيضا منع إنشاء السجل الذي سيتم إنشاؤه.

مثال، القادح الآتي يقوم بإضافة سجل جديد إلى الجدول عوض عملية الإضافة إليه:

```
USE SmallBusiness;
GO

CREATE TRIGGER CreateCustomer
ON Customers
INSTEAD OF INSERT
AS
    BEGIN
        INSERT INTO DatabaseOperations
        VALUES(SUSER_SNAME(),
            N'Attempt to create new record', GETDATE())
    END
GO
```

إذا كنت ترغب في الحصول على نسخة من السجل الذي تمت إضافته إلى الجدول، يمكنك الولوج إليه من خلال السجل (الجدول) النظامي **inserted** (لقوادح الإضافة **INSERT** أو التحديث **UPDATE**) أو السجل **deleted** (لقوادح الحذف). مثال:

```
USE SmallBusiness;
GO

DROP TRIGGER CreateCustomer;
GO

CREATE TRIGGER CreateCustomer
ON Customers
INSTEAD OF INSERT
AS
BEGIN
```



```

INSERT INTO Customers
SELECT AccountNumber, FullName FROM inserted
END
GO

```

تطبيق عملي: إنشاء قوادم بديلة

1- في نافذة استعلام، اكتب ما يلي لإنشاء قادم إضافة وآخر للتحديث:

```

USE CeilInn4;
GO

-- =====
-- Database:    CeilInn4
-- View:        Logistics
-- Description: This view retrieves the list of rooms
--              of this hotel
-- =====
CREATE VIEW Logistics
AS
    SELECT RoomNumber, LocationCode, RoomType,
           BedType, Rate, Available
    FROM Rooms;
GO

-- =====
-- Database:    CeilInn4
-- DML Trigger: AttemptedRecordInsertion
-- Description: This trigger acts on a table to update the
--              DatabaseOperations to let it know that an
--              attempt was made to create a new room
-- =====
CREATE TRIGGER AttemptedRecordInsertion
ON Rooms
INSTEAD OF INSERT
AS
BEGIN
    INSERT INTO DatabaseOperations
    VALUES(N'Table', N'Rooms', SUSER_SNAME(),
           N'Attempted to create a new record', GETDATE())
END
GO

-- =====
-- Database:    CeilInn4
-- DML Trigger: AttemptedRecordUpdate
-- Description: This trigger acts on a view to update the
--              DatabaseOperations to let it know that an
--              attempt was made to edit a record of
--              the Rooms table
-- =====
CREATE TRIGGER AttemptedRecordUpdate
ON Logistics
INSTEAD OF UPDATE
AS
BEGIN
    INSERT INTO DatabaseOperations
    VALUES(N'View', N'Logistics', SUSER_SNAME(),
           N'Attempted to change a room's information',
           GETDATE())

```

END
GO

2- اضغط F5 لتنفيذه

3- من متصفح الكائنات وتحت CeilInn4، انقر بالزر الأيمن على Rooms، وانقر Edit
Top 200 Rows

4- قم بإضافة السجلات التالية:

RoomNumber	LocationCode	RoomType	BedType	Rate	Available
104	LRL		King	95.50	
112	SLSP		King	95.50	True

5- لاحظ إشعار التنبيه على رؤوس السجلات التي تم تغييرها في الجدول. أغلق الجدول

6- من متصفح الكائنات، تحت Views، انقر بالزر الأيمن على الجدول Logistics واختر
Edit Top 200 Rows

7- قم بتعديل السجلات الآتية:

RoomNumber	LocationCode	RoomType	BedType	Rate	Available
108	LRL	Conference	Delete	425.75	

8- لاحظ إشعار التنبيه على رؤوس السجلات لكائن View. اضغط مفتاح Esc

9- أغلق الجدول

10- من متصفح الكائنات، انقر بالزر الأيمن على الجدول DatabaseOperations ثم اختر
Select Top 1000 rows لمعاينة سجلاته

11- أغلق الجدول.

خصائص القوادح البديلة:

- هناك عدة فروق بين قوادح AFTER/FOR و INSTEAD OF. مثلا:
- قوادح INSTEAD OF UPDATE و INSTEAD OF DELETE لا يمكنها التأثير على الجداول التي تحوي حقولا معلمة بخاصية DELETE أو UNIQUE.
- لا يمكنك إنشاء أكثر من قوادح بديل (INSTEAD OF) لكل جدول.

قواعد DDL:

رأينا في الدروس الأولى، أن إنشاء قاعدة بيانات يستخدم تعليمات Data Definition Language (DDL). ورأينا فيما بعد أمثلة أخرى على ما تنطوي عليه أوامر DDL بما فيها إنشاء الجداول، تقوم كل من هذه التعليمات بإطلاق حدث DDL.

قواعد DDL هي القواعد التي يتم تنفيذها عندما تقوم أوامر DDL معينة بإطلاق حدث. ويشمل ذلك إنشاء أو تعديل أو حذف كائن وليس سجلاته، وهذا هو الفرق الأساسي بين هذا النوع من القواعد وبين قواعد DML التي يتم تنفيذها عند وقوع حدث متعلق بالسجلات.

تمنحك قواعد DDL فرصة القيام ببعض الأعمال الإدارية استجابة لبعض الأحداث، على سبيل المثال، يمكنك الحصول على إشعار أو إخطار من أي شخص تلقائياً عبر البريد الإلكتروني، بأن (وأي) كائناً تم إنشاؤه. أو يمكنك استخدام قواعده DDL لإلغاء هذه العملية.

إنشاء قواعد DDL:

يمكنك إنشاء قواعد DDL باستخدام الصيغة الأساسية التالية:

```
CREATE TRIGGER TriggerName
ON DATABASE/ALL SERVER
FOR/AFTER WhatEvent
AS
TriggerCode
```

البداية بعبارة CREATE TRIGGER يليها اسم القواعد وفق قواعد العامة لتسمية الأغراض، ثم كلمة ON متبوعة:

- بكلمة DATABASE لتطبيق القواعد على قاعدة البيانات الحالية، وسيتم تنفيذ أوامر القواعد على قاعدة البيانات المحددة.
 - بعبارة ALL SERVER إذا كنت تود تطبيق القواعد على الخادم. في هذه الحالة، سيتم تنفيذ القواعد عندما يقع الحدث المقصود في أي جزء من أجزاء الخادم.
- بعد تحديد الكائن (الخادم بأكمله أو قاعدة البيانات الحالية) الذي سيطبق عليه القواعد، أضف كلمة FOR أو AFTER متبوعة بالحدث الذي سينفذ القواعد لأجله. وكما ذكرنا سابقاً،

فإن الأحداث هنا هي أوامر DDL. لتحديد أحد هذه الأحداث، استخدام صيغة الحدث مع الفصل بين الكلمات بالرمز "_". مثلاً، إذا كنت تريد أن يتم تنفيذ القادح عند إرسال استعلام إنشاء الجدول (CREATE TABLE)، حدد الحدث CREATE_TABLE.

بعد ذلك، اكتب كلمة AS متبوعة بالتعليقات البرمجية التي سينفذها القادح.

في هذا المثال يقوم القادح عند إنشاء جدول جديد، بإضافة ملاحظة في الجدول:

```
USE SmallBusiness;
GO

CREATE TRIGGER LogNewTableCreation
ON DATABASE
FOR CREATE_TABLE
AS
BEGIN
    INSERT INTO DatabaseOperations
    VALUES(SUSER_SNAME(),
           N'A new table was created', GETDATE())
END
GO
```

كلما تم إنشاء جدول جديد في قاعدة البيانات الحالية، يتم تنفيذ القادح، لإضافة اسم المستخدم الذي أنشأ الجدول وتاريخ ووقت إنشائه، وعبارة نصية قصيرة، ثم تخزين هذه البيانات في جدول DatabaseOperations.

كما أوضحنا في قوادح DML، يمكنك إدارة (تعديل أو حذف) قوادح DDL بنفس الطريقة. ويتم ذلك باستخدام نفس المواصفات التي رأيناها مع قوادح DML.

مسائل أخرى في SQL Server

هذه بعض النقاط المهمة التي لم تتعرض لها السلسلة السابقة.

ربط وفصل (Attach/Detach) ملف قاعدة البيانات SQL Server:

تتجسد قاعدة البيانات SQL Server في نوعين من الملفات أساسا:

- **Primary Data File**: هي ملفات بلاحقة MDF، وهي الملفات التي تخزن البيانات.
- **Transaction Log File**: هو ملف بلاحقة LDF، ويخزن فيه كل الاستعلامات والتعليمات المنفذة على قاعدة البيانات.
- **Secondary Data Files**: عند إنشاء قاعدة البيانات يمكن توزيعها على عدة أقسام ملفات (Filegroups)، وتخزن في القرص بلاحقة NDF.

للمزيد حول ملفات أنماط الملفات وطريقة تخزين البيانات انظر الرابط:

[http://msdn.microsoft.com/en-us/library/aa174545\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa174545(SQL.80).aspx)

ربط (Attach) ملف قاعدة البيانات:

عند إنشاء قاعدة البيانات باستخدام Microsoft SQL Server Management Studio يقوم هذا الأخير (تلقائيا) بربط ملف قاعدة البيانات (MDF File) إلى الخادم، وبالتالي يمكنه التعرف إلى قاعدة البيانات وتنفيذ الاستعلامات عنها إضافة إلى عمليات الحذف، التعديل، وغير ذلك...

إذا قمت بنسخ ملف قاعدة البيانات (MDF File) إلى جهاز آخر، فإن الخادم في هذا الجهاز لن يتعرف على قاعدة البيانات، وسيُرسل رسالة خطأ إذا تلقى أي استعلام عنها.

ليتعرف الخادم على ملف قاعدة البيانات يجب ربطها (Attach) بإرسال استعلام مثل

الآتي:

```
CREATE DATABASE [MyDataBase] ON  
( FILENAME = N'C:\path\MyDataBase.mdf' ),  
( FILENAME = N'C:\path\MyDataBase_log.ldf' )
```

FOR ATTACH

في البداية عبارة **CREATE DATABASE** يتبعها الاسم المستعار الذي سيستعمل لتسجيل (ربط) قاعدة البيانات إلى الخادم، ويمكن لهذا الاسم أن يختلف عن اسم الملف.

بعد **FILENAME** يأتي المسار الكامل لاسم ملف قاعدة البيانات (MDF File)

ثم **FILENAME** الثانية (اختيارية) لربط ملف أي ملف آخر لقاعدة البيانات (في المثال ملف LDF وهو ملف اختياري).

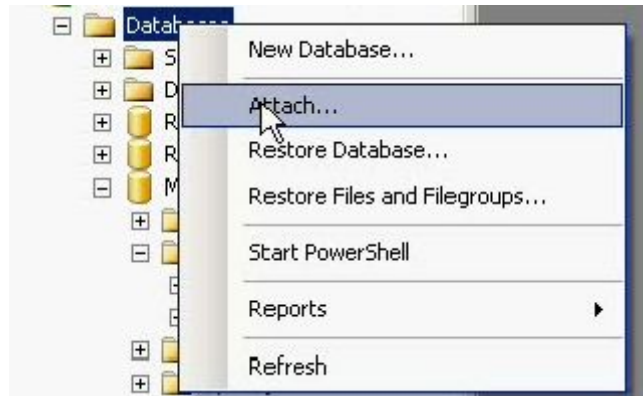
FOR ATTACH تعني إنشاء قاعدة البيانات على الخادم من خلال ربط ملف قاعدة بيانات موجود على القرص الصلب.

بطريقة أخرى يمكن استخدام الإجراء المخزن **sp_attach_db** كما يلي:

```
sp_attach_db 'MyDataBase',  
            'C:\path\MyDataBase.mdf',  
            'C:\path\MyDataBase.ldf'
```

للقيام بذلك باستخدام MS SQL Server Management Studio حدد البند Databases ثم

انقر Attach، قم بإضافة ملفات قواعد البيانات التي تريد ربطها بالزر Add...، ثم OK.



ملاحظة:

- قبل إرسال مثل هذه الاستعلام يجب التأكد أن قاعدة البيانات الحالية تختلف عن التي سنقوم بربطها أو فصلها، كذلك الأمر فيما يخص عمليات الحذف والتعديل.

لنتأكد أن قاعدة البيانات الحالية غير التي سنجري عمليات عليها، نقوم بتغيير قاعدة البيانات الحالية إلى أي قاعدة بيانات أخرى، مثل master قاعدة بيانات النظام في SQL Server.

لأجل ذلك نضيف قبل الاستعلامات السابقة عبارة:

```
USE [master]
GO
```

وبذلك نكون متأكدين أن قاعدة البيانات الحالية بعد هذين السطرين هي master.

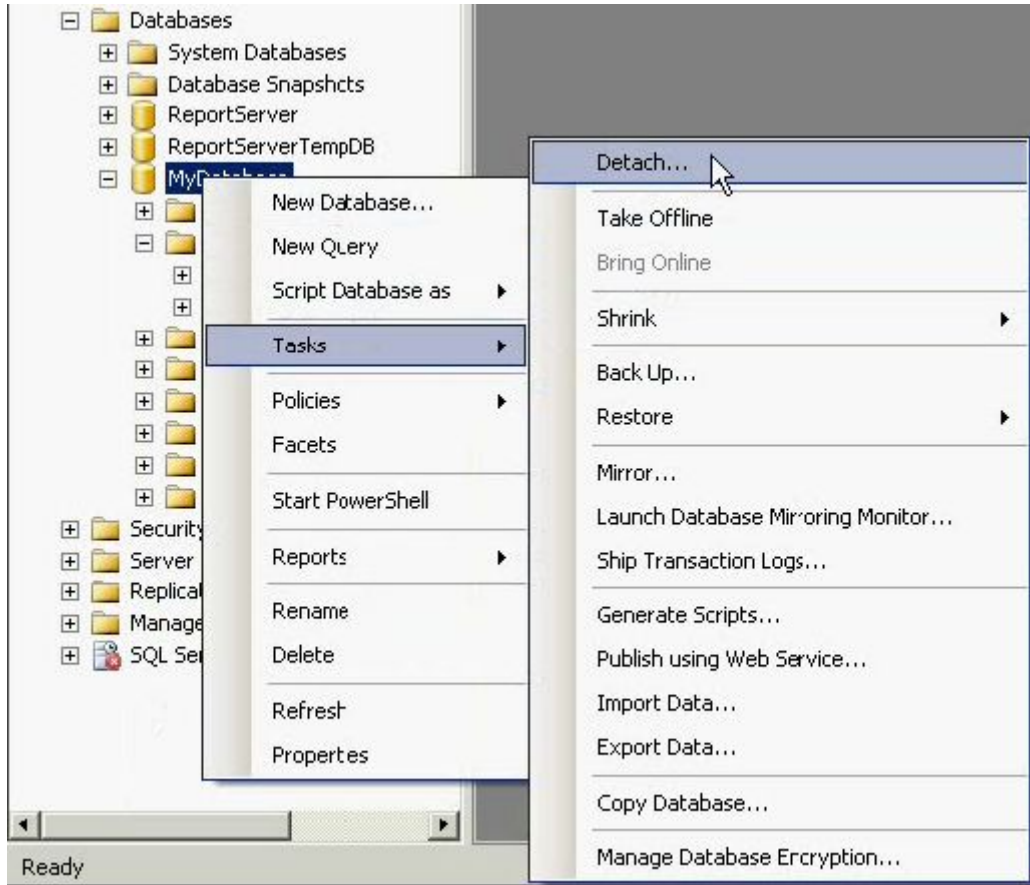
فصل (Dettach) ملف قاعدة البيانات:

إذا أردت نقل ملف قاعدة بيانات من جهاز لآخر، عليك أولاً فصل ملف قاعدة البيانات عن الخادم، ويتم ذلك باستخدام الإجراء المخزن `sp_detach_db` كما يلي:

```
USE [master]
GO
EXEC sp_detach_db @dbname = N'MyDataBase'
GO
```

باستخدام MS SQL Server Management Studio حدد قاعدة البيانات من متصفح

الكائنات ثم Detach -> Tasks، ثم انقر OK للموافقة.



ملاحظة:

- بعض العمليات في SQL Server يمكن تنفيذها بعدة طرق باستخدام SQL، حيث تضيف Microsoft في الإصدارات الحديثة أوامر جديدة، وتبقي على الإجراءات التي كانت تستخدم في الإصدارات السابقة (لأغراض التوافقية Backward Compatibility)، تمهيدا لحذف هذه الإجراءات لاحقا.

مثال:

لتغيير اسم قاعدة البيانات في SQL Server حسب الإصدارات السابقة يستخدم الإجراء المخزن **sp_renamedb** كما يلي:

```
USE [master]
GO
EXEC sp_renamedb 'OldDBName', 'NewDBName'
GO
```

بقي ذلك ساري المفعول لأغراض التوافق مع الإصدارات الجديدة. والآن في الإصدارات الحديثة يستحسن استعمال ما يلي:


```

USE [master]
GO
/* تغيير اسم قاعدة البيانات */
ALTER DATABASE MyDBOldName MODIFY NAME = MyDBNewName
GO

```

للمزيد حول ذلك طالع ملفات المساعدة المرفقة مع SQL Server.

حفظ قاعدة البيانات Backing up SQL Server:

يوفر SQL Server عدة خيارات لحفظ واسترجاع قاعدة البيانات. يمكن حفظ نسخة من قاعدة البيانات بثلاث طرق:

- **Full Backup**: حفظ نسخة من قاعدة البيانات بكاملها دون ملف الأحداث (Log file)
- **Differential Backup**: لحفظ نسخة من التغييرات الحاصلة منذ عملية الحفظ الأخيرة
- **Transaction Log**: لحفظ نسخة من ملف الأحداث (Log file)

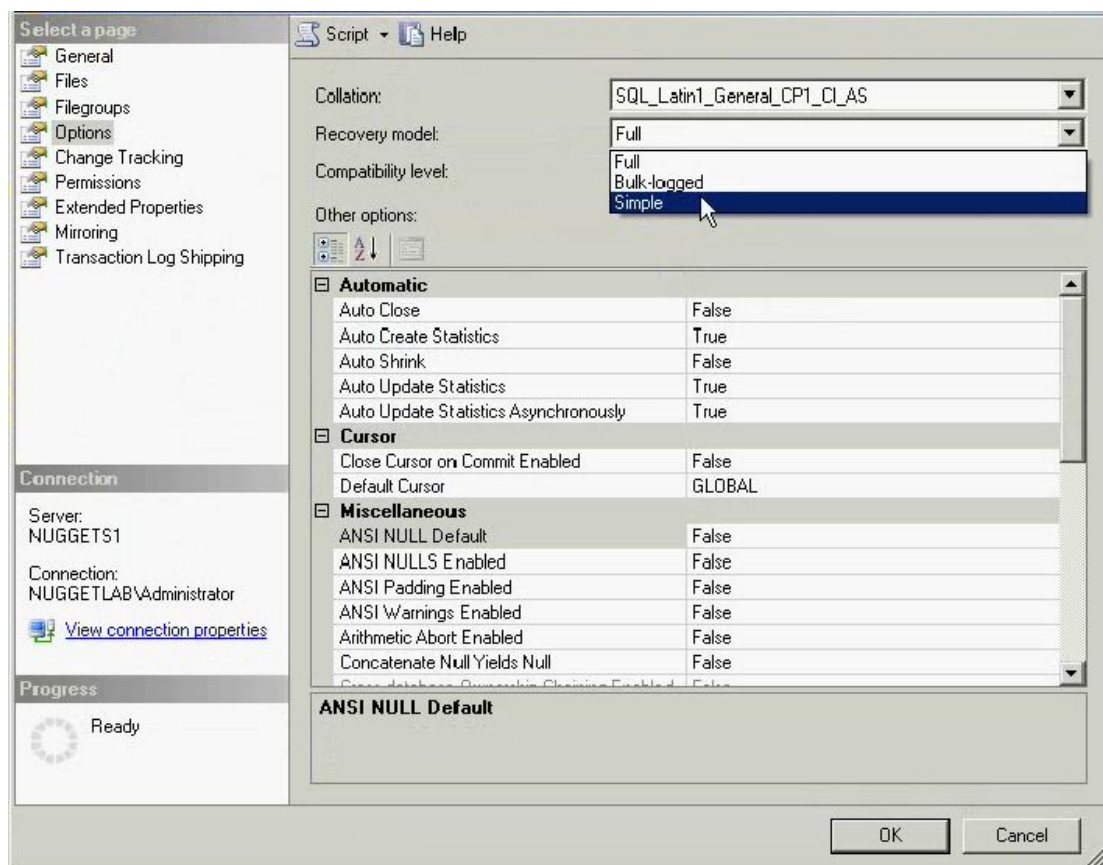
يوفر SQL Server ثلاث أوضاع مختلفة لحفظ نسخة قاعدة البيانات:

- **Full**: سيستخدم هذا الوضع ملف الأحداث (Transaction Log) لاستعادة قاعدة البيانات باسترجاع الاستعلامات المنفذة، وبذلك يمكنك العودة إلى أي وضع سابق.
- **Bulk-logged**: سيتم في هذا الوضع استخدام ملف الأحداث لاسترجاع التغييرات الحاصلة على قاعدة البيانات من خلال استرجاع أهم الاستعلامات المنفذة.
- **Simple**: في هذا الوضع لا حاجة لملف الأحداث، ويتم استرجاع نسخة من قاعدة البيانات بكاملها، وأي تغيير يحصل بعد آخر عملية حفظ لن يتم استرجاعه.

للمزيد من التوضيحات وتفاصيل أخرى حول حفظ نسخة احتياطية، راجع الرابط:

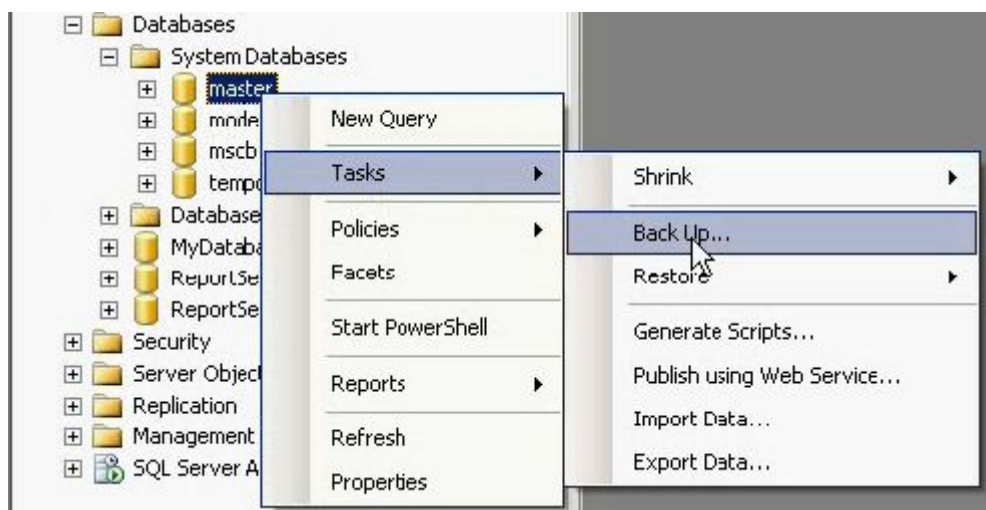
<http://msdn.microsoft.com/en-us/library/ms187510.aspx>

نأتي إلى SQL Server Management Studio لاختيار وضع الاسترجاع، انقر بالزر الأيمن على قاعدة البيانات واختر Properties، حدد التبويب Options لتجد طريقة الاسترجاع من قائمة الاختيارات Recovery Model (Simple افتراضياً):



لحفظ نسخة من قاعدة البيانات انقر بالزر الأيمن على قاعدة بيانات أخرى¹ (ولكن قاعدة

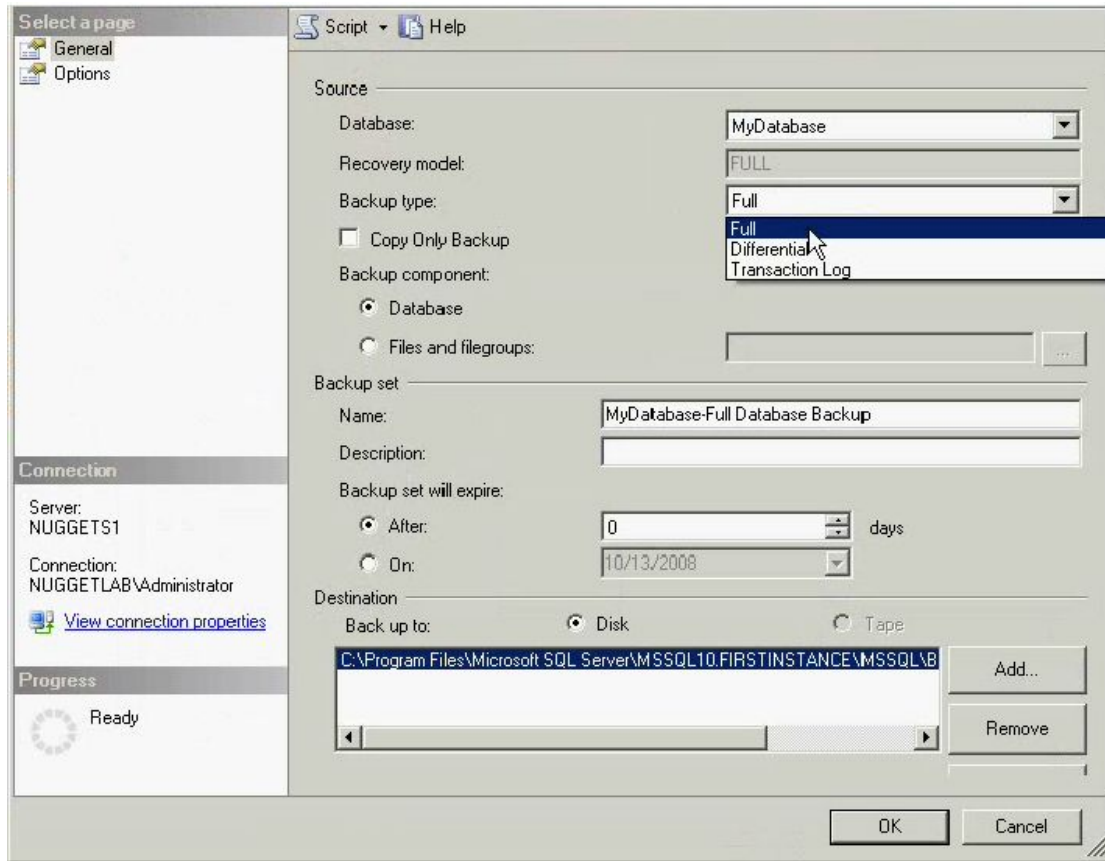
البيانات النظامية master) ثم اختر Tasks -> Back Up...



حدد من قائمة الاختيارات Database قاعدة البيانات التي تود حفظ نسخة منها، ومن

Backup type اختر نوع الحفظ (Full كخيار افتراضي):

¹ لا يمكنك حفظ نسخ من قاعدة البيانات المستعملة حالياً لأنها قيد الاستعمال، لذلك قم بالاتصال بقاعدة بيانات أخرى.



هناك أيضا خيارات إضافية أخرى، منها حفظ أحد الملفات أو قسم من الملفات (Filegroups)، أيضا تحديد تاريخ انقضاء النسخة المحفوظة لمنع استرجاع نسخ قديمة، وغير ذلك من الخيارات المتقدمة في التبويب Options.

انقر Add... لإضافة مجلد للحفظ مع اختيار اسم الملف بأي لاحقة شئت (*.bak افتراضيا)، ثم انقر OK للموافقة.

باستخدام SQL يمكنك الحصول على نفس النتيجة بإرسال استعلام كالآتي:

```
BACKUP DATABASE [MyDatabase]
TO DISK = N' C:\Program Files\Microsoft SQL
Server\MSSQL10.SQLEXPRESS\MSSQL\Backup\BackupMyDatabase.bak', SKIP,
NOWIND, NOUNLOAD, COMPRESSION, STATS = 10
GO
```

استرجاع قاعدة البيانات SQL Server :Restoring

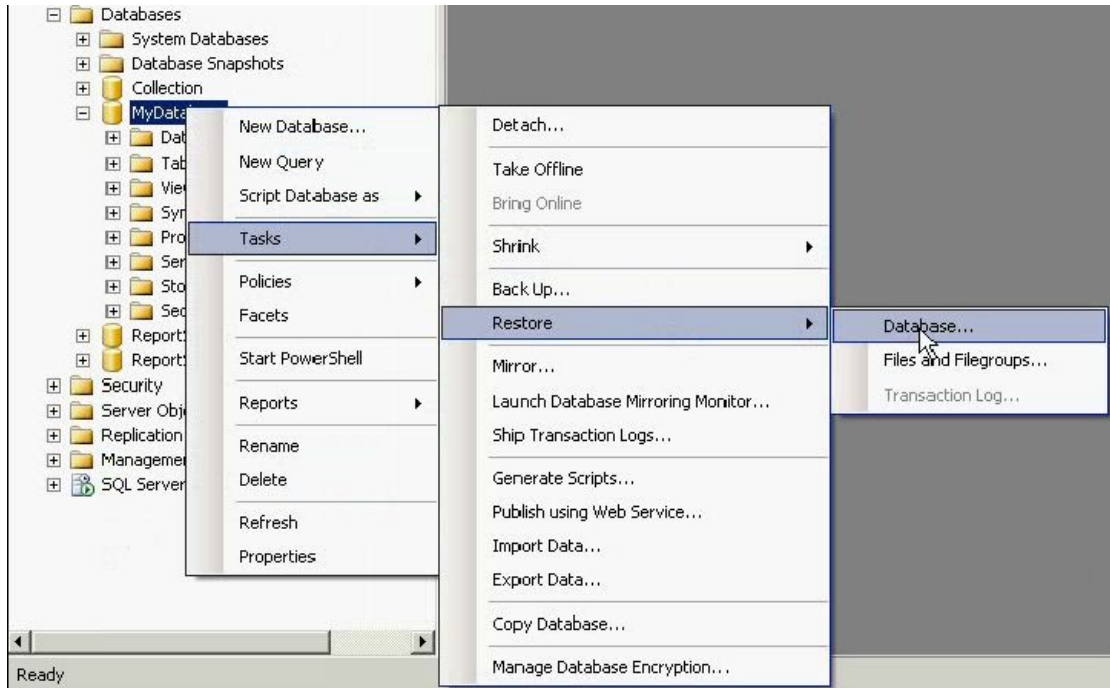
تتم عملية استرجاع البيانات في SQL Server من ثلاثة طرق:

- **Database Restore**: وهو الخيار الافتراضي الذي يتم فيه استرجاع نسخة قاعدة البيانات بأكملها.
 - **File Restore**: بهذه الطريقة يمكنك استرجاع أحد ملفات قاعدة البيانات دون الملفات الأخرى.
 - **Page Restore**: في هذه الطريقة يتوجب اختيار إحدى طريقتين الحفظ Bulk أو Full فقط، ومن خلالها يمكنك استعادة جزء محدد من قاعدة البيانات.
- لاسترجاع نسخة من قاعدة البيانات يجب قطع الاتصال بها، والاتصال بقاعدة بيانات أخرى (مثلًا master) لتنفيذ عملية الاسترجاع¹.
- يمكن عند حفظ نسخة من قاعدة البيانات استخدام كل طرق الحفظ السابقة (Full، Differential، Log) وتخزين الناتج في ملف وحيد، وحينها يمكن أن تتم عملية استرجاع نسخة من قاعدة بيانات SQL Server وفق الترتيب الآتي:
- **Full Backup**: استرجاع نسخة من قاعدة البيانات بأكملها، وهنا يمكن تحديد خيار NORECOVERY لنخبر SQL Server ألا يقوم بالاسترجاع فوراً، لأننا نود استرجاع بيانات أخرى من:
 - **Differential Backup**: الذي يحتوي على بيانات (أو عمليات) تم حفظها بعد حفظ نسخة كاملة من قاعدة البيانات، ويمكن هنا أيضاً استعمال الخيار NORECOVERY لاسترجاع:
 - **Log Backup**: الذي يخزن الأحداث التي جرت على قاعدة البيانات، وبعده تبدأ فعلياً عملية الاسترجاع.

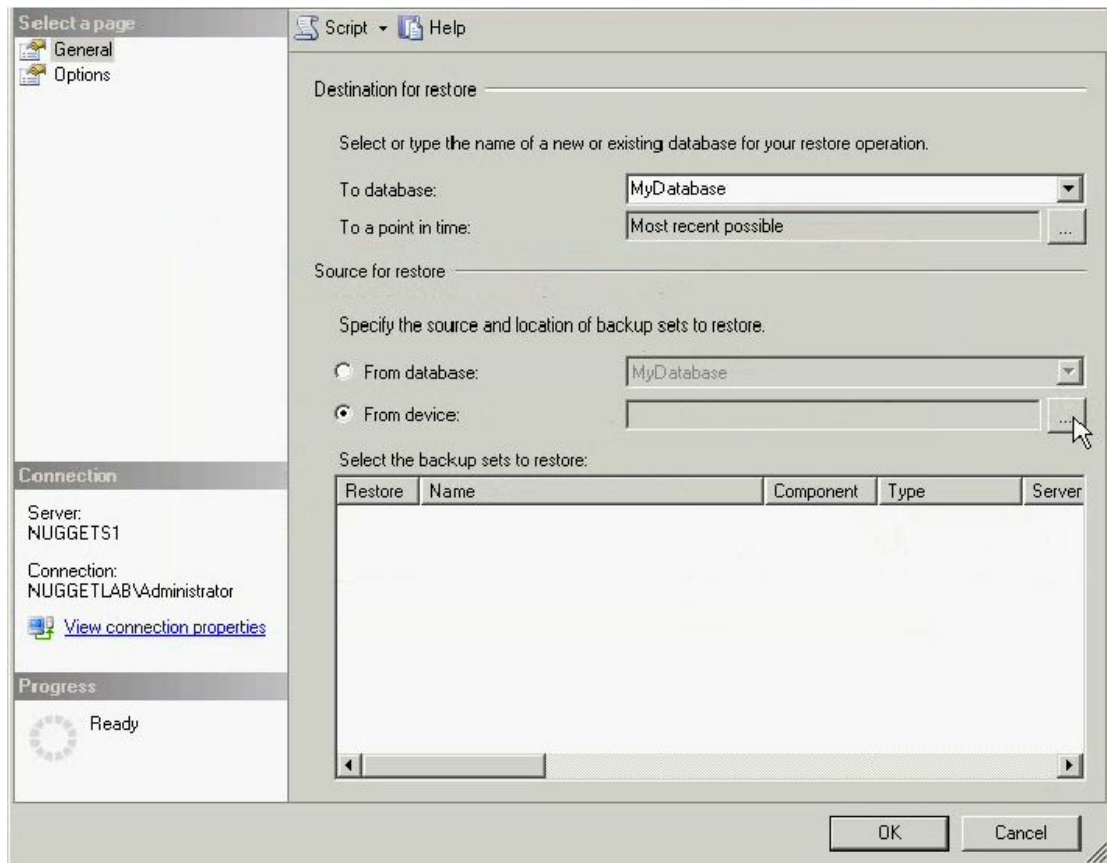
¹ لا حاجة لذلك عند استخدام بعض نسخ SQL Server مثل Enterprise Edition.

لاسترجاع نسخة من قاعدة البيانات باستخدام SQL Server Management Studio قم

بتحديدها من متصفح الكائنات ثم اختر Tasks -> Restore -> Database...

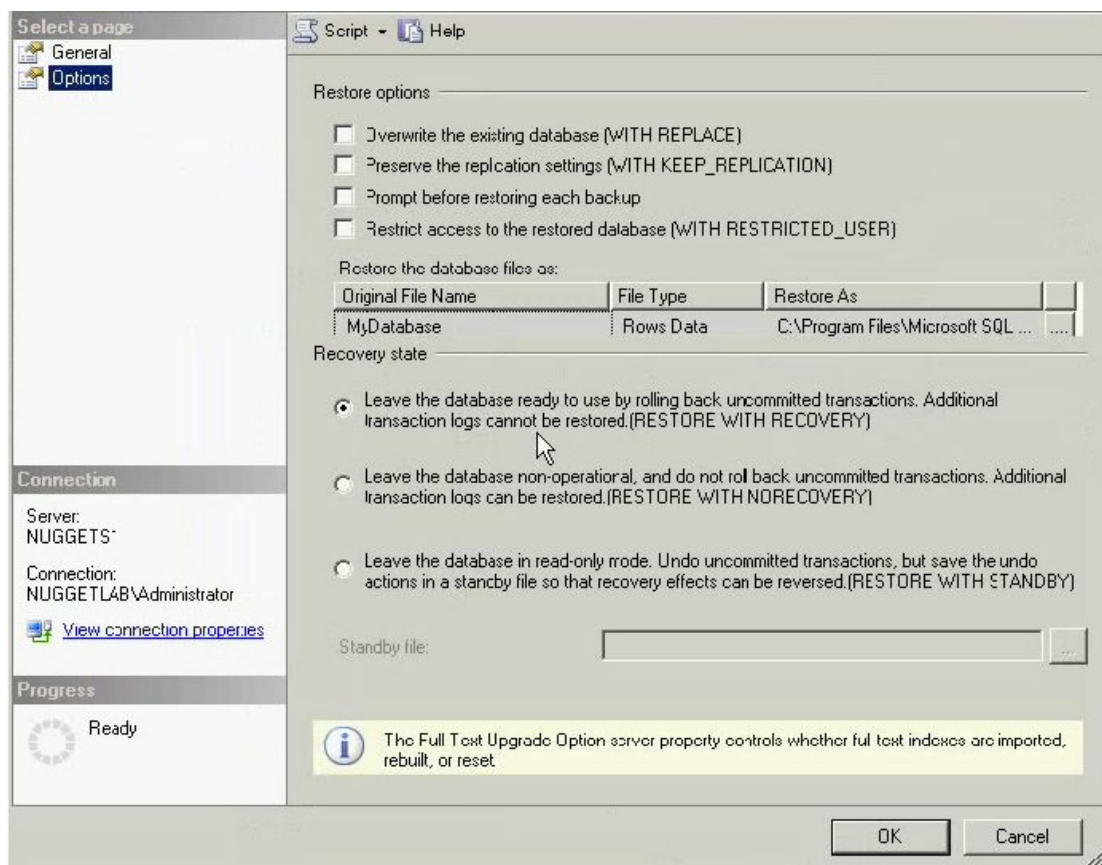


حدد قاعدة البيانات ثم من Form device وزر التفاصيل (...) اختر ملف الاسترجاع:



بعد اختيار ملف الاسترجاع، قم بتحديد (Check) العناصر التي ترغب باسترجاعها، ثم OK للموافقة على ذلك.

هناك خيارات إضافية عديدة يوفرها SQL Server Management Studio، مثل الرجوع إلى بيانات وقت معين، استرجاع البيانات مع تعويض البيانات الموجودة، استرجاع أجزاء من البيانات تدريجياً وغير ذلك...



طالع المزيد حول استرجاع البيانات في SQL Server من خلال الرابط التالي:

<http://msdn.microsoft.com/en-us/library/ms177429.aspx>

باستخدام SQL يمكن حفظ نسخة من البيانات بمثل الاستعلام الآتي:

```
RESTORE DATABASE [MyDatabase]
FROM DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL10.SQLEXPRESS\MSSQL\Backup\BackupMyDatabase.bak' WITH
FILE = 1, NOUNLOAD, STATS = 10
GO
```

ملاحظة:

تأكد عند استرجاع نسخة من قاعدة البيانات أنك (أو النظام) لا تستخدمها حالياً، يمكنك لأجل ذلك الاتصال بقاعدة البيانات أخرى.

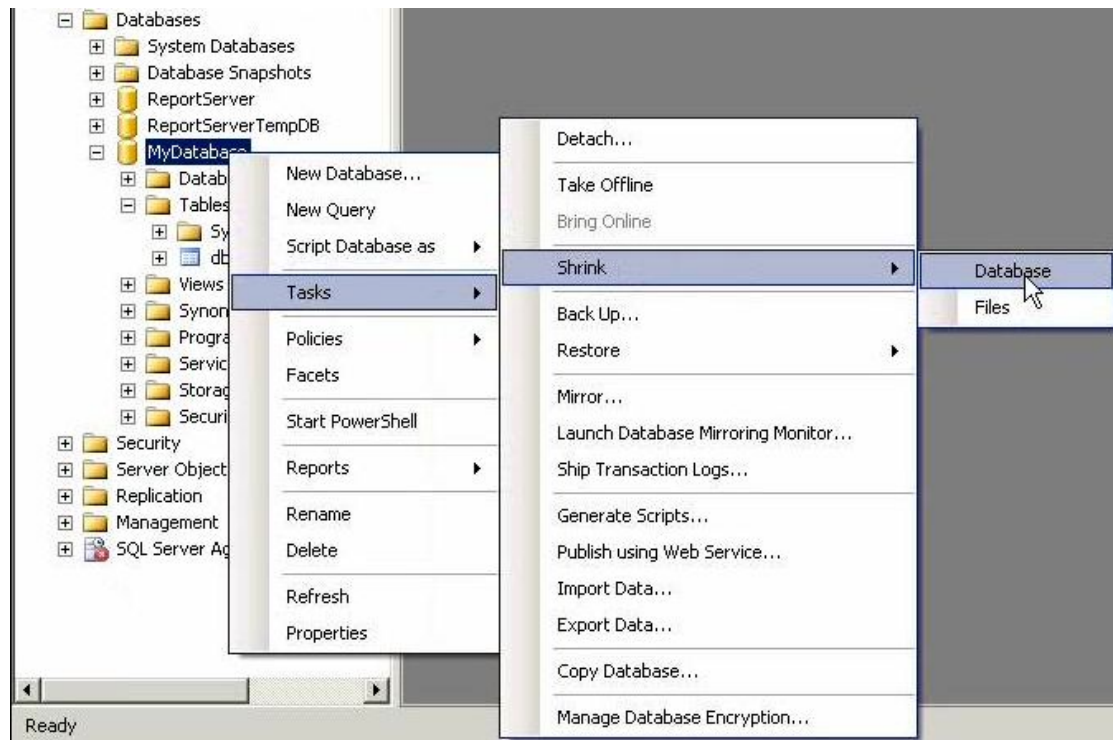
تقليص (Shrink) حجم ملف قاعدة البيانات:

عند إنشاء قاعدة بيانات جديدة يقوم SQL Server بإنشاء ملفات (أو ملف) قاعدة البيانات، ويحجز لذلك مساحة من القرص (افتراضياً 3 Mb للملف MDF، و 1 Mb للملف LDF).

يمكنك التحكم في حجم ملفات قاعدة البيانات عند إنشائها، كما يمكنك تقليص أحجام الملفات لحجز مساحة أقل.

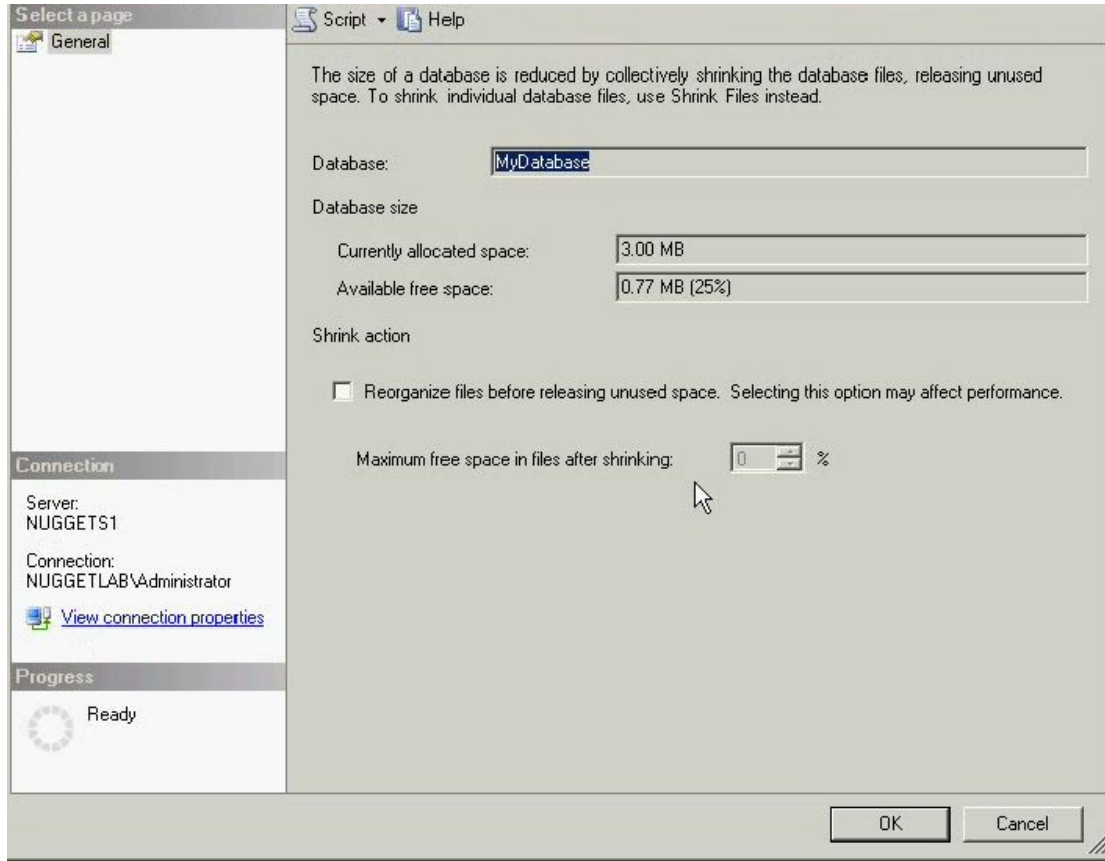
للقيام بذلك باستخدام SQL Server Management Studio حدد قاعدة البيانات من Object

Explorer ثم Database -> Shrink -> Tasks



يمكنك تقليص حجم جميع ملفات قاعدة البيانات باختيار Database، أو تقليص حجم أحد الملفات (MDF أو LDF) باختيار Files، مع بعض الخيارات الإضافية.

تقليص حجم الملفات يعني حذف الجزء غير المستغل من قبل قاعدة البيانات، أيضاً يمكن تفعيل خاصية التقليل التلقائي (Auto Shrink) من خصائص قاعدة البيانات.



في الصورة Currently allocated space يمثل الحجم الإجمالي لجميع الملفات. الحجم Available free space بنسبة 25% تمثل المساحة غير المستغلة من الملفات.

كيفية الحصول على الاستعلامات:

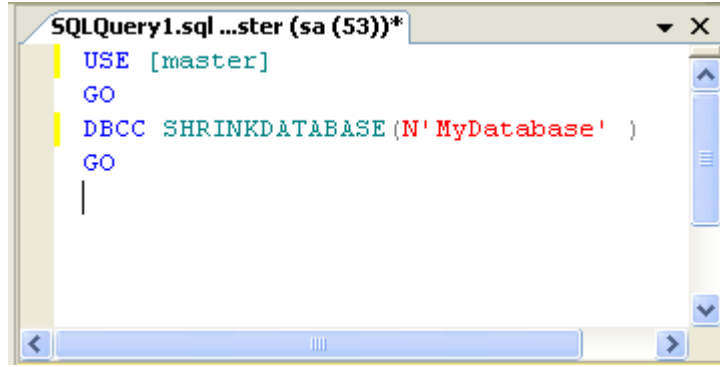
الآن، كيف نقوم بتقليص حجم الملفات باستخدام SQL؟

لاحظ، في جميع نوافذ الخيارات SQL Server Management Studio هناك زر Script وبه قائمة منسدلة تحمل بعض الخيارات الإضافية.



من خلال هذا الزر يمكنك الحصول على الاستعلام الذي يؤدي إلى تنفيذ الخيارات المحددة في نافذة الخيارات.

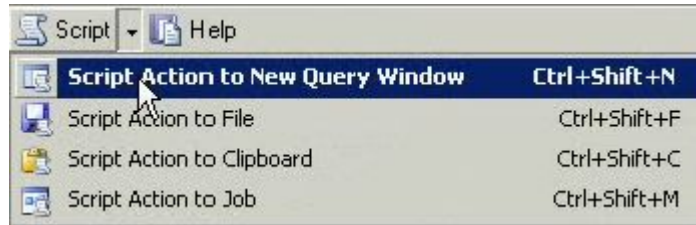
مثلا لو نظرنا في النافذة السابقة على الزر Script لحصلنا على استعلام تقليص حجم ملفات قاعدة البيانات:



```
SQLQuery1.sql ...ster (sa (53))*
USE [master]
GO
DBCC SHRINKDATABASE (N' MyDatabase' )
GO
|
```

بنفس الطريقة يمكنك الحصول على الاستعلام المقابل لتنفيذ أي عملية يمكنك القيام بها من خلال SQL Server Management Studio.

في القائمة المنسدلة بعض الخيارات الإضافية لحفظ الاستعلام في ملف أو إلى الحافظة:



أو نقله إلى قائمة المهام (Script Action to Job) حيث هناك المزيد من الخيارات المتقدمة، منها على سبيل المثال: القيام بخطوات إضافية (Steps)، إرسال إشعارات (Notifications)، تنفيذ المهمة في جدول زمني (Schedules) وتوكيل SQL Server Agent¹ للقيام بذلك بشكل آلي، وغير ذلك...

إلى هنا تنتهي دروس هذه السلسلة، وبذلك تتكون لديك نظرة عامة عن كيفية التعامل مع محرك قواعد البيانات Microsoft SQL Server باستخدام Microsoft SQL Server Management Studio، للمزيد من المعلومات طالع SQL Server Books Online.

انتهى

¹ SQL Server Agent غير متوفر في بعض نسخ SQL Server مثل نسخة Express Edition.

إلى هنا ينتهي محتوى الكتاب بعد أن تجول في أهم النقاط الأساسية التي ينبغي أن يعرفها من أراد تعلم برمجة قواعد البيانات باستخدام SQL Server، تبقى المسائل التفصيلية التي لم يتعرض لها الكتاب متوفرة في مصادر الدعم الفني أو المواقع الرسمية.

تم إعداد هذا الكتاب الإلكتروني ليوزع على هذه الهيئة مجاناً ولا يجوز استخدامه لأغراض تجارية، كل الحقوق محفوظة لأصحابها.

والصلاة والسلام على نبينا محمد وآله وصحبه أجمعين.

لا تنسوا إخوانكم بالدعاء.

تم بحمد الله في شعبان 1431 الموافق لـ أغسطس 2010.

فهرس

3	مدخل إلى Microsoft SQL Server
32	البداية مع Microsoft SQL Server
51	مدخل إلى قواعد البيانات
75	العمليات في SQL
86	المتغيرات وأنواع البيانات
104	أنواع العبارات SQL
118	مدخل إلى الدوال في SQL
133	الدوال المدمجة Built-In Functions
167	الجدول في قاعدة البيانات
188	السجلات في قاعدة البيانات
218	تحليل البيانات
242	تحليل البيانات (الجزء الثاني)
254	عمليات تحليل البيانات
276	العلاقات وتكامل البيانات
294	ربط البيانات
324	الإجراءات المخزنة
353	Views
373	الفهارس Indexes
383	القوادم Triggers
397	مسائل أخرى في SQL Server