المقصود بتطبيقات الخادم/العميل قواعد البيانات التى تستخدم لخدمة أكثر من مستفيد واحد داخل شبكة اتصالات. سنتعرف فى هذا الفصل على كيفية إنشاء تطبيقات متعددة المستخدمين عن طريق تقسيم قاعدة البيانات أو تحويلها إلى SQL Server 2005،

بانتهاء هذا الفصل ستتعرف على:

- مزايا استخدام قواعد بيانات "الخادم/العميل"
- طرق الترقية إلى قواعد بيانات "الخادم/العميل"
- الترقية عن طريق ربط الجداول.
- الترقية باستخدام Upsizing Wizard.

مزايا استخدام قواعد بيانات "الخادم/العميل"

تسمى قاعدة البيانات التى تستخدم داخل شبكة اتصالات لخدمة أكثر من مستفيد Client/Server database وسنقول عنها قاعدة بيانات الخادم/العميل. وتعتبر قواعد بيانات "الخادم/العميل" الحديثة الممثلة من خلال Microsoft SQL Server 2005 أكثر ثقة وثباتاً من قواعد البيانات التى تستخدم ملفات مشتركة مثل Access على سبيل المثال، فالثقة Reliability تتمثل فى الوصول السهل إلى البيانات فى زمن بسيط والقدرة على الاتساع Scalability يتمثل فى القدرة على وجود عدد أكبر من المستخدمين فى وقتٍ واحد مع عمل قاعدة البيانات بنفس الكفاءة. فمعظم قواعد البيانات المستخدمة لدى الشركات والمؤسسات الكبيرة تعمل بنظام "الخادم/العميل"، حيث يتم فى هذا النظام وضع قاعدة البيانات على الخادم الذى يكون عليه العبء الأكبر من العمليات من جهة كما يعمل على عدم زيادة التحميل على الشبكة من جهة أخرى. فعند تنفيذ استعلام تحديد يحتوى على الكود * SELECT من جهاز العميل على جهاز الخادم ويحتوى هذا الاستعلام على معايير معينة داخل الجزء WHERE، يتم إرجاع السجلات التى تتفق مع هذه المعايير فقط مما يعمل على عدم ازدحام الشبكة بالبيانات. فإذا تم استبدال * بالحقول التى ترغب فى استرجاع بياناتها، يقوم الخادم بإرجاع قيم هذه الحقول فقط دون غيرها من الحقول الأخرى الموجودة بالجدول مما يقلل من كمية البيانات التى يتم إرسالها إلى جهاز العميل وبالتالى زيادة كفاءة عمل الشبكة ككل.

وبجانب زيادة كفاءة الشبكة، تعمل قواعد بيانات "الخادم/العميل" على الاستغناء عن عمليات الضغط والإصلاح Compact/Repair المستخدمة مع قواعد بيانات Access متعددة المستخدمين بهدف الحذف الفعلى للسجلات المحذوفة من الجداول. لأنك عندما تقوم بحذف السجلات من الجداول، يقوم Access بوضع علامة الحذف على هذه السجلات إلا أنه لا يقوم بحذفها فعلياً، لذا يجب أن تقوم بضغط قاعدة البيانات بصورة دورية كى يتم حذف هذه السجلات نهائياً وبالتالى تحرير المساحة التى تشغلها هذه

السجلات. كما أن استخدام قواعد بيانات "الخادم/العميل" يعمل أيضاً على تجنب عمليات الحجب الناتجة عن حدوث خطأ ما إذا انقطع التيار عن الحاسب وكان أحد المستخدمين يقوم ببعض التعديلات على الجداول.

طرق الترقية إلى قواعد بيانات "الخادم/العميل"

قبل ظهور Access 2000، كانت عملية الربط Linking هى الطريقة الوحيدة لترقية قواعد البيانات العادية إلى قواعد بيانات تعمل بنظام "الخادم/العميل". وميزة هذه الطريقة تتمثل فى قيام محرك قاعدة البيانات الذى يعمل على جهاز العميل بتنفيذ استعلامات SQL، أى تعمل الاستعلامات الجدولية Crosstab Queries بالطريقة المعتادة كما يمكنك استخدام كائنات MSGraph داخل النماذج والتقارير دون الحاجة إلى إعادة تعريف الخاصية Record Source المصاحبة لكلٍ منها. كما يمكنك من خلال عملية الربط الحصول على ميزة استعلامات التمرير Pass-through Queries لإرسال عبارات T-SQL مباشرةً إلى خادم SQL. وعلى الرغم من ذلك، يعاب على طريقة الربط هذه أنها تفتقد كفاءة معالجة الاستعلامات على الخادم، حيث يتم معالجتها على جهاز العميل كما ذكرنا منذ قليل، وذلك لأن معالجة الاستعلامات على الخادم من أهم السمات التى تميز قواعد البيانات التى تعمل بنظام "الخادم/العميل". وفيما يلى نتعرف على الطرق المستخدمة داخل Access 2007 للانتقال من قواعد البيانات العادية إلى قواعد بيانات "الخادم/العميل" (قواعد بيانات SQL Server).

يدعم Access 2007 طرق الترقية التلقائية الثلاث التالية؛ والتى يطلق عليها Upsizing، لتحويل قواعد البيانات من تنسيق Access إلى تنسيق SQL Server:

- تقسيم قاعدة بيانات Access ذات المستخدم الواحد (Splitting) ثم تحويلها (Upsizing) ثم ربط (Linking) كائنات التطبيق بالجداول. فعلى فرض احتواء قاعدة بيانات Access ذات الامتداد accdb. على كائنات التطبيق (الاستعلامات والنماذج والتقارير وصفحات البيانات والوحدات النمطية ووحدات

الماكر وغيرها) وكائنات البيانات الممثلة فى الجداول، يمكنك تقسيم وتحويل الجداول وربط كائنات التطبيق بالجداول الموجودة على الخادم فى خطوة واحدة. وخير مثال على هذه الطريقة عملية تقسيم قاعدة البيانات **Sales.accdb** التى سنشرحها بعد قليل.

- تحويل وربط (**Upsizing and Linking**) قاعدة بيانات **Access** متعددة المستخدمين. فإذا كنت قد استخدمت مقسم قاعدة البيانات **Database Splitter** فى فصل كائنات التطبيق عن كائنات البيانات بتطبيق أمامى **Front-end** وتطبيق خلفى **Back-end** على الترتيب، يتم تحويل ملف التطبيق الأمامى ذى التنسيق **accdb.** من خلال عملية التحويل **Upsizing**، حيث لا يمكنك من خلال هذه العملية تحويل التطبيق الخلفى الذى يحتوى على الجداول.
- تحويل (**Upsizing**) تطبيق **Access** إلى مشروع بيانات (**Data Access Project**)، وفى هذه الحالة يتم نقل جداول **Access** إلى خادم **SQL** مع محاولة تحديث الاستعلامات إلى كود **T-SQL** المصاحب.

سنركز فى هذا الفصل على كيفية استخدام الطريقتين الأولى والثانية على أن نتعرف على كيفية استخدام الطريقة الثالثة فى فصلٍ آخر إن شاء الله.

الترقية عن طريق ربط الجداول

يتميز **Access** عن غيرة من التطبيقات ولغات البرمجة الأخرى فى احتواء قاعدة بياناته على ملف وحيد بالامتداد **accdb.** مما يسهل بالطبع من التعامل مع محتويات قاعدة البيانات وسهولة نسخها إلى أى حاسب آخر مثبت عليه **Access 2007**. ويحتوى هذا الملف على نوعين أساسيين من الكائنات، الأول عبارة عن كائنات التطبيق **Application Objects** ويشمل النماذج والتقارير ووحدات الماكرو والوحدات النمطية، والثانى عبارة عن كائنات البيانات **Data Objects** ويشمل الجداول والاستعلامات. فإذا أردت مشاركة قاعدة البيانات الواحدة بين أكثر من مستخدم، أى استخدام قاعدة البيانات فى

بيئة متعددة المستخدمين داخل شبكة اتصالات تحتوى على أكثر من حاسب، وبالتالى يستطيع أكثر من مستخدم العمل مع قاعدة بيانات واحدة فى نفس الوقت، يجب فى هذه الحالة فصل كائنات التطبيق (أى النماذج والتقارير وغيرها) عن كائنات البيانات (أى الجداول والاستعلامات). وعلى الرغم من سهولة استخدام هذه الطريقة، إلا أنها لا تخلو من العيوب والتى يأتى فى مقدمتها بطء استجابة التطبيق نظراً لعمليات إرسال واستقبال البيانات المتكررة من خلال الشبكة، كما أن تعديل أى كائن من كائنات قاعدة البيانات فى طور التصميم من قِبَل أحد المستخدمين يمنع بقية المستخدمين من التعامل مع هذا الكائن حتى يفرغ الأول من تعديلاته.

الحل الأمثل فى هذه الحالة هو تقسيم قاعدة البيانات إلى مكونات أمامية تسمى **Front-end** ومكونات خلفية تسمى **Back-end**، وفى حالة قواعد بيانات **Access**، تحتوى المكونات الأمامية على جميع كائنات التطبيق (أى النماذج والتقارير ووحدات الماكرو والوحدات النمطية والاستعلامات)، بينما تحتوى المكونات الخلفية على الجداول فقط.

وتحتاج تطبيقات **Access** متعددة المستخدمين عادةً إلى تثبيت ملف الكائنات الأمامية **(Front-end)** على الحاسب الخاص بكل مستخدم مع القدرة على الوصول إلى الكائنات الخلفية **(Back-end)** أى الجداول من خلال الشبكة، حيث يتم عادةً تخزين الجداول على الخادم أو على أى جهاز آخر بالشبكة، وبالتالى يمكن ربط كائن خلفى معين بأكثر من كائن أمامى.

تستخدم **Access 2007** عبارة **Upsize** أو **Upsizing** للتعبير عن الانتقال أو التحويل من التطبيقات المعدة لخدمة مستفيد واحد – وهى التى شرحناها فى الفصول المتقدمة – إلى التطبيقات المعدة لخدمة عدة مستفيدين داخل شبكة اتصالات وهى التى تستخدم مع **SQL Server**.

وتستخدم عدة طرق من أجل الانتقال من (أو تحويل) تطبيقات تخدم مستفيد واحد إلى تطبيقات تستخدم داخل شبكة اتصالات سنشرحها بعد قليل فى هذا الفصل. واحدة من هذه الطرق هى استخدام معالج التحويل **Upsizing Wizard**. وعندما تستخدم هذه

الطريقة وتختار ربط تطبيقاتك إلى SQL Server 2005 (سواءً الإصدار Standard أو Express)، ستحل جداول SQL Server محل جداول Access.

تسمى نسخة SQL Server 2005 Express Edition اختصاراً SSX أو SQL Express.

يعتمد الشرح فى هذا الفصل على أنك تفهم معنى مشاركة الملفات وأن لديك حساب مسئول عن الشبكة وتعرف كيف تدير مستخدمى الشبكة التى يعمل عليها Windows.

استخدام مقسم قواعد البيانات لإنشاء جداول مرتبطة

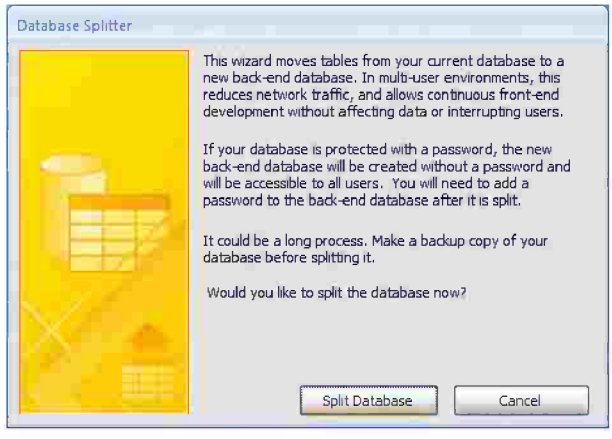
يمكنك استخدام مقسم قواعد البيانات Database Splitter لتقسيم قاعدة بيانات Access إلى جزئين، الأول يحتوى على الجداول فقط (تسمى المكونات الخلفية أو Back-end) بينما يحتوى الآخر على المكونات الأمامية (Front-end) التى تشمل الاستعلامات والنماذج والتقارير وغيرها، حيث يقوم المقسم بتقسيم قاعدة البيانات إلى قاعدتى بيانات، الأولى تحتوى على الجداول فقط، بينما تحتوى الثانية على بقية الكائنات مع ربطها بالجداول الموجودة بقاعدة البيانات الأولى.

للتعرف على كيفية أداء ذلك من خلال قاعدة البيانات Sales.accdb التى بين أيدينا فى هذا الكتاب، تابع معنا الخطوات التالية:

١. إذا كان حاسبك متصلاً بشبكةٍ ما من أجهزة الحاسب سواءً من خلال خادم Server أو من خلال محطة عمل أخرى Workstation، قم بإنشاء مجلد مشاركة على الخادم أو محطة العمل كى تتمكن من وضع قاعدة البيانات الخلفية (أى التى تحتوى على الجداول) بداخله. مع الوضع فى الاعتبار عدم اشتراط احتواء الخادم أو محطة العمل على Access 2007. أما إذا أردت تخزين قاعدة البيانات الخلفية على حاسبك، فيجب أن تنشئ مجلداً جديداً باسم مناسب وليكن BackEnd.

٢. قم بإنشاء نسخة احتياطية من قاعدة البيانات التى ترغب فى تقسيمها حتى تتمكن من الرجوع إليها واستخدامها إذا لزم الأمر. سنقوم فى هذا المثال باستخدام نسخة من قاعدة البيانات Sales.accdb مع تغيير اسمها إلى اسم آخر وليكن SalesClient.accdb.

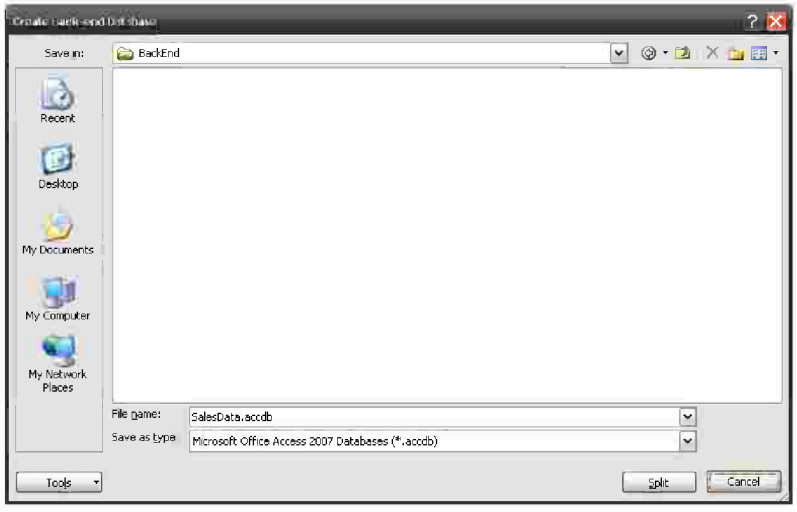
٣. قم بفتح قاعدة البيانات التى ترغب فى تقسيمها (سنستخدم فى هذا المثال قاعدة البيانات SalesClient.accdb وهى موجودة على القرص المدمج المرفق مع الكتاب) ثم نشط التبويب Database Tools وانقر الزر Access Database الموجود بالمجموعة Move Data. تظهر شاشة معالج التقسيم وهى الشاشة الوحيدة لهذا المعالج والتى تحتوى على شرح بسيط لوظيفة المعالج (انظر شكل ١-١٤).



شكل ١-١٤ الشاشة الأولى من معالج تقسيم قاعدة البيانات.

٤. انقر زر Split Database. يظهر المربع الحوارى Create Back-end Database والذى يمكنك من خلاله تحديد اسم ومكان قاعدة البيانات الخلفية (أى التى ستحتوى على الجداول)، حيث يظهر افتراضياً اسم قاعدة البيانات الأصلية مضافاً إليه الجزء be_ (انظر شكل ٢-١٤). قم بتغيير الاسم إلى أى اسم معبر آخر وليكن SalesData.accdb فى هذا المثال ولا تنس تحديد المكان

المناسب لقاعدة البيانات سواءً على الخادم أو محطة العمل أو على حاسبك. نفترض فى هذا المثال وضع قاعدة البيانات داخل المجلد **BackEnd** الـذى أنشـأناه فى الخطوة رقم ١.



شكل ٢-١٤ تحديد اسم ومكان قاعدة البيانات الخلفية.

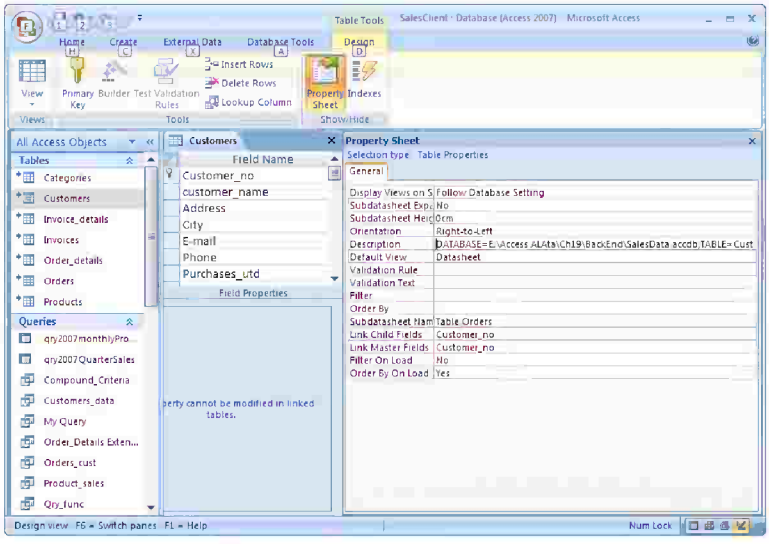
٥. انقر الزر **Split**. يقوم **Access** بتقسيم قاعدة البيانات عن طريق إنشاء قاعـدة بيانات خلفية (قاعدة البيانات **SalesData.accdb** فى هـذه الحالـة) ونقـل الجداول إليها، حيث يظهر مربع رسالة يخبرك بنجاح عملية التقسيم (انظر شـكل ٣-١٤). انقر زر **Ok** لإغلاق مربع الرسالة.وإنهاء المعالج. وبذلك يكون لـدينا قاعدتى بيانات، الأولى قاعدة البيانات الخلفية باسم **SalesData.accdb** وتحتوى على الجداول فقط، والثانيـة قاعـدة البيانـات الأماميـة (الأصـلية) باسـم **SalesClient.accdb** وتحتوى على بقية الكائنات.

شكل ٣-١٤ يخبرك **Access** بنجاح عملية التقسيم.

٦. تظهر الجداول داخل قاعدة البيانات الأمامية مصحوبة بالرمز دلالةً على أنها موجودة فى قاعدة بيانات أخرى وليس فى قاعدة البيانات الحالية (انظر شكل ٤-١٤). قم بفتح كل جدول على حده وتأكد من عمل الجدول بشكل سليم كما لو كان جزءاً من قاعدة البيانات الحالية.

شكل ٤-١٤ تظهر الجداول المرتبطة مصحوبة بسهم صغير فى قاعدة البيانات الأمامية.

٧. قم بفتح أحد الجداول فى نمط التصميم. ستحصل على رسالة تخبرك بعدم القـــدرة على تعديل بعض الخصائص من خلال نمط التصميم. انقر زر Yes لإغلاق الرسالة ثم انقر زر خصائص الجدول Property Sheet لإظهار نافذة الخصائص ولاحظ وجود اسم ومسار قاعدة البيانات الخلفية التى تحتوى الجدول المفتوح أمام الخاصية Description (انظر شكل ٥-١٤).



شكل ٥-١٤ يظهر اسم ومسار قاعدة البيانات الخلفية بالخاصية **Description**.

٨. قم بتشغيل بعض الاستعلامات وفتح بعض النماذج والتقارير وتأكد من عملها بطريقة سليمة كما لو أن الجداول جزءاً لا يتجزأ من قاعدة البيانات الحالية.

التحكم فى أذونات قاعدة البيانات الخلفية

الخطوة التالية بعد إنشاء قاعدة البيانات الخلفية هى تعيين أذونات المشاركة المختلفة الخاصة بقاعدة البيانات. فلو أن لجميع مستخدمى الشبكة الصلاحيات الكاملة فى التعامل مع قاعدة البيانات، لكان من السهولة حذفها أو نقلها أو نسخها أو تعديلها من قِبَل أى مستخدم من مستخدمى الشبكة. لذا يجب إعادة تخصيص الأذونات الخاصة بالملف والمجلد والمشاركة، حيث عادةً ما ترث قاعدة البيانات والمجلد الموجودة بداخله الأذونات الموجودة بالمجلد أو المشغل الأبوى. وعادةً ما يحتاج جميع مستخدمى قاعدة البيانات الأمامية إلى أذونات القراءة للمشاركة وأذونات القراءة والكتابة والإنشاء والحذف للمجلد، وذلك حتى يتمكن المستخدم الأول من إنشاء الملف ذى الامتداد **laccdb**. (وهو الملف **SalesData.laccdb** فى المثال الذى بين يدينا) بينما يتمكن المستخدم الأخير من حذفه،

حيث يتم تلقائياً إنشاء هذا الملف مع أول مستخدم يدخل إلى قاعدة البيانات وحذفه مع آخر مستخدم يقوم بإغلاقها.

يحتوى الملف laccdb. على اسم الحاسب واسم الأمان المستخدم داخل Access (وهو عادةً الاسم Admin ما لم تقم بتغييره) لكل شخص متصل حالياً بقاعدة البيانات الخلفية، حيث يحتوى هذا الملف على مجموعة من السجلات بحد أقصى ٢٥٥ وهو الحد الأقصى لعدد مستخدمى قاعدة بيانات Access.

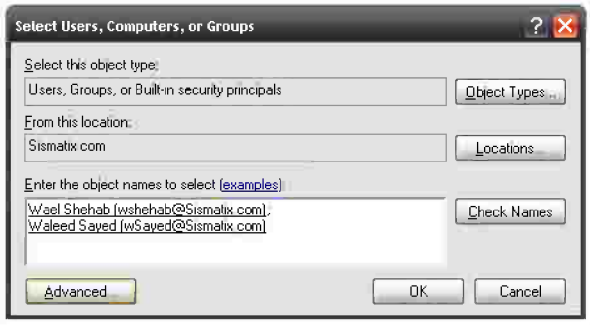
ويمكنك تخصيص الأذونات المختلفة إلى مجموعات أمان أو مستخدمين. للتعرف على كيفية أداء ذلك، تابع معنا الخطوات الآتية:

١. قم بالدخول إلى الخادم أو الحاسب الذى يحتوى على قاعدة البيانات التى ترغب فى مشاركتها بحساب عضو فى مجموعة المديرين حتى يكون لك صلاحيات التعديل المختلفة.

٢. انتقل إلى نافذة مستكشف Windows ثم انقر مجلد المشاركة (المجلد BackEnd فى هذه الحالة) بزر الفأرة الأيمن ثم اختر Properties من القائمة الموضعية الناتجة لإظهار مربع الخصائص الخاص بالمجلد. نشط التبويب Sharing ثم انقر الزر Permissions لإظهار المربع الحوارى الخاص بأذونات مجلد المشاركة.

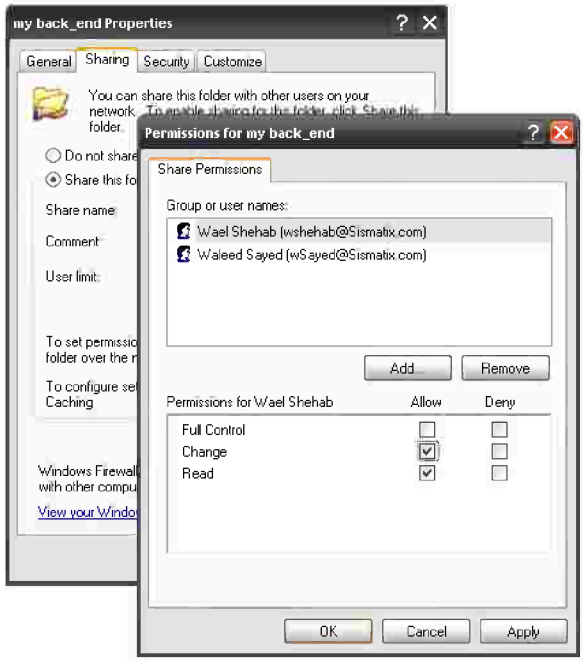
٣. فى حالة وجود Everyone بمربع السرد Group or user name، قم باختياره ثم انقر زر Remove حتى لا يتمكن أحد المستخدمين من الوصول للمجلد إلا من نقوم بإضافته فى الخطوات التالية.

٤. انقر زر Add، يظهر المربع الحوارى Select Users, Computers or Groups. قم بإدخال أسماء المستخدمين (أو المجموعات) الذين ترغب فى إضافتهم إلى قائمة من لهم حق الوصول إلى المجلد مع فصلهم باستخدام علامة الفاصلة المنقوطة وأخيراً انقر زر Check Names للتأكد من صحة هذه الأسماء، حيث يظهر خط أسفل كل اسم صحيح قمت بإدخاله (انظر شكل ٦-١٤).



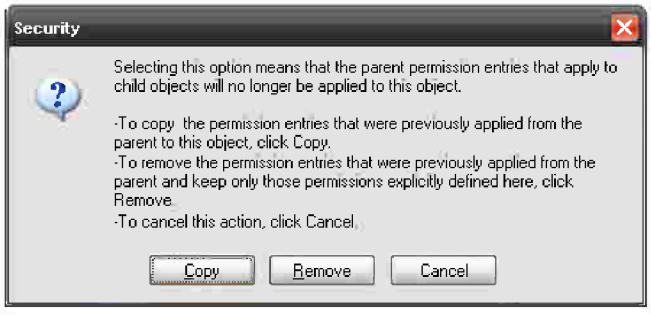
شكل ٦-١٤ يمكنك من خلال هذا المربع تعيين أسماء من يستطيع الوصول إلى المجلد.

٥. انقر زر **Ok** لإغلاق المربع الحوارى وإضافة الأسماء (أو المجموعات) التى قمت بتحديدها إلى قائمة المستفيدين أو المجموعات. اختر كل مستخدم (مجموعة) على حده ثم قم بتنشيط مربعى الاختيار **Change** و**Read** بالعمود **Allow** لتمكين المستخدمين من قراءة وتعديل مجلد المشاركة (انظر شكل ٧-١٤). وقد قمنا بتنشيط الإذن **Change** حتى يتمكن المستخدمون من إنشاء وحذف الملف **ldb.** كما ذكرنا منذ قليل والذى يتم إنشاؤه مع أول مستخدم وحذفه مع آخر مستخدم. انقر زر **Ok** لإغلاق المربع الحوارى ... **Permissions** والعودة مرةً أخرى إلى مربع الخصائص.



شكل ٧-١٤ تمكين المستخدمين من القراءة والتعديل لمجلد المشاركة.

٦. نشط التبويب **Security** ثم انقر زر **Advanced**، يظهر المربع الحوارى **Advanced Security Settings**. قم بتعطيل مربع الاختيار **Allow Inherited Permissions from the parent to This object and All Child Objects** حتى يتمكن بقية المستخدمين من الوصول إلى ملف قاعدة البيانات الموجود داخل مجلد المشاركة، وحينئذٍ يظهر مربع رسالة بعنوان **Security** يسألك إذا كنت ترغب فى نسخ الأذونات الموروثة من المستخدمين (انظر شكل ٨-١٤). انقر زر **Copy** ثم انقر زر **Ok** لإغلاق المربع الحوارى. من صفحة **Security** اختر **Users** ثم انقر زر **Remove**.

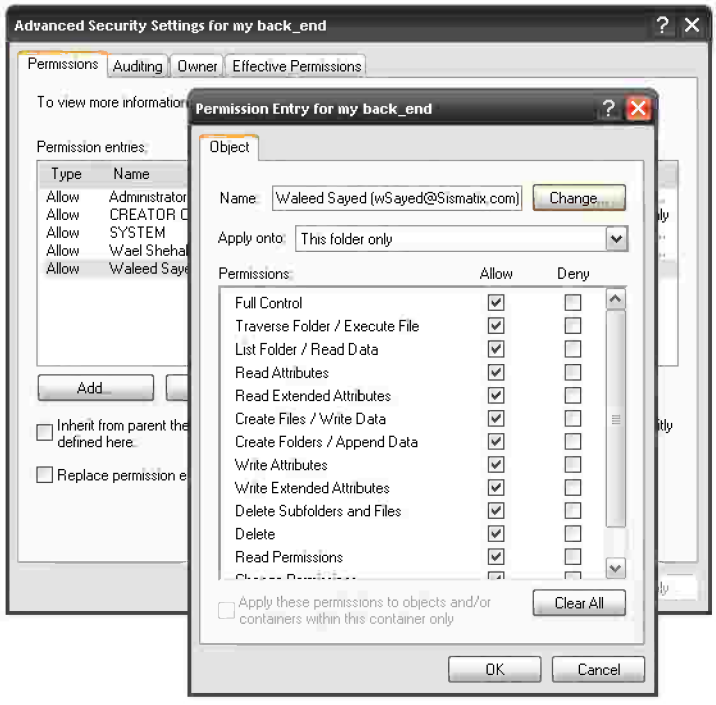


شكل ٨-١٤ نسخ الأذونات حتى نتمكن من حذف بقية المستخدمين.

٧. قم بإضافة المستخدمين (أو المجموعات) إلى قائمة **Permission List** وقم بتخصيص كل مستخدم أو مجموعة بإذن عرض محتويات المجلد وقراءتها بتنشيط مربع الاختيار **Read & Execute** بالعمود **Allow**. قم بتخصيص الأذن **Write Permission** للمستخدمين الذين ترغب فى تمكينهم من الكتابة إلى ملف قاعدة البيانات الموجود بمجلد المشاركة.

٨. انقر زر **Advanced** لإظهار المربع الحوارى **Advanced Security Settings**. اختر المستخدم الذى ترغب فى إعطائه صلاحيات القراءة فقط ثم انقر زر **Edit** لإظهار المربع الحوارى **Permission Entry** ثم قم بتنشيط مربع الاختيار **Create Files/Write Data** ومربع الاختيار **Delete Subfolders and Files** وذلك لكى يتمكن المستخدمون من فتح وإغلاق الملف **laccdb.** (انظر شكل ٩-١٤).

٩. انقر زر **Ok** عدة مرات لإغلاق المربعات الحوارية المفتوحة وحفظ التعديلات التى قمت بها.



شكل ٩-١٤ تمكين المستخدمين من تعديل الملف **laccdb.** حتى وإن لم تكن لهم صلاحيات الكتابة.

الترقية باستخدام معالج التحويل Upsizing Wizard

فى حالة إنشاء تطبيق **Access** مكون من ملف واحد فقط يحتوى على الامتداد **accdb.** وترغب فى جعله متاحاً للزملاء المثبت على حواسبهم **Access 2007**، يمكنك فى هـــذه الحالة استخدام معالج التحويل **Upsizing Wizard**. ويجب فى هذه الحالة بالطبع تثبيت **SSX** أو أى من إصدارات **SQL Server** الأخرى كما يجب التأكد من عـــدم تشـــفير التطبيق الذى ترغب فى تحويله.

تهيئة الجداول لعملية التحويل للحصول على نتائج صحيحة

يحتوى معالج التحويل على العديد من القيود التى تتطلب بقدر الإمكان تهيئة الجداول قبل إجراء عملية التحويل حتى نحصل على نتيجة صحيحة. ففى بعض الحالات لا تتم عمليـــة التحويل بشكل صحيح كما أن التقرير الناتج ربما لا يوضح سبب ذلك، لذا هناك بعـــض

التعديلات التى يجب أن تتم على خصائص الجداول للتأكد من عدم وجـــود مشـــاكل فى عملية التحويل والحصول على النتيجة المرجوة، وهذه التعديلات هى:

- *تعبيرات القيم الافتراضية* Default Value *وقواعد التحقق مـــن الصـــحة* Validation rules

لا يستطيع T-SQL احتواء العديد من تعبيرات Access أو تعبيرات VBA التى تقوم بإنشاء قيم افتراضية أو قيم التحقق من الصحة، ففى مثل هذه الحالات ربما لا يستطيع معالج التحويل إنشاء الجدول من الأساس. وفى هذه الحالة يجـــب حـــذف التعبيرات الغير مرغوبة وتشغيل المعالج مرةً أخرى لربط الجداول التى لم يتم إنشاؤها فى الجولة الأولى فقط، وبعد ذلك يجب كتابة هذه التعبيرات لكى تتوافق مـــع T-SQL.

- *الجداول المخفية* Hidden Tables

إذا قمت بإخفاء أيٍ من الجداول الموجودة بقاعدة البيانات والتى ترغب فى تحويلها، فسيتجاهل المعالج هذه الجداول تماماً أثناء عملية التحويل.

- *الحقول المضافة من خلال النسخ المطابقة* Replication Fields

إذا قمت بتنفيذ عملية النسخة المطابقة Replication، (وهى عملية تيم بها نسخ جداول قاعدة البيانات فى أكثر من مكان)، يجب أن تقوم بحذف جميع حقول النظام التى تم إضافتها للجدول أثناء العملية قبل أن تقوم بعملية التحويل، حيث لا يـــتم تحويل أى جدول يحتوى على مثل هذه الحقول.

- *الجداول التى لا تحتوى على فهارس فريدة*

يمكنك داخل Access تحديث الجداول التى لا تحتوى على فهارس بقيم فريـــدة، أما داخل SQL Server فيجب وجود فهرس فريد داخل الجدول لكى يتم تحديثه باستمرار. لذا تأكد من احتواء جميع الجداول على فهارس فريدة. فإذا لم تتمكن من إنشاء الفهرس بأيٍ من الحقول الموجودة بالجدول، قم بإضافة حقل مـــن النـــوع AutoNumber إلى هذا الجدول (حيث يتم تحويل النـــوع AutoNumber إلى

النوع integer مع تخصيص القيمة Yes للخاصية Identity عند تحويل الجدول إلى SQL Server). وعادةً يكون الفهرس الفريد هو حقل المفتاح الأساسي الموجود بالجدول.

- *الحقول المرتبطة التى لا تحتوى على نفس الحجم*

يمكنك من خلال Access إنشاء العلاقات بين الحقول النصية مختلفة الحجم، بينما لا يسمح SQL Server بذلك، وفى هذه الحالة يتم بالفعل تحويل الجداول، إلا أن المعالج يتجاهل العلاقة الموجودة بينها، لذا تأكد فى كل علاقة من تساوى حجم حقل المفتاح الأساسى وحقل المفتاح الأجنبى المرتبط به، حيث يمكنك استخدام الحجم الأكبر فى كلا الحقلين حتى لا يتم فقد أيٍ من البيانات الموجودة بهما.

- *الجداول كبيرة الحجم*

أثناء إضافة البيانات إلى الجدول بعد تحويله، يقوم SQL Server بإضافة بيانات إلى ملف تسجيل الإجراءات Transaction log file المصاحب لقاعدة البيانات. فإذا كان بقاعدة بيانات Access جدول كبير الحجم وكانت المساحة التخزينة الموجود عليها ملف التسجيل وقاعدة البيانات بسيطة، فربما يتعدى حجم الجدول وملف التسجيل المساحة المتاحة، لذا يجب أن يكون حجم القرص الهدف المستخدم فى تخزين قاعدة البيانات بعد تحويلها أكبر ثلاث مرات من حجم قاعدة بيانات Access الأصلية.

تشغيل معالج التحويل

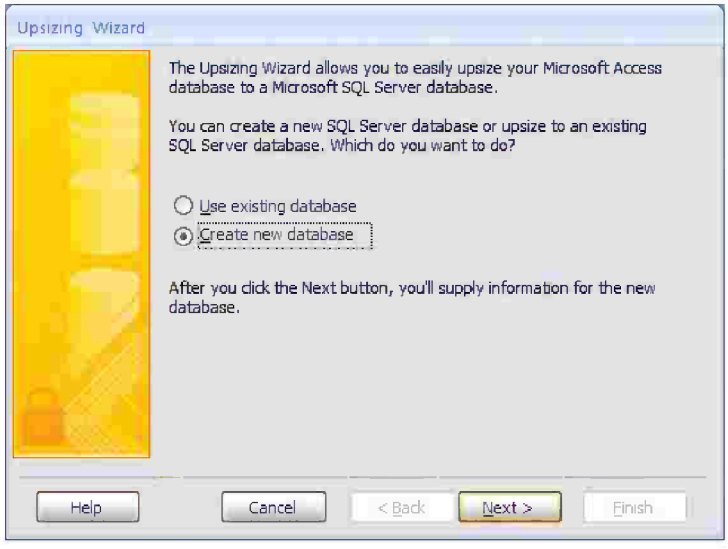
بعد أن قمنا بتهيئة جداول قاعدة البيانات لعملية التحويل، يمكنك الآن البدء فى تحويل وربط قاعدة البيانات من خلال معالج التحويل Upsizing Wizard. وفى هذا المثال سنقوم بتحويل قاعدة البيانات Sales.accdb الموجودة على القرص المدمج المرفق بالكتاب. تابع معنا الخطوات الآتية:

١. قم بإنشاء نسخة احتياطية من قاعدة البيانات وتأكد من عمل هذه النسخة بشكل

صحيح حتى تتمكن من استخدامها فى حالة حدوث خلل فى النسخة الأصلية لا قدر الله.

٢. قم بفتح قاعدة البيانات Sales.accdb داخل نافذة Access ولا تقم بفتح أى كائنات بداخلها.

٣. نشط التبويب Database Tools ومن مجموعة Move Data انقر الزر SQL Server. تظهر الشاشة الافتتاحية لمعالج التحويل Upsizing Wizard (انظر شكل ١٠-١٤) .

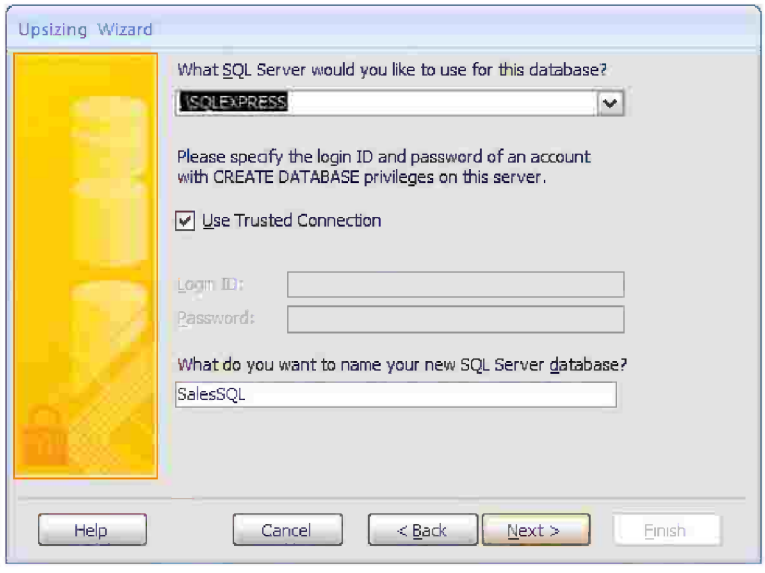


شكل ١٠-١٤ الشاشة الافتتاحية لمعالج التحويل.

٤. تأكد من تنشيط زر الاختيار Create new database ثم انقر زر Next، تظهر الشاشة الثانية من المعالج تسألك عن اسم خادم SQL الذى تريد استخدامه مع قاعدة البيانات هذه. افتح قائمة ... Select SQL Server ثم اختر اسم الخادم الذى ترغب فى تخزين قاعدة بيانات SQL Server عليه. فى هذا المثال اخترنا .SQLExpress\. لأننا نستخدم SQL Server 2005 Express Edition أو SSX، كما يمكنك استخدام (local) أو إدخال اسم حاسبك أو اسم خادم آخر إذا

أردت تخزين قاعدة البيانات الناتجة على هذا الخادم (انظر شكل ١١-١٤).

٥. تأكد من تنشيط مربع الاختيار **Use Trusted Connection** للدخول إلى الخادم بصلاحيات **Windows** دون الحاجة إلى تعيين اسم مستخدم أو كلمة مرور. ابق على الاسم الافتراضي لقاعدة البيانات الجديدة كما هو وإلا قم بتغييره إن أحببت من خلال مربع النص **What do you want to name our new SQL Server database**، حيث يقوم المعالج باستخدام نفس اسم قاعدة البيانات الأصلية مضافاً إليها الخاتمة **SQL**. ففي هذا المثال نقوم بتحويل قاعدة البيانات **Sales.accdb**، لذا يقوم المعالج بتسمية قاعدة بيانات **SQL Server** بالاسم **SalesSQL** (انظر شكل ١١-١٤).

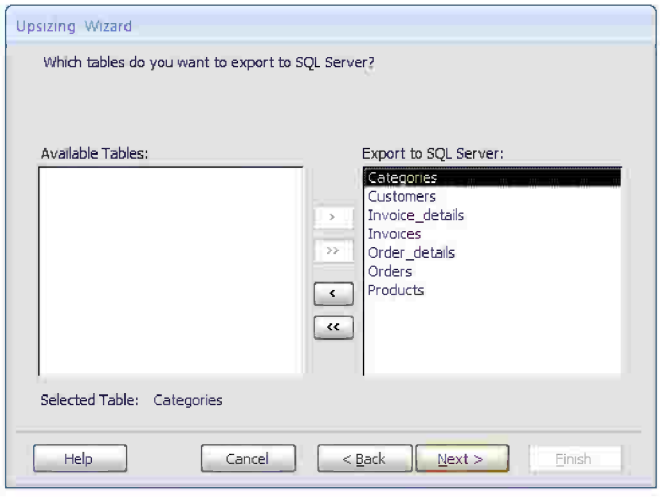


شكل ١١-١٤ تعيين اسم الخادم واسم قاعدة البيانات.

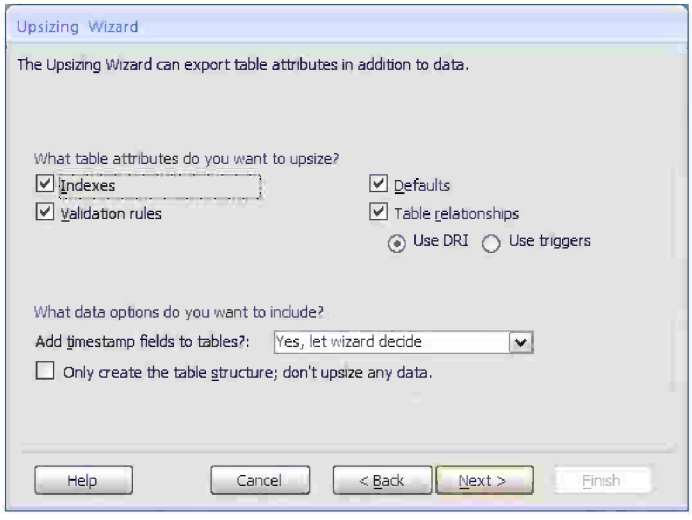
٦. انقر زر **Next**، تظهر الشاشة الثالثة من المعالج والتي يتم فيها تحديد الجداول التي سيتم إدراجها إلى قاعدة البيانات الجديدة. انقر زر >> لنقل جميع الجداول (انظر شكل ١٢-١٤).

٧. انقر زر **Next**، تظهر الشاشة الرابعة والتي يسألك المعالج فيها عما إذا كنت تريد

تصدير السمات المختلفة مع الجداول أم لا .ابق على الخيارات الافتراضية كما هى (انظر شكل ١٣-١٤).



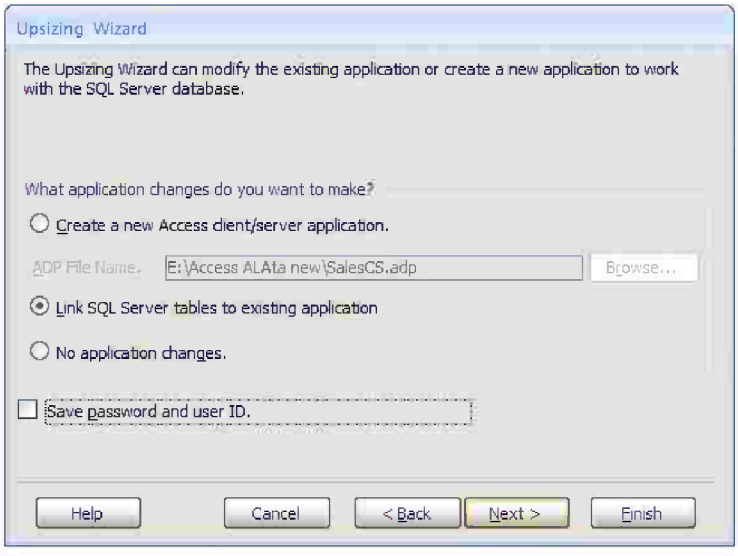
شكل ١٢-١٤ تعيين الجداول المطلوب إضافتها إلى قاعدة البيانات الجديدة.



شكل ١٣-١٤ تعيين السمات المطلوب نقلها مع الجداول.

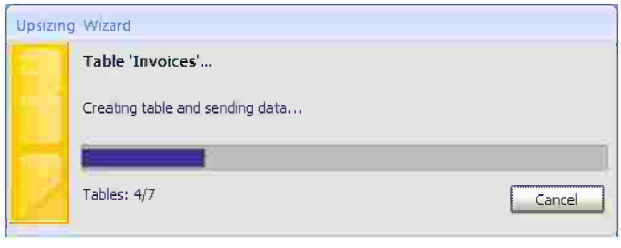
٨. انقر زر **Next**، تظهر الشاشة الخامسة والتى يسألك المعالج فيها إذا كنت ترغب فى إنشاء مشروع بيانات جديد أم استخدام التطبيق الحالى (انظر شكل ١٤-١٤).

نشط زر الاختيار **Link SQL Server tables to existing application** لربط الجداول بالتطبيق الحالى. أما زر الاختيار الأول فيتسبب فى إنشاء مشروع بيانات جديد بينما فى حالة تنشيط زر الاختيار الثالث يجب أن تقوم بربط الجداول بقاعدة البيانات الأمامية بنفسك.



شكل ١٤-١٤ تعيين التطبيق المصاحب لقاعدة البيانات الجديدة.

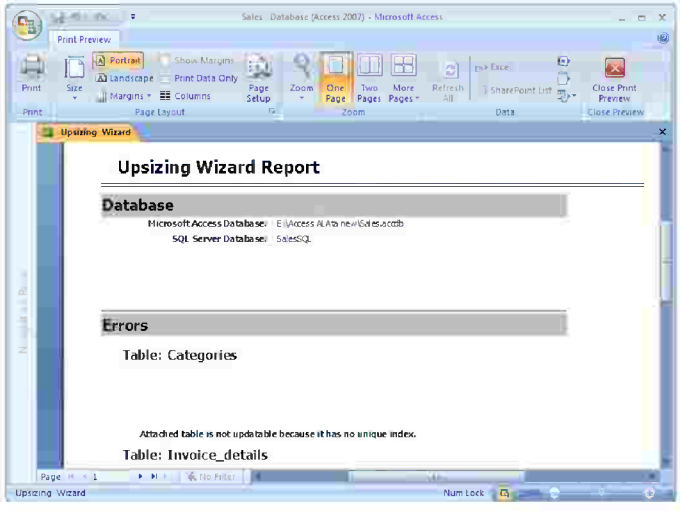
٩. انقر زر **Next**، تظهر الشاشة الأخيرة من المعالج والتى تستحثك على نقر زر **Finish** لبدء المعالج. انقر زر **Finish**، يبدأ المعالج فى العمل ويظهر مربع حوارى صغير يحدد لك نسبة الإنجاز (انظر شكل ١٥-١٤).



شكل ١٥-١٤ تحديد نسبة إنجاز عملية التحويل.

١٠. بانتهاء المعالج من عملية التحويل، يظهر تقرير عن عملية التحويل والأخطاء التى

صادفت المعالج أثناء العملية إن وجدت (انظر شكل ١٦-١٤).



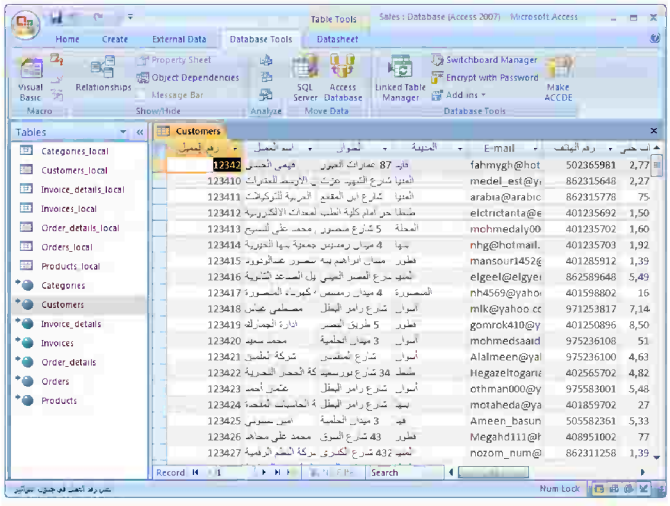
شكل ١٦-١٤ تقرير عن عملية التحويل.

١١. قم بحفظ التقرير إن أحببت أو طباعته ثم انقر زر **Close** لإغلاقه، حيث لا يقوم المعالج بحفظ التقرير من تلقاء نفسه.

اختبار عملية التحويل وربط الملفات

بالنظر إلى نافذة التطبيق **Sales.accdb** بعد تحويل الجداول من تنسيق **Access** إلى تنسيق **SQL Server** ووضع الجداول داخل قاعدة بيانات على الخادم باسم **SalesSQL**، تلاحظ إعادة تسمية المعالج لأسماء جداول **Access** بإضافة **"_local"** إلى اسم الجدول وكذلك إضافة رموز ارتباط إلى جداول **SQL Server** (انظر شكل ١٧-١٤). وإذا قمت بتمرير مؤشر الفأرة فوق أحد الجداول المرتبطة (جداول **SQL Server**) يظهر تلميح صغير مكون من سطر واحد يوضح سلسلة اتصال **ODBC** الخاصة بقاعدة البيانات.

الحروف **ODBC** اختصار للعبارة **Microsoft Open Database Connectivity API** وهى تستخدم فى الغالب للاتصال بقواعد بيانات الخادم/العميل مثل **Microsoft SQL Server** أو **Oracle**



شكل ١٧-١٤ لوحة التنقل وبها جداول SQL Server وجداول Access.

بعد أن تتأكد من احتواء جميع جداول الخادم المطلوبة على رمز الارتباط، تابع معنا الخطوات الآتية للتأكد من عمل الجداول بشكل سليم مع التطبيق الأمامى الحالى:

١. افتح النماذج والتقارير واحداً تلو الآخر للتأكد من عملها جميعاً بشكل سليم.

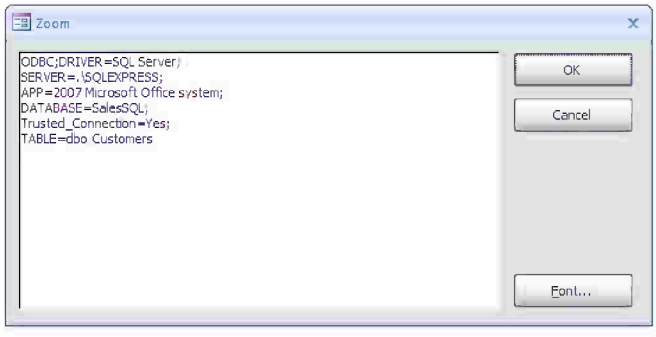
٢. من خلال النماذج أو الجداول فى طريقة عرض البيانات، انتقل إلى سجل جديد وتأكد من وجود القيم الافتراضية والتنسيقات وأقنعة الإدخال ومن عمل قواعد التحقق من الصحة بطريقة سليمة.

٣. فى حالة احتواء أى من الجداول على حقول بحث، تأكد من عمل هذه الحقول بشكل صحيح.

٤. قم بفتح الجداول داخل طريقة عرض التصميم ثم انقر زر Yes رداً على مربع الرسالة الذى يخبرك بعدم القدرة على تعديل بعض خصائص الجدول واختبر نوع بيانات حقول الارتباط التشعبى Hyperlink التى يتم تحويلها تلقائياً إلى حقول مذكرة Memo.

٥. انقر نافذة تصميم الجدول بزر الفأرة الأيمن ثم اختر Properties من القائمة الموضعية الناتجة لإظهار لوحة خصائص الجدول Property Sheet. انقر داخل

مربع النص **Description** ثم اضغط مفتاحى **Shift+F2** لإظهار المربع الحوارى **Zoom** وبه محتويات مربع النص الذى يحتوى بدوره على سلسلة الاتصال الكاملة الخاصة بالجدول (انظر شكل ١٨-١٤).



شكل ١٨-١٤ المربع الحوارى **Zoom** وبه سلسلة الاتصال الخاصة بالجدول **Customers**.

٦. قم بتنفيذ كل استعلام تحديد وتأكد من عدم تعديل المعالج لمحتوياتها ومن عملها بطريقة صحيحة.

وبمجرد الانتهاء من اختبار عملية التحويل والتأكد من عمل جميع الكائنات بطريقة صحيحة، يمكنك حذف الجداول المحلية التى تحتوى على الخاتمة **_local** طالما أنك قمت بإنشاء نسخة احتياطية من قاعدة البيانات قبل تحويلها، وإلا قم بتصدير هذه الجداول إلى قاعدة بيانات جديدة ثم قم بحذفها من قاعدة البيانات الحالية لعدم الحاجة إليها.

تحويل تطبيق يحتوى على جداول مرتبطة

نعيد القول أن كلمة تحويل هنا هى المرادف الذى اخترناه لكلمة **Upsizing** التى تستخدمها **Access** لنفس المعنى. ولا تختلف طريقة تحويل تطبيق يحتوى على جداول مرتبطة من خلال معالج التحويل **Upsizing Wizard** عن طريقة تحويل التطبيقات ذات الملف الواحد التى تعرفنا عليها منذ قليل، حيث يتم تشغيل المعالج من خلال التطبيق الأمامى **Front-end** الذى يحتوى على كائنات التطبيق، حيث يقوم المعالج تلقائياً بالاتصال بالتطبيق الخلفى **Back-end** المصاحب الذى يحتوى بدوره على كائنات البيانات (الجداول).

سنقوم فيما يلى بتحويل قاعدة البيانات الأمامية SalesClient.accdb وقاعدتها الخلفية المصاحبة SalesData.accdb وهما اللتان قمنا بإنشائهما فى بداية هذا الفصل. تابع معنا الخطوات الآتية:

١. قم بإنشاء نسخة احتياطية من قاعدتى البيانات SalesClient.accdb وSalesData.accdb.

٢. قم بفتح قاعدة البيانات الأمامية SalesClient.accdb ثم نشط التبويب Database Tools ثم انقر الزر SQL Server بالمجموعة Move Data، وحينئذٍ تظهر الشاشة الافتتاحية لمعالج التحويل Upsizing Wizard (راجع شكل ١٠-١٤).

٣. تأكد من تنشيط زر الاختيار Create new database ثم انقر زر Next، تظهر الشاشة الثانية من المعالج. أدخل اسم الخادم الذى ترغب فى تخزين قاعدة بيانات SQL Server عليه وقم بتغيير اسم قاعدة البيانات إلى SalesDataSQL.

٤. انقر زر Next للانتقال إلى الشاشة الثالثة. اختر جميع الجداول ثم انقر زر Next للانتقال إلى الشاشة الرابعة وابق على الخيارات الافتراضية كما هى.

٥. انقر زر Next للانتقال إلى الشاشة الخامسة وتأكد من تنشيط زر الاختيار Link SQL Server tables to existing application وتأكد أيضاً من تعطيل مربع الاختيار Save password and user ID.

٦. انقر زر Next ثم زر Finish لإنهاء المعالج والبدء فى عملية التحويل.

٧. تأكد من عمل الكائنات المحولة بطريقة صحيحة بمجموعة الاختبارات التى ذكرناها منذ قليل.