

العلاقات بين الجداول

من الأخطاء الشائعة التي يقع فيها المبرمجين الجدد ، و خاصة من لهم خبرة أكثر في استعمال جداول البيانات Spreadsheets أكثر من قواعد البيانات ، هو اهمال التوصيات الخاصة بتصميم الجداول و العلاقات بينها و وضع معظم البيانات في جدول واحد كبير

و مثال علي ذلك الجدول في الصورة التالية ، و هو مثال ممتاز للتصميم الخاطئ و هذا الجدول المقصود منه تسجيل حركة الدورات التدريبية ، أي الدورات التي حصل عليها كل موظف في الشركة

تاريخ النھو	تاريخ البدء	إسم الدورة	كود الدورة	رقم الموظف	مسلّم
30/01/2003	01/01/2003	الأكسس	ACC01	101	1
15/02/2003	01/02/2003	الأكسيل	EXC01	101	2
30/05/2003	01/05/2003	بلوربوينت	PPOINT2	101	3
30/01/2003	01/01/2003	الأكسس	ACC01	102	4
15/02/2003	01/02/2003	الأكسيل	EXC01	102	5
30/05/2003	01/05/2003	بلوربوينت	PPOINT2	102	6
30/01/2003	01/01/2003	الأكسس	ACC01	103	7
15/02/2003	01/02/2003	الأكسيل	EXC01	104	8
15/02/2003	01/02/2003	الأكسيل	EXC01	105	9
15/02/2003	01/02/2003	الأكسيل	EXC01	106	10

أي أن الموظف ذو الرقم المبين قد حضر دورة معينة و التي كودها و عنوانها و تواريخها كما هو مبين

و لنلاحظ أن اللون الأصفر يمثل بيانات دورة الاكسيل رقم ١ و التي تكررت بالكامل ٥ مرات في جدول لم تتعد سجلاته عشرة سجلات فهل ترون هذا منطقي؟؟

فما بالك لو كان لدينا دورة يحضرها مائة موظف ، هل سنكرر إدخال بياناتها مائة مرة؟؟ بالطبع لا ، و قبل أن نتطرق الي التصميم السليم ، تعالوا نفند عيوب جدول مثل هذا.

و الآن ما هي عيوب هذا التصميم؟؟

- تكرار المعلومات و استهلاك مساحة تخزينية لا داعي لها ، و يتطلب أيضا مجهود في ادخال بيانات مكررة
- صعوبة التغيير ، فلو أردت تعديل بيانات دورة ما فلا بد من تعديلها في سجلات كل من حضروها
- إحتمال الخطأ في بيانات الدورة من سجل لآخر أثناء الادخال
- صعوبة الحذف ، ماذا لو أردت حذف دورة ما ؟
- و أخيرا تحديث البيانات ، في كم سجل يتقوم بتحديث تاريخ البدء لدورة معينة لة اردت تغييره

التصميم القياسي للجدول

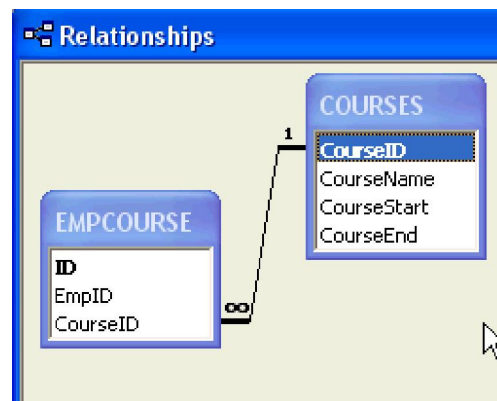
للتغلب علي المشاكل أعلاه يتم فصل الجدول الاصلي المسمي oneTable في المثال المرفق الي جدولين الاول به بيانات الدورات (سجل واحد لكل دورة) و الاخر به تسجيل لحركة الدورات عبارة عن كود الموظف و كود الدورة فقط بينما بيانات الموظفين موجودة في جدول الموظفين و بيانات الدورات موجودة في جدول الدورات ، فنضمن بذلك عدم التكرار بدون داعي للبيانات الخاصة بالموظفين أو الدورات مع كل تسجيل لموظف في دورة جديدة .

و يكون شكل الجدولان كالتالي

EMPCOURSE : Table			
كود الدورة	رقم الموظف	مسلسل	
ACC01	101	1	
EXC01	101	2	
PPOINT2	101	3	
ACC01	102	4	
EXC01	102	5	
PPOINT2	102	6	
ACC01	103	7	
EXC01	104	8	
EXC01	105	9	
EXC01	106	10	
	0	*	Imber)

COURSES : Table				
تاريخ النهر	تاريخ البدء	إسم الدورة	كود الدورة	
30/01/2003	01/01/2003	الأكسمن	ACC01	
15/02/2003	01/02/2003	الأكسمن	EXC01	
30/05/2003	01/05/2003	بالوروبونت	PPOINT2	

و بعمل علاقة بينها يمكن الحصول علي استعلام يناظر الجدول الأصلي مثلما في الصورة query1



Query1 : Select Query						
	تاريخ النھو	تاريخ البدء	إسم الدورة	كود الدورة	رقم الموظف	مسلسل
▶	30/01/2003	01/01/2003	الأكسس	ACC01	101	1
	30/01/2003	01/01/2003	الأكسس	ACC01	102	4
	30/01/2003	01/01/2003	الأكسس	ACC01	103	7
	15/02/2003	01/02/2003	الأكسس	EXC01	101	2
	15/02/2003	01/02/2003	الأكسس	EXC01	102	5
	15/02/2003	01/02/2003	الأكسس	EXC01	104	8
	15/02/2003	01/02/2003	الأكسس	EXC01	105	9
	15/02/2003	01/02/2003	الأكسس	EXC01	106	10
	30/05/2003	01/05/2003	بلوربونت	PPOINT2	101	3
	30/05/2003	01/05/2003	بلوربونت	PPOINT2	102	6

و سنعرض هنا في هذا الموضوع للعلاقات و تكوينها ، أما الاستعلام فسيأتي دوره لاحقاً بإذن الله

المهم هو أننا استطعنا تفادي الخمسة عيوب السابقة عن طريق فصل الجداول الي جداول قياسية ، مما يمنع تكرار البيانات أو يمنع ما يطلق عليه ب DATA REDUNDANCY

و الآن سنتحدث عن أنواع العلاقات المختلفة:

هناك ثلاث أنواع من العلاقات بين الجداول

١. علاقات من نوع ١ الي متعدد (رأس بأطراف - ولكن لا أحب هذه التسمية !! و أفضل متعدد)
٢. علاقات من نوع ١ الي ١
٣. علاقات من نوع متعدد الي متعدد

و لكي يسهل علينا وصف العلاقات ، سنعرف بالاضافة الي حقل المفتاح حقل آخر يسمى الحقل الغريب Foreign Key و الموجود فى الجدول الرئيسي و الذي سيتم ربطه بحقل المفتاح فى جدول فرعي.

<p>علاقات من نوع واحد الي متعدد</p>	<p>هذا هو النوع لأعم من العلاقات و فيه تكون القيم فى حقل المفتاح يناظرها قيمة أو أكثر فى الحقل الغريب فى الجدول الرئيسي</p> <p>و لكن أي قيمة فى الحقل الغريب فى الجدول الرئيسي يناظرها قيمة واحدة فقط فى حقل المفتاح فى الجدول الفرعي</p> <p>شكل العلاقة</p>
<p>علاقات من نوع واحد الي واحد</p>	<p>و فيه تكون القيم فى حقل الربط فى الجدول الثاني يناظرها قيمة واحدة فى الجدول الأول و العكس صحيح</p> <p>و هذا ليس نوعا نمطيا من العلاقات ، لأنه فى هذه الحالة يمكن ضم بيانات الجدولين فى جدول واحد ، و لكن قد نلجأ لذلك فى حالة الحاجة لعزل جزء من البيانات لأسباب أمنية أو تكون البيانات تمثل حالة خاصة</p> <p>شكل العلاقة</p>
<p>علاقات من نوع متعدد الي متعدد</p>	<p>و فيه تكون القيم فى حقل الربط فى الجدول الثاني يناظرها قيم عديدة فى الجدول الأول و العكس صحيح</p> <p>هذا جزء من ملف التعليمات بالانجليزية</p> <p>This type of relationship is only possible by defining a third table (called a junction table)</p>

whose primary key consists of two fields — the foreign keys from both Tables A and B.

هذا النوع من العلاقات لا يمكن حدوثه الا بتعريف جدول جديد يسمى جدول **الوصلة** ، و يكون المفتاح في جدول الوصلة هو الحقلين الغريبيين في كلا الجدولين المراد ربطهما بهذا النوع من العلاقات

A many-to-many relationship is really two one-to-many relationships with a third table.

إذا العلاقة من نوع متعدد الي متعدد هي حقيقة علاقتين من نوع واحد الي متعدد مع جدول ثالث

For example, the Orders table and the Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.

مثال : جدول الطلبيات و المنتجات بينهما علاقة من نوع متعدد الي متعدد يتم عملها عن طريق جدول تفاصيل الطلبيات .

الطلبية الواحدة قد يكون فيها أكثر من منتج ، و كل منتج يكون موجود في عدة طلبيات

و لكن هنا العلاقة بين جدولي الطلبيات و المنتجات تكون من خلال جدول **الوصلة** و الذي هو جدول تفاصيل الطلبيات في

	<p>هذه الحالة</p> <p>شكل العلاقة</p>
--	--

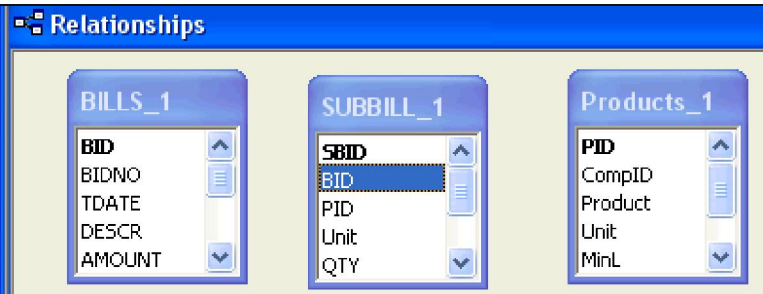
إنشاء العلاقات المختلفة:

<p>من قائمة</p> <p>Tools</p> <p>نختار Relationships</p>	
<p>يظهر مربع حوار لاختيار الجداول المتاحة في القاعدة ، و نختار الجدول ثم نضغط ADD</p> <p>أو ننقر نقر مزدوج علي اسم الجدول فيظهر في قاعدة البيانات</p>	
<p>و ان أردنا إضافة جدول بعد ذلك نختار أي منطقة خالية في شاشة البيانات و نختار show Table و لحذف جدول من لوحة العلاقات</p>	

نختاره و نضغط

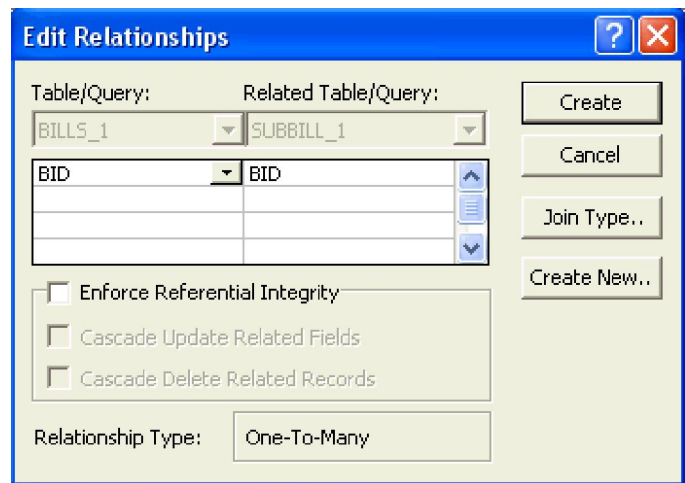
Delete زر

تظهر الجداول فى لوحة الجداول
بدون علاقات



لبناء علاقة واحد الي متعدد
و بعد ذلك لعمل العلاقة نختار الحقل
الغريب فى الجدول الثاني و نسحبه
الي نظيره المفتاح فى الجدول
الرئيسي

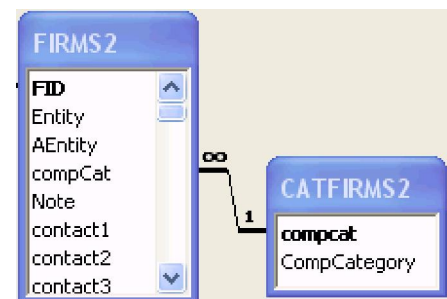
فيظهر مربع الحوار كما هو مبين
للتأكيد علي الحقول المختارة لانشاء
العلاقة



والمسمي مربع تحرير العلاقات

و بالضغط علي زر Create أو
إنشاء يتم انشاء العلاقة بين الجدولين
كما فى الشكل

و هذا هو شكل العلاقة من نوع
واحد الي مالانهاية



[شرحها](#)

بالضغط علي زر Create New

في مربع تحرير العلاقات يظهر مربع

الحوار التالي لكي نضيف منها

علاقات جديدة مع جداول أخرى

و فيه يتم تحديد كل من اسم الجدول

و الحقل للحقول المراد ربطها



بالضغط علي زر Join Type في

مربع تحرير العلاقات يظهر مربع الحوار

التالي لكي نحدد نوع العلاقة بين

الجدولين من حيث تواجد البيانات

علي طرفي العلاقة

و هي اما:

1. اختيار أن تظهر البيانات

الموجودة علي الطرفين فقط دون

الموجودة في أحدهما فقط

2. إختيار اظهار كل ما في

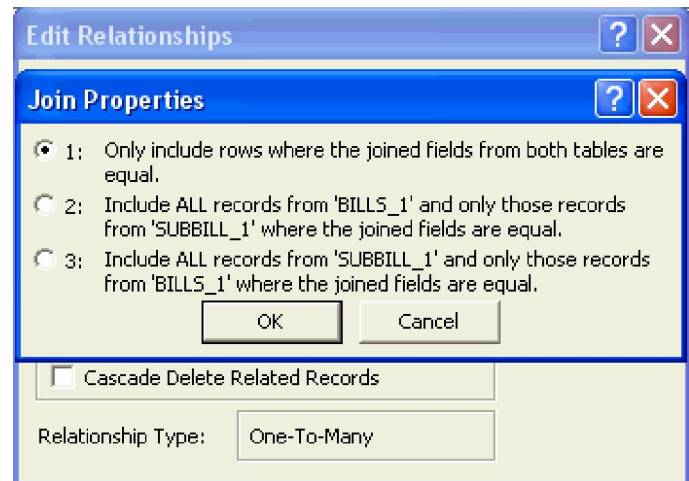
الجدول الرئيسي و ما يناظرها فقط

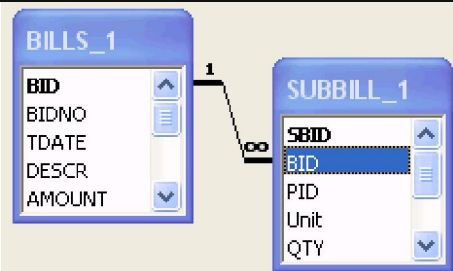
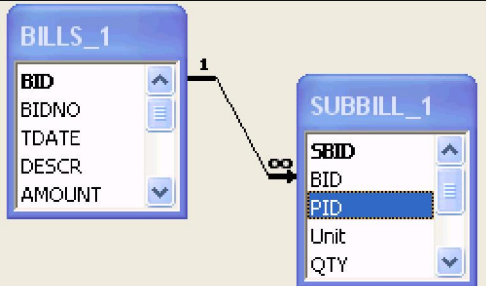
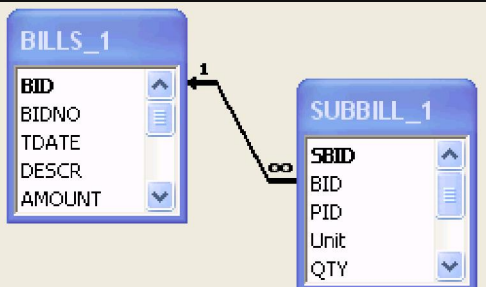
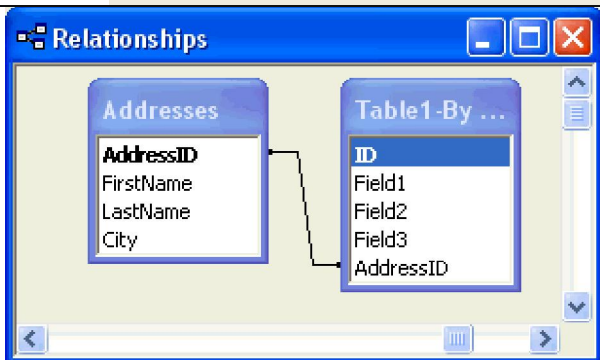
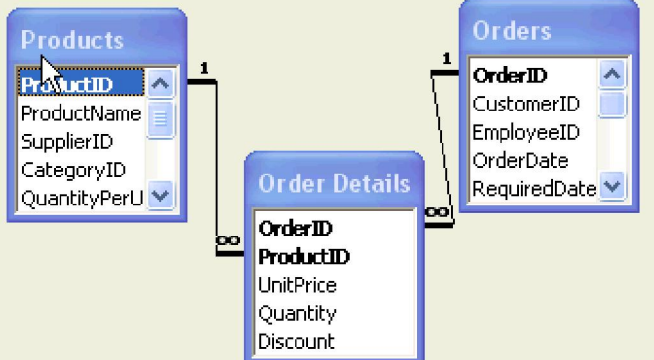
من بيانات في الفرعي

3. إختيار اظهار كل ما في

الجدول الفرعي و ما يناظرها فقط

من بيانات في الرئيسي



نتيجة الحالة ١	
نتيجة الحالة ٢	
نتيجة الحالة ٣	
هذا مثال علي شكل علاقة من نوع واحد الي واحد	
و هذا مثال علي شكل العلاقة من نوع متعدد الي متعدد كما	

Referential Integrity

التكامل المرجعي للبيانات ، هو نظام عبارة عن مجموعة قواعد يتبعها الأكسس لضمان أن العلاقة بين جدولين علاقة سليمة و أنك ال تمسح أو تغير البيانات الخاصة بالعلاقة بدون قصد .

هذا يعني أن هناك علاقة ارتباطية بين بيانات الجدولين عن طريق الحقلين في طرفي العلاقة و باختيار هذا الخيار ، فاننا نعني الآتي :

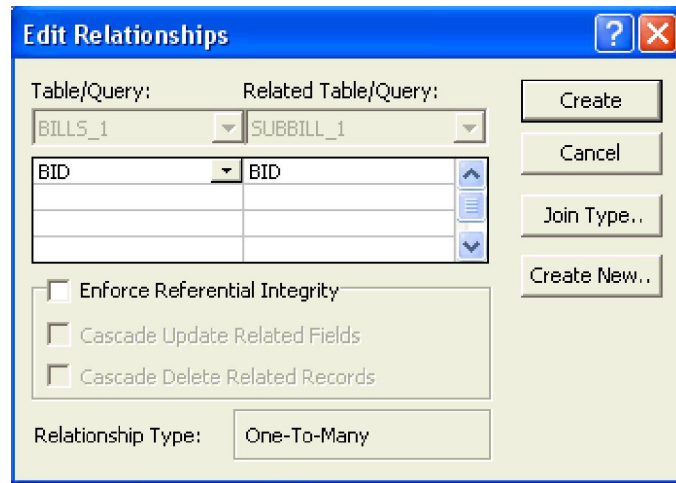
١- طرف العلاقة في الجدول الرئيسي هو حقل المفتاح أو علي الأقل حقل فريد في الجدول الرئيسي (Courses)

٢- الحقل الآخر الذي يمثل الطرف الثاني من العلاقة في جدول EMPCOURSES له نفس نوع البيانات

٣-الجدولان موجودان في نفس قاعدة البيانات

و يتم تفعيل هذا الخيار بالنقر علي العلاقة و اختيار خصائص من قائمة الماوس المختصرة

ثم اختيار Enforce referential Integrity



و باختيار هذا الخيار لا يمكنك إضافة دورة اسمها XXXX في جدول تسجيل حركات الدورات الا اذا كان هناك بالفعل دورة مسجلة بهذا الاسم في جدول الدورات ن و هذا يعتبر قيد مفيد جدا يقلل الخطأ في ادخال البيانات

و بعد اختيار Enforce referential Integrity

يمكننا الاختيار بين تطبيق الخيارات الآتية

تفعيل توالي ارتباطات التحديث للعلاقة

باختيار Cascade update related field

فبتحديث كود الدورة في جدول ادورات ، يتم التحديث لكل السجلات المناظرة في جدول حركات الدورات

تفعيل توالي ارتباطات الحذف للعلاقة

باختيار Cascade delete related field

فبحذف دورة من جدول الدورات ، يتم حذف كل الشجلات المناظرة لهذه الدورة فى جدول تسجيل حركة الدورات

و هكذا نكون أكملنا الحديث حول العلاقات و أنواعها و طرق إنشاءها

و سيكون لنا حديث آخر بإذن الله حول الأسلوب العلمي لبناء الجداول و العلاقات