

تكون قواعد المعلومات فرعاً أساسياً من فروع المعلوماتية، وتعتبر الأدوات التي تقدمها هامة لتطوير أنواع عديدة من التطبيقات المعلوماتية وخاصة في التطبيقات الإدارية والمالية والنظم المساعدة في اتخاذ القرار.

- **قاعدة المعلومات (Database -DB)**

مجموعة متكاملة من المعلومات المتعلقة ببعضها البعض (البيانات سواء أكانت **data** أو **metadata**).

وبالطبع هذه البيانات سوف تنظم في مجموعة من الملفات وينظم التعامل مع هذه الملفات ببرنامج نسمي نظام إدارة قواعد المعلومات **DBMS**.

- **قواعد المعلومات العلائقية (Relational Databases)**

هي أحد أنواع قواعد المعلومات التي تعتمد على النموذج العلائقي (وهناك نماذج أخرى مثل النموذج الشبكي، النموذج المرمي،...)، وعلاقتي يعني أن البيانات منظمة في جداول بأعمدة وصفوف وبالتالي ستكون قاعدة البيانات مكونة من الجداول، وتعتمد على مفاهيم الجبر العلائقي (والذي يتمثل ببساطة بعمليات التقاطع، الاجتماع، الفرق، المجموعات، ...الخ.

- **نظم إدارة قواعد المعلومات (Database Management Systems -DBMS)**

من الأخطاء الشائعة، إطلاق اسم قواعد المعلومات على أنظمة إدارة قواعد المعلومات العلائقية. فأنظمة **Oracle**, **MS SQL server**، **MS Access** هي أنظمة إدارة قواعد المعلومات ، والمهدف منها توفير بيئة ملائمة تمكننا من تخزين المعلومات ضمن قاعدة المعلومات واسترجاعها والعديد من المهام، ومن أهم مهامها:

١. تعريف بـ تخزين المعلومات.

٢. إيجاد التقنيات الملائمة للتعامل مع المعلومات المخزنة.

٣. تقديم نظم أمن وحماية للمعلومات المخزنة من الوصول غير المشروع.

٤. تحجب التضارب في المعلومات نتيجة تشارك عدة مستثمرين في الوصول إلى المعلومات.

يتم التواصل مع أنظمة قواعد المعلومات باستخدام لغة الاستعلامات البنوية **SQL( Structure Query Language)**

- **توضيف قاعدة المعلومات:**

يوجد ثلاث مستويات لتوضيف قاعدة المعلومات، ويقوم نظام إدارة قواعد المعلومات بالترابط بين هذه المستويات.

١. مستوى المخطط المفاهيمي: يظهر عناصر المعلومات التي ستقوم قاعدة المعلومات بإدارتها ويعبر عنه بعدة طرق منها.

### Entity Relationship Diagrams – ERD

٢. المستوى المنطقي: يتم فيه توصيف قاعدة المعلومات بلغة قياسية تمكن من التعامل مع المعلومات مثل لغة

### **Data Definition Language - DDL**

٣. المستوى الفيزيائي (الداخلي): توصف الطريقة الفعلية لتخزين المعلومات.

- **لغات المعلومات:**

يوفر كل نظام إدارة معلومات على الأقل لغة واحدة تتيح لمستخدميه تعريف بنية قاعدة المعلومات، وصلاحيات الوصول إلى المعلومات، وغيرها، وتعتبر لغة الاستفسارات البنوية SQL لغة قياسية للتعامل مع قواعد المعلومات العلائقية، وهي لغة غير إجرائية، وإنما لغة تصريحية، وتقسام إلى:

#### **١. لغة استعلام المعلومات (Data Query Language - DQL)**

تستخدم لاستفسار عن البيانات واستعادتها من قاعدة المعلومات.

#### **٢. لغة تعريف المعلومات (Data Definition Language – DDL)**

تستخدم لتعريف كائنات قواعد المعلومات

#### **٣. لغة التعامل مع المعلومات (Data Manipulation Language – DML)**

تستخدم للوصول والتعامل مع المعلومات مثل تعديلها أو إضافتها أو حذفها.

#### **٤. لغة التحكم بالمعلومات (Data Control Language – DCL)**

تستخدم من أجل منح أو سحب الصلاحيات.

#### ملاحظة:

قاموس المعلومات (Data Dictionary): يعرف بأنه ملف يحوي meta data (المعلومات السامية) وهي معلومات عن المعلومات تستدعي قراءة أو تعديل المعلومات في نظم قواعد المعلومات.

**:SQL Server •**

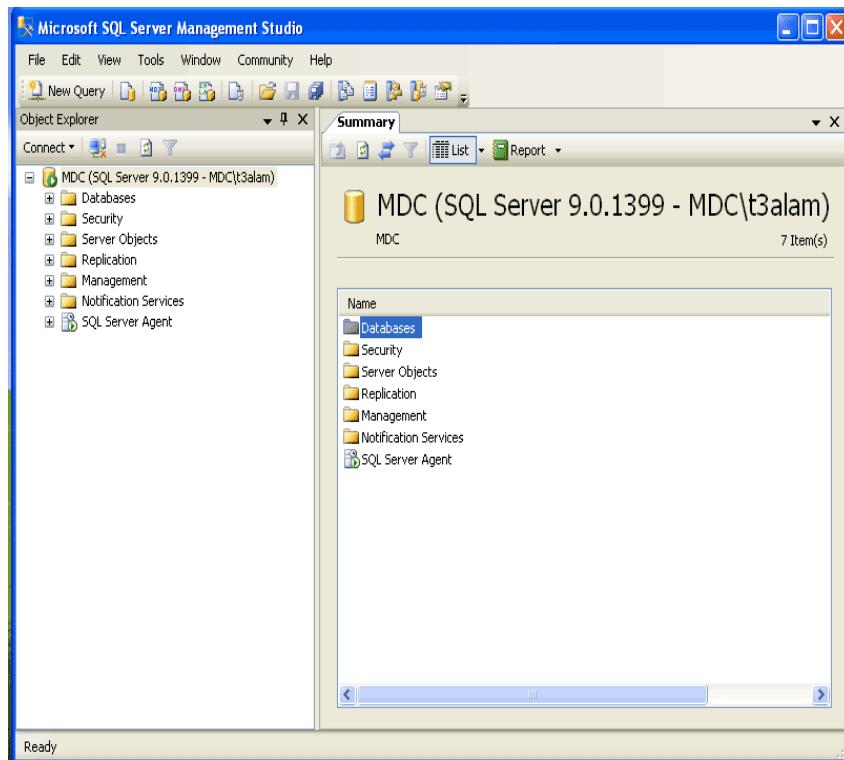
بعد تنزيل SQL Server على جهازك اذهب إلى start ثم programs ثم SQL Server Management Studio كما في الشكل.



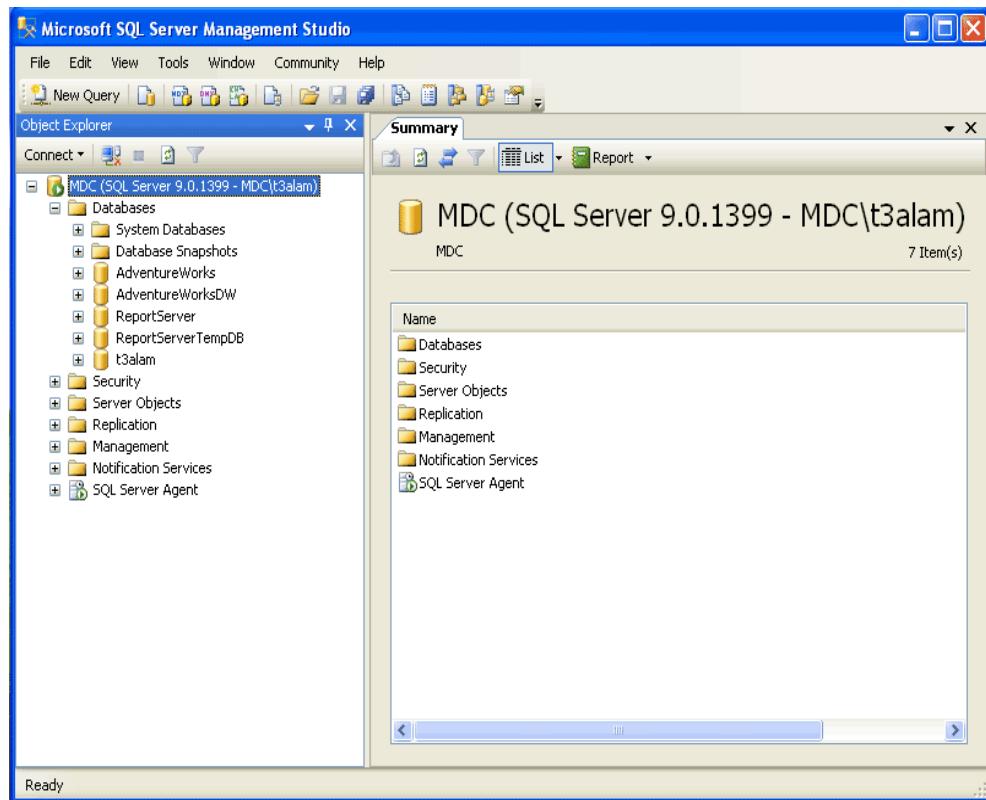
بعد ذلك قم بالاتصال بالserver، وإن كنت على نفس الجهاز اضغط connect .

ملاحظة: window authentication تعني العمل على الجهاز الحالي

. فيجب تحديد login و password وذلك لوجود أكثر من مستخدم .



وبالنقر على علامة الزائد بجوار System Databases سترى مجلد باسم Databases يحتوي على قواعد بيانات النظام والتي هي:



١. **قاعدة البيانات master**: تعتبر أهم قاعدة بيانات تحتوى على كل المعلومات الخاصة بالمستخدمين وكلمات سرهم وصلاحياتهم في التعامل مع قواعد البيانات ، أماكن وجود ملفات الـ **Data** ، إعدادات قواعد البيانات ، رسائل الأخطاء ..

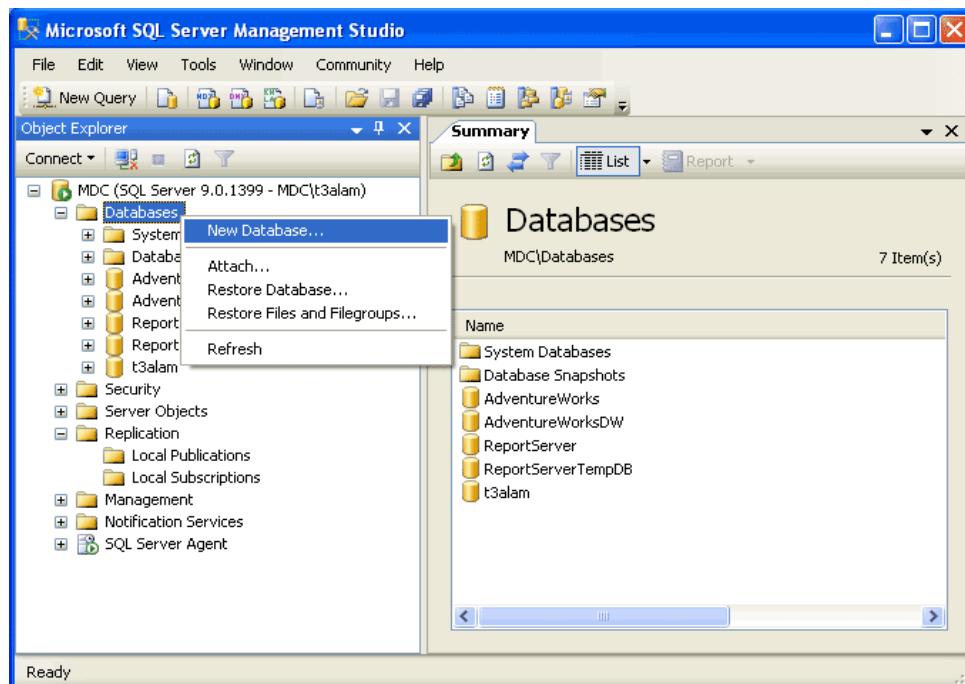
٢. **قاعدة البيانات tempdb** : قاعدة بيانات تنشأ عند تشغيلك الـ **server** وتوضع فيها البيانات المؤقتة التي يحتاج الـ **server** لتخزينها مؤقتاً أثناء عمل الإجراءات المخزنة مثلا .. وعند إغلاق الـ **Sql server** وإعادة فتحه تضيع هذه البيانات وتصبح قاعدة البيانات **tempdb** فارغة وجاهزة لاستقبال بيانات مؤقتة جديدة.

٣. **قاعدة البيانات model** : هي عبارة عن موديل يمكنك التعديل فيه حيث أن كل قاعدة بيانات جديدة تتشكل من حفظ كل خصائصها المبدئية من هذه القاعدة ، إذ يمكنك تعديلها لتتناسب احتياجاتك في قواعد البيانات الجديدة من حيث الحجم مثلاً أو إنشاء جدول افتراضي بما ليتم إنشائه تلقائياً داخل أي قاعدة بيانات جديدة تقوم بإنشائها.

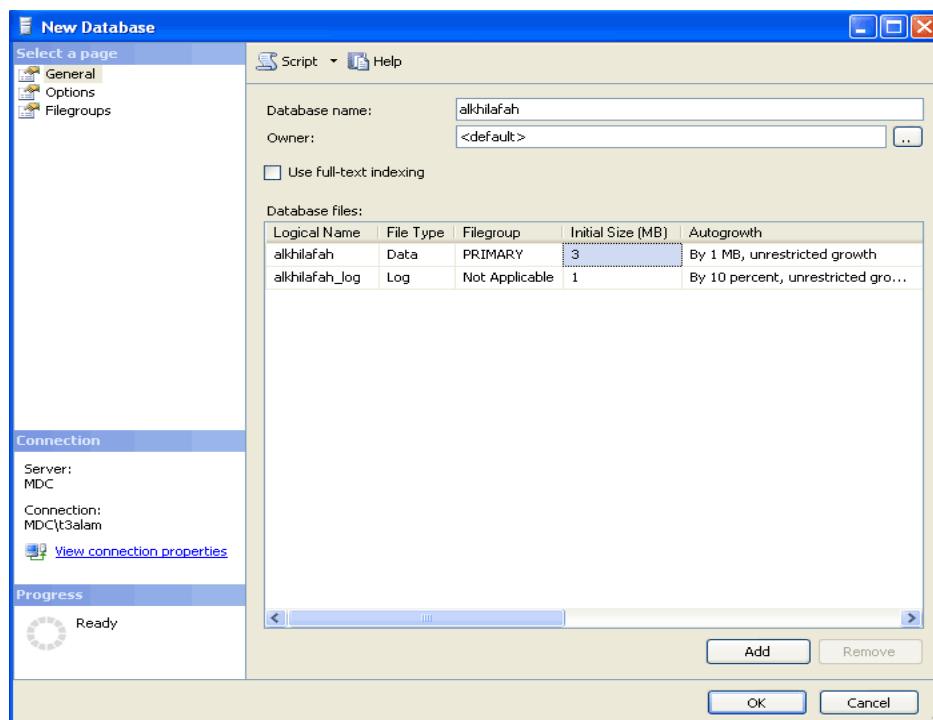
٤. **قاعدة البيانات msdb** : وهي المسؤولة عن كل الأوامر والوظائف التي سيقوم بها الإنسان الآلي المساعد في الـ **SQL Server** وهو أي تفاصيل كل الوظائف التلقائية التي تريد تنفيذها بالتعاون مع الـ **Windows** على قاعدة البيانات أو على أي شيء متعلق بالـ **Windows** بناء على حدث معين يحدث في قاعدة البيانات أو وقت معين .

• إنشاء قاعدة بيانات:

بالنقرة بواسطة الزر اليميني للماوس على **New Database** و اختيار **Databases** كما في الشكل التالي.



نتيجة ذلك يظهر مربع حوار جديد لإنشاء قاعدة بيانات جديدة كما في الشكل التالي.



نستطيع من خلاله تحديد اسم قاعدة المعلومات ومالك هذه القاعدة( وهو المستخدم الذي لديه كل صلاحيات التعامل مع قاعدة المعلومات ) ولو تركها سيكون مستخدم الـ **Sql server** الأساسي هو مالك هذه القاعدة .

- بالتأشير على المربع **use full-text indexing** وهو المسؤول عن تنشيط عملية البحث داخل قاعدة المعلومات عن الكلمات أو الجمل داخل النصوص.

- نجد أن هناك ملفان يتم إنشاؤهما من قبل السيرفر لقاعدة المعلومات الأول (ملف **Data**) يملك الامتداد **MDF** وهو المسؤول عن تخزين المعلومات.

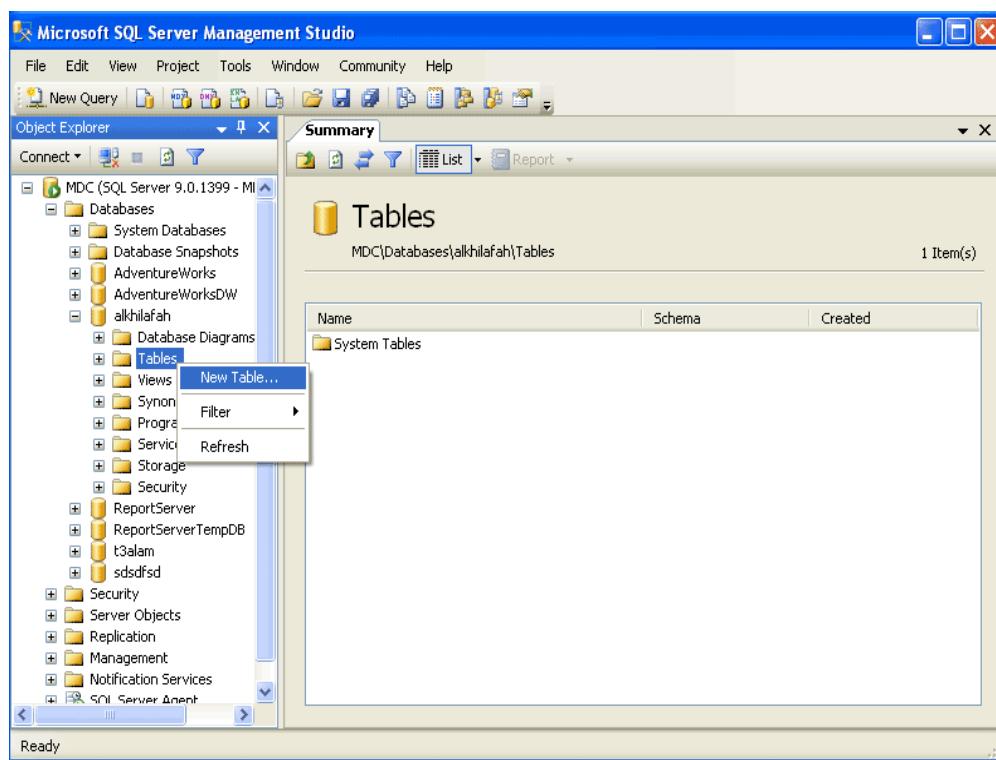
الثاني (ملف **Log**) يملك الامتداد **LDF** وهو المسؤول عن تخزين أي تعديلات على قاعدة المعلومات ، ويعتبر المصدر الأساسي لإرجاع البيانات المفقودة من ملف المعلومات.

ويمكنك إضافة ملفات أخرى عن طريق الزر **add** .

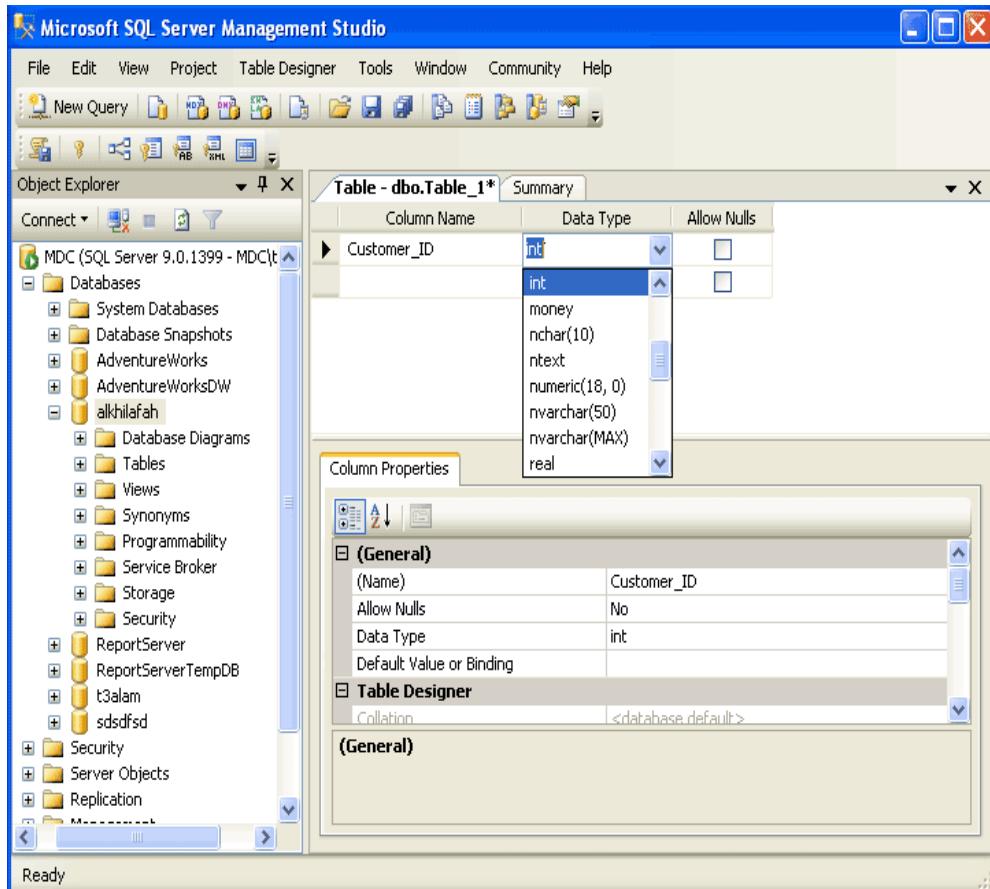
ويمكنك من خلال نافذة إنشاء قاعدة المعلومات تحديد الحجم الابتدائي للملف ومقدار الزيادة في حجمه كلما زادت المعلومات المخزنة فيه.

## • إنشاء جداول قاعدة المعلومات:

فأصل مجده قاعدة المعلومات يوجد مجلد يسمى **Tables** وهو الجلد الذي يحوي جداول قاعدة المعلومات وعندما قام بالنقرة بواسطة الزر اليميني للماوس واختار من القائمة **New Table**



فتظهر نافذة تصميم الجداول والتي نستطيع من خلالها تحديد أسماء حقول الجدول ونوعها وخصائصها.



## • أنواع المعمليات:

الأعداد:

name	Size	domain	Description
Int	4 B	-2 <sup>31</sup> to 2 <sup>31-1</sup>	
bigint	8 B	-2 <sup>63</sup> to 2 <sup>63-1</sup>	
smallint	2 B	to 2 <sup>15-1</sup> -2 <sup>15</sup>	
tinyint	1 B	0-255	صغير جداً
Decimal أو numeric			الأرقام العشرية decimal(8,0) - decimal(8,2) ٨ عدد الخانات الكلي ومجمله من ١ إلى ٣٨ . والقيمة الافتراضية ١٨ ٢ عدد الخانات بعد الفاصلة ومجمله من الصفر إلى عدد الخانات الكلي و القيمة الافتراضية .

<b>money</b>	<b>8 B</b>		خانات عشرية ٤ (بعد الفاصلة)
<b>smallmoney</b>	<b>4 B</b>		خانات عشرية ٤ (بعد الفاصلة)
<b>Bit</b>	<b>1 b</b>		للقيم المنطقية True=1 False=0

الأعداد التقريرية:

<b>name</b>	<b>Size</b>	<b>Domain</b>
<b>Float</b>	<b>8 B</b>	<b>- 1.79E+308 to -2.23E-308</b>
<b>real</b>	<b>4 B</b>	<b>-3.40E + 38 to -1.18E - 38</b>

الوقت و التاريخ:

<b>name</b>	<b>Domain</b>
<b>smalldatetime</b>	<b>January 1, 1900, through June 6, 2079</b>
<b>datetime</b>	<b>January 1, 1753, through December 31, 9999</b>

السلال الحرفية:

<b>name</b>	<b>Size</b>	<b>domain</b>	<b>Description</b>
<b>Char</b>		<b>1-8000 char</b>	
<b>varchar</b>		<b>1-8000 char</b>	سلسلة حرفية تحجز الحجم المخاض بالتخزين بشكل ديناميكي
<b>Text</b>	<b>2 GB</b>		

السلال الحرفية من النوع Unicode

<b>name</b>	<b>Size</b>	<b>Domain</b>
<b>nchar</b>		<b>1-4000 char</b>
<b>nvarchar</b>		<b>1-4000 char</b>
<b>ntext</b>	<b>1 GB</b>	

السلال الثنائية : (Binary Strings)

<b>name</b>	<b>domain</b>	<b>Description</b>
<b>binary</b>	<b>٨٠٠٠ بايت</b>	

<b>Varbinary</b>		
<b>image</b>		Binary (img,video,Sound)

أنواع أخرى:

<b>name</b>	<b>Description</b>
<b>xml</b>	خاص بلغة التأشير الموسعة
<b>Timestamp</b>	مؤشر زمني لآخر التغييرات على السجل وبالتالي يزداد العدد عند كل عملية تعديل ويحذف كنوع binary
<b>Uniqueidentifier</b>	يحذف كـ binary ويقبل رموز ست عشرى وهي قيمة فريدة

## • ملاحظات:

## ١. كيفية إنشاء حقل ذو قيمة تزايدية :Auto increment

من خواص الحقل نختار الخاصية **Is Identity=yes** ، ويمكن تحديد قيمة التزايد من خلال **Identity Increment** الخاصية

## ٢. لتحديد قيمة افتراضية لحقل :

نحدد القيمة الافتراضية للخاصية **Default value or binding**

٣. لنقل قاعدة البيانات يجب أخذ الخيار Tasks ومن ثم Detach (log) ldf (data)mdf

٤. للاتصال مع قاعدة ما يجب أخذ الخيار Attach ، وفي حالة عدم مشاهدة المخطط نأخذ خصائص القاعدة ومن ثم نختار Owner ونحدد الـ

## • أنواع المفاتيح (Keys)

## ١. المفتاح الرئيسي (Primary Key) :

a. Unique لا يسمح بقيم تكرارية أي يملك قيم فريدة لكل سجل.

b. لا يسمح بقيمة Null

c. يفيد في تحديد العلاقات بين الجداول.

d. لا يمكن أن نعرف إلا مفتاح أساسى واحد لكل جدول.

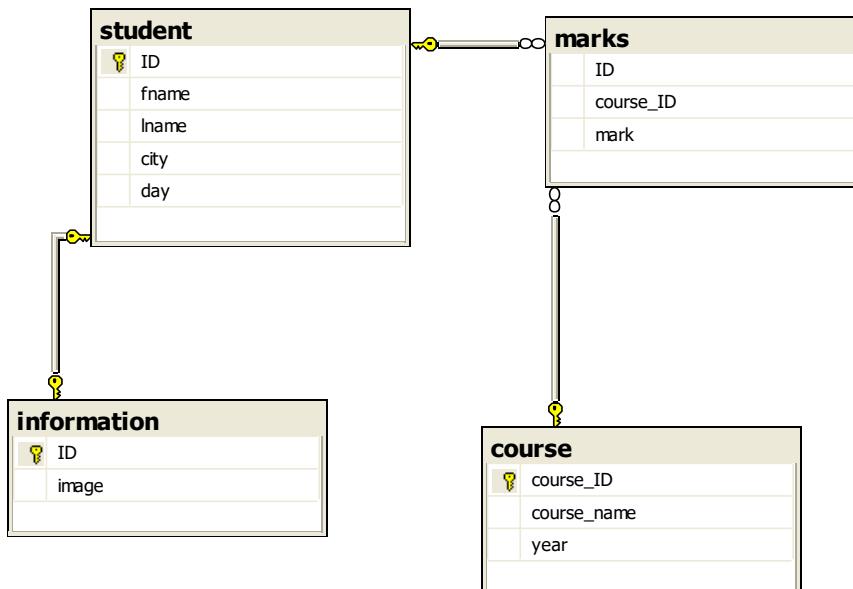
## ٢. المفتاح الثانوي (Foreign Key) :

a. يسمح بالقيم التكرارية

b. يستخدم لإظهار العلاقات بين الجداول ويضمن سلامة وتكامل البيانات، وهو يمثل الحقل في جدول الابن والذي يربط مع حقل الجدول الأب وهو حقل ذو مفتاح رئيسي، وبالتالي أي سجل يتم إضافته في الجدول الابن يجب أن يقابل سجل في الجدول الأب أو أن يحوي قيمة **null** في العمود الموافق.

c. يمكن ان يحوي الجدول ٢٥٣-٠ مفتاح ثانوي

d. السلوك الإفتراضي للمفتاح الثانوي أن يمنع حذف أي سجل من السجلات الأب عند وجود سجلات مرتبطة بهذا السجل في الجدول الابن؟ لكن أحياناً تحتاج إلى هذا الحذف الآلي، ويتم عن طريق **cascade on delete**

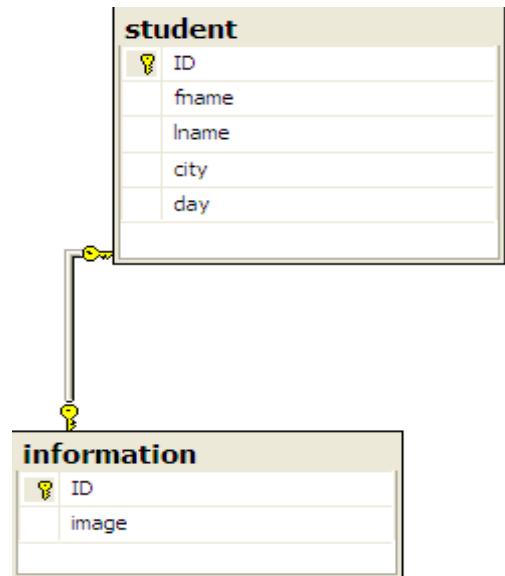


## • العلاقات ( relationships )

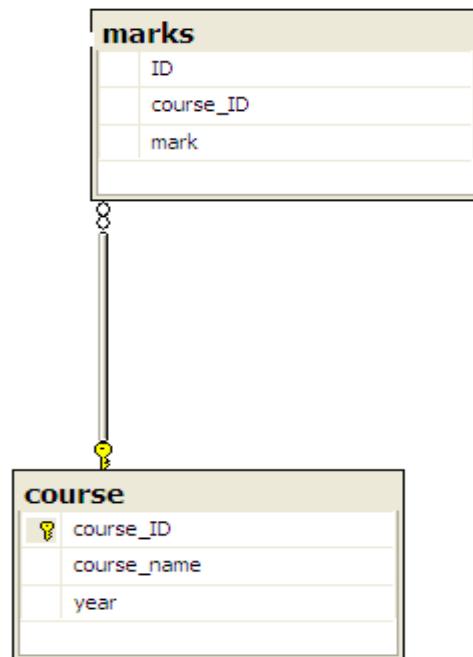
### ○ أنواع العلاقات:

#### 1. One to one .

مثال : جدول الطلاب وجدول يحوي صور عن الطلاب.

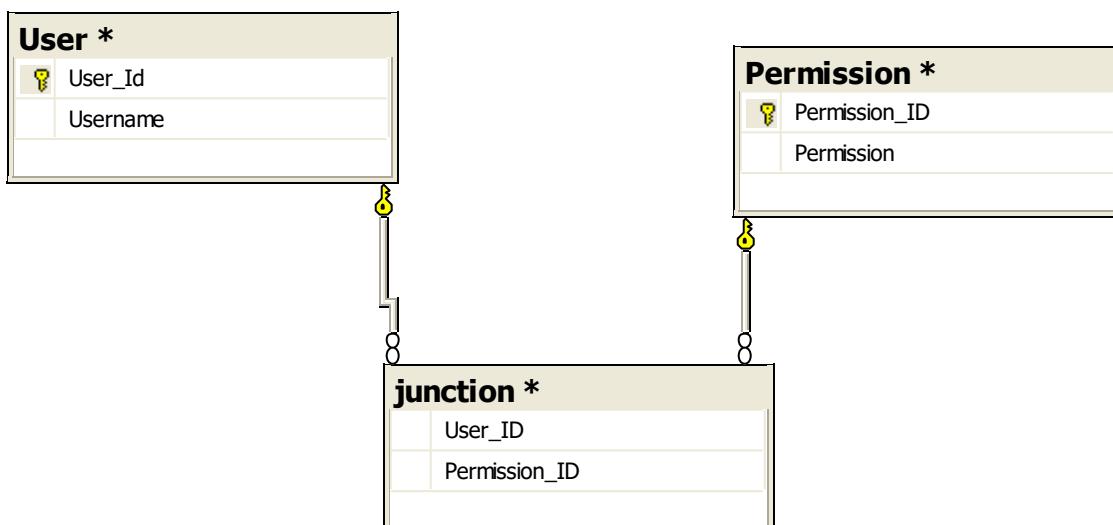
**:One to many .٢**

مثال : جدول الطلاب و جدول العلامات

**:Many to many .٣**

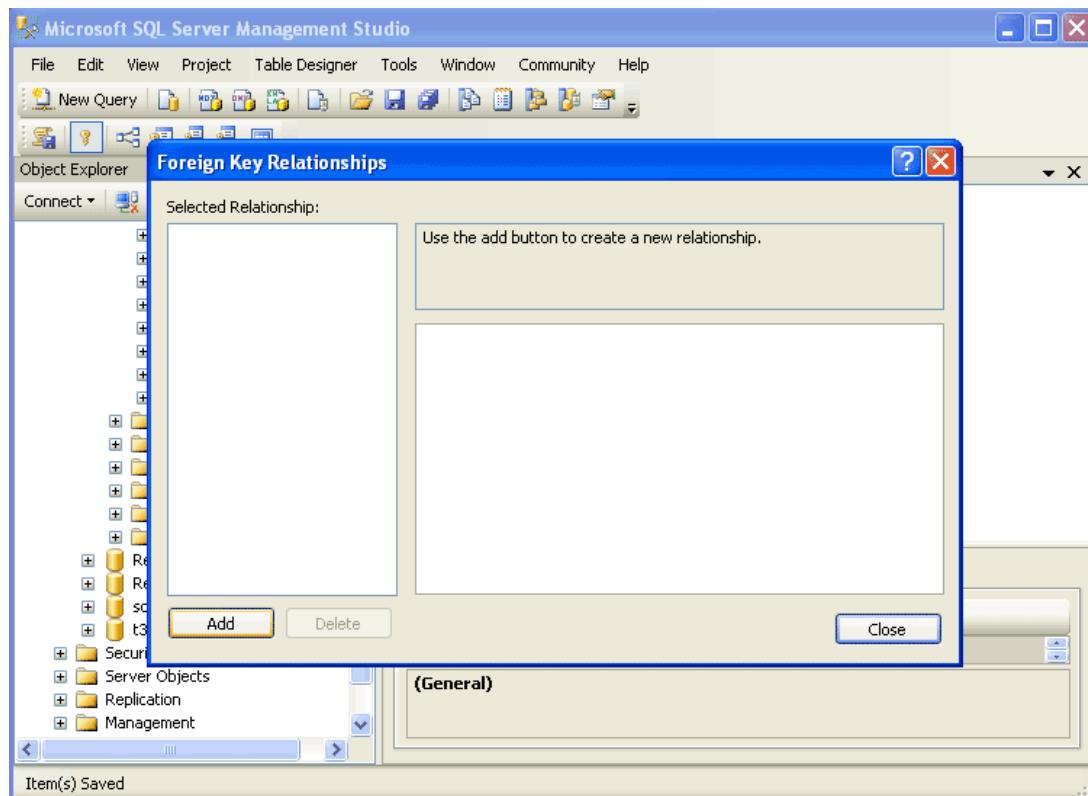
تنشأ العلاقة بين ثلات جداول أي يوجد جدول وسيط (جدول الملتقي) يخزن صفة متعلقة بالعلاقة وليس بالجدولين.

مثال : جدول المستخدمين وجدول السماحيات والجدول الوسيط يحوي رقم المستخدم ورقم السماحية .

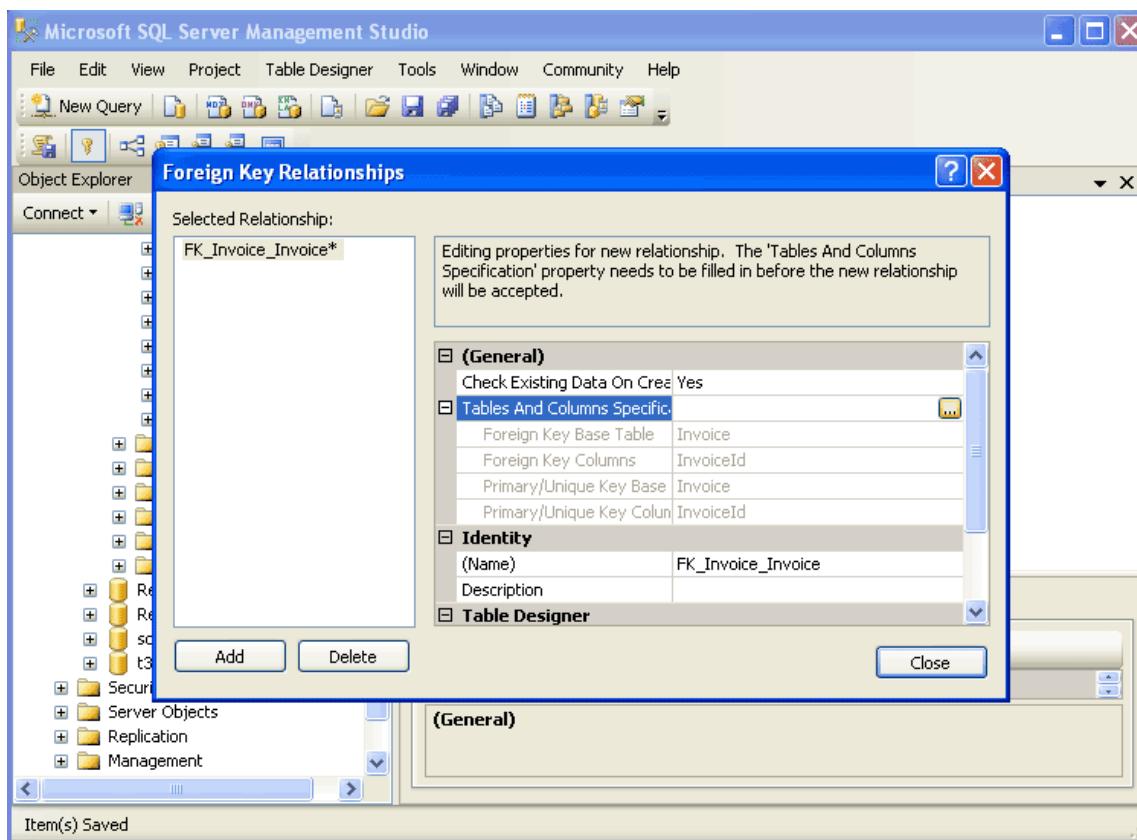


○ إنشاء العلاقات:

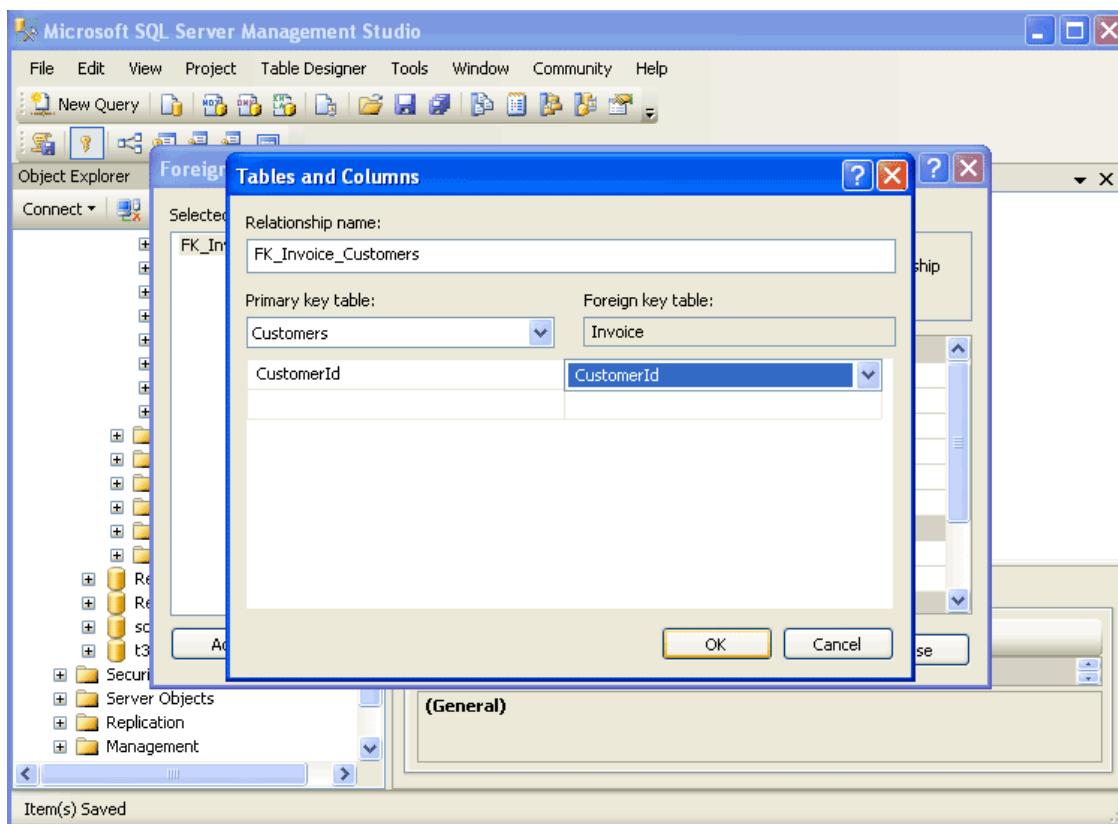
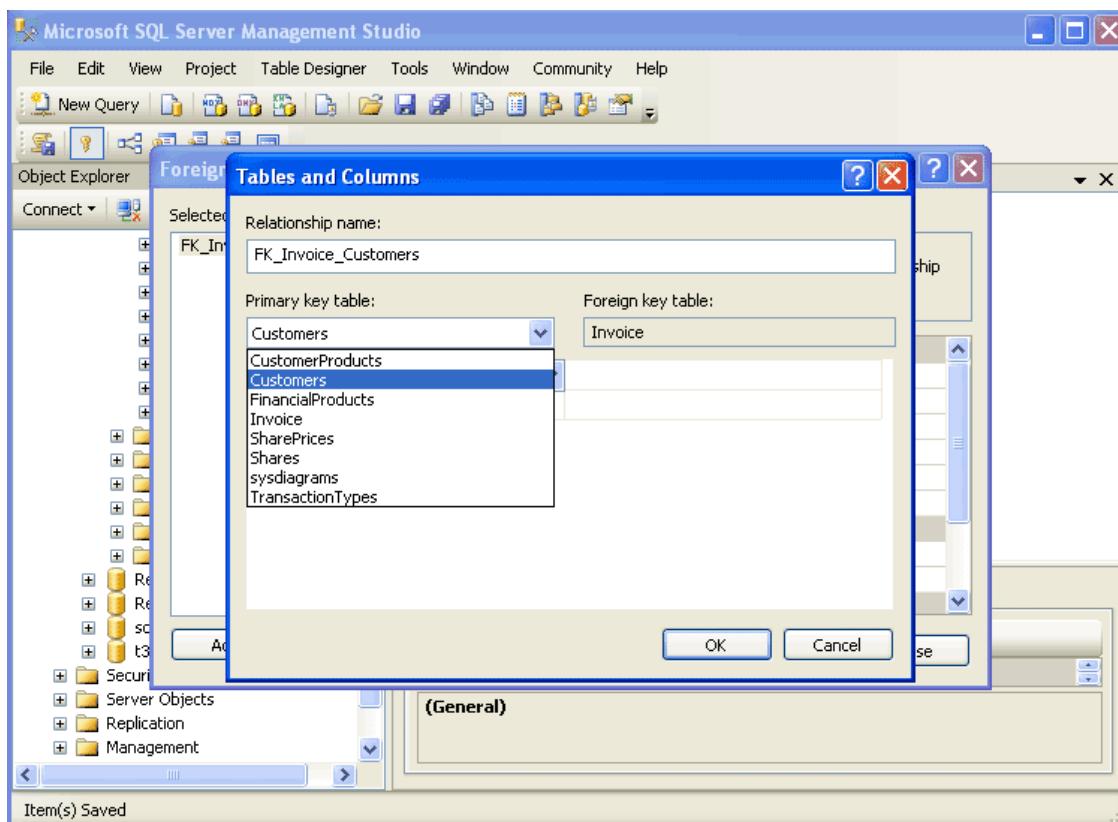
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left lists various database objects like System Tables, dbo.CustomerProducts, dbo.Customers, etc. The main area displays the 'Table - dbo.Invoice\*' designer. The table structure includes columns: InvoiceId (primary key, bigint), CustomerId (bigint), InvoiceType (int), DateEntered (datetime), Amount (numeric(18, 5)), ReferenceDetails (nvarchar(50)), Notes (nvarchar(MAX)), and RelatedProductId (int). A context menu is open over the table, with 'Relationships...' highlighted.

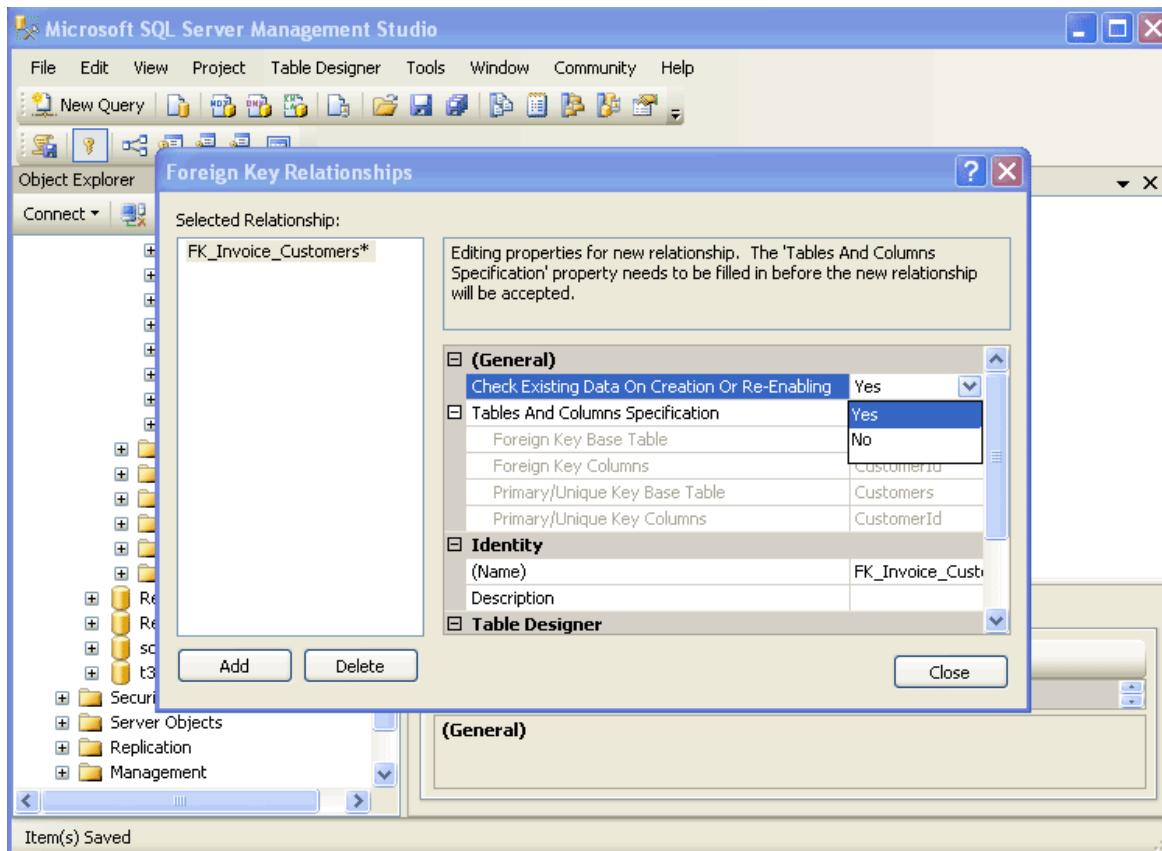


نقوم بضغط المفتاح add لإضافة علاقة جديدة



نقوم بفتح علامة الزائد بجوار tables and columns specification ثم نقوم بضغط الزر المنقط بجوارها ...



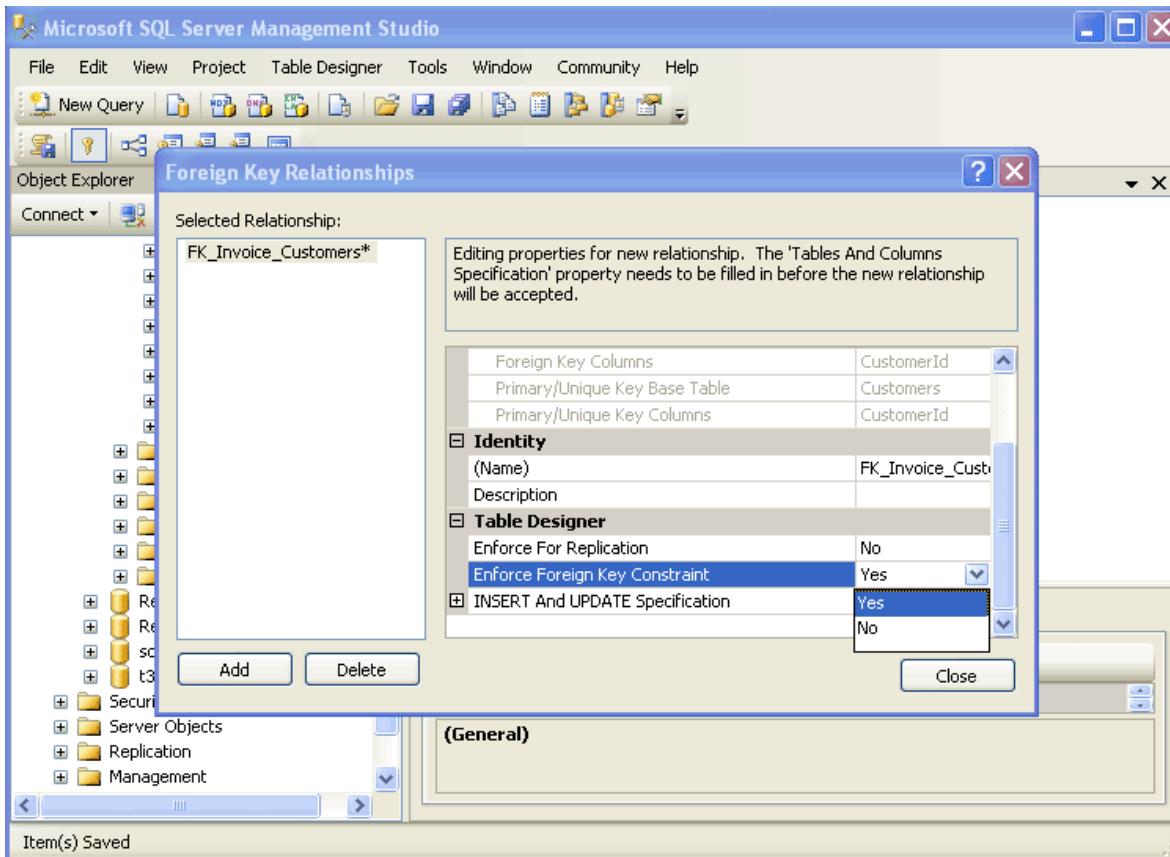


• في المجموعة **General** نجد أول خيار هو **Check Existing Data On Creation Or Re-Enabling**

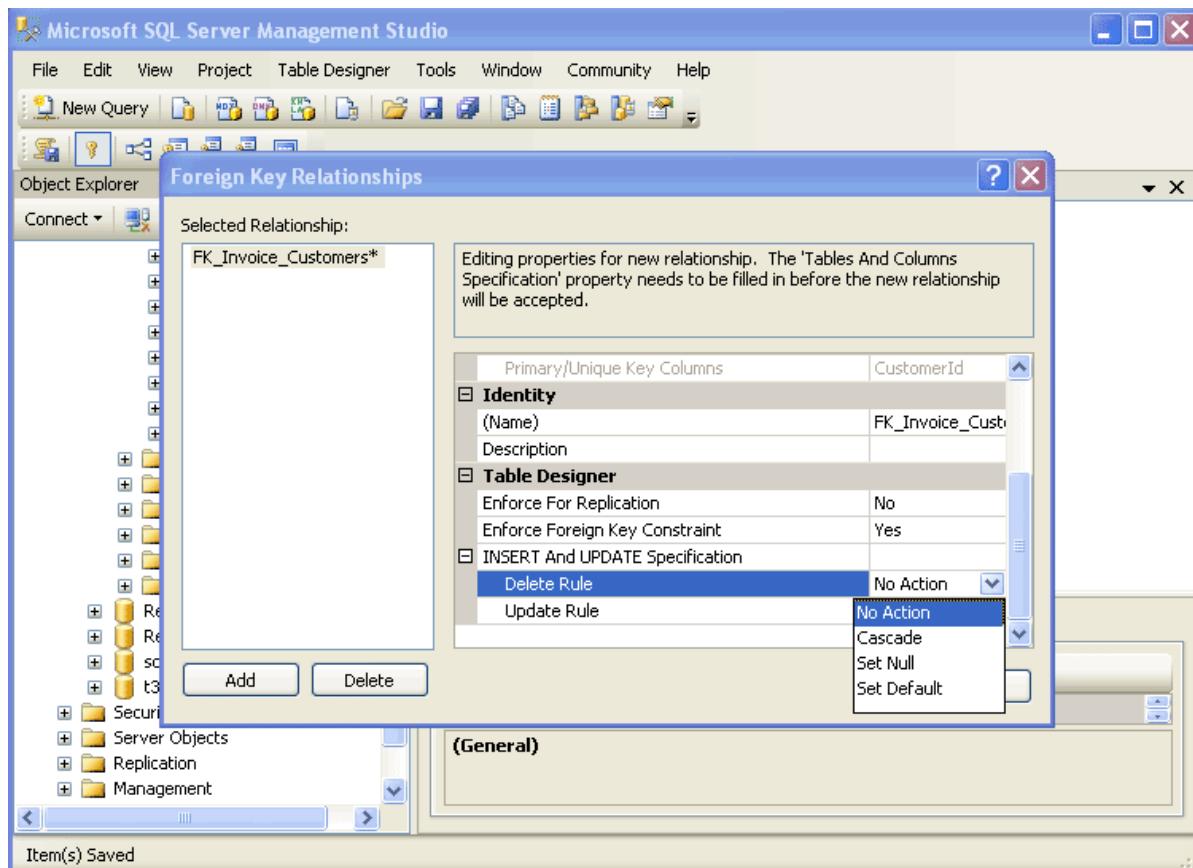
وهو المسؤول عن التأكد من أن بيانات الصف الذي سيستخدم في الربط متطابقة في الجداولين وذلك إن كنا نقوم بالربط وفي الصف بيانات وليس فارغا .. وعند اختيار **Yes** .. لابد أن تكون البيانات متطابقة وإلا فلن تتم عملية الربط بنجاح .

• أما الخيار **Enforce Foreign Key Constraints** فهو المسؤول عن جعل أي تغيير يحدث في أي صفات من الصفين يتم مثله في

الصف الآخر بالضبط كي يكون الصفين دائمًا متطابقين



- ومن الخيار الذي أسفل منه Insert And Update Specification نحدد طبيعة هذا التغيير بالنسبة إلى الرقم المسلسل وكل البيانات المتعلقة به.



- وهناك أربع خيارات في حالة الحذف أو التعديل .. هي بالترتيب

١ - No Action ... أي لا شيء يحدث .

٢ - Cascade ... أي شيء يحدث هنا يحدث مثله هناك .. أي لو حذفنا مستخدم هنا ستحذف كل الفواتير المتعلقة به " وهذا خيار خطير ولكن نحتاجه أحيانا " .

٣ - Set Null ... أي أنه عند حذف أو تعديل الرقم المسلسل مثلاً في جدول الزبائن سيتم استبداله بالقيمة Null في جدول الفواتير وذلك للحفاظ على بيانات الفواتير وللدلالة على أن الزبون الخاص بما تم حذفه أو تعديله " لازم الصد يكون يقبل القيمة Null " .

٤ - Set Default ... نفس عمل الخيار Null .. ولكن هنا يحط القيمة الافتراضية إلى أنت كنت عاملها للصف عند إنشاء الجدول . " لازم تكون واضع قيمة إفتراضية على الصد " .

• أما عن الخيار Enforce For Replication وهي عبارة عن صنع صورة طبق الأصل من قاعدة البيانات في مكان آخر .. وعند تغيير البيانات في أحدهم يتم نفس التغيير في قاعدة البيانات الأخرى

• ملاحظة:

لتغيير خواص العلاقة بين جدولين ، نختار بالزر اليميني على الجدول Relationship .