We Translate
Business Processes

from the Mind
to the Computer
to the Bottom Line.

BUSINESS & COMPUTERS, Inc.
13839 Mur-Len Rd, Suite M
OLATHE, KANSAS 66062

Phone: (913) 764-2311
Fax: 764 7515
larryg@kcnet.com

Access 2000 (mdb) front-end To SQL 7/2000 backend

A note – the below is my humble opinion – with testing – If you use my ideas please test them and if you have problems or learn more let me know.

The following are some of the steps needed to move an Access 2000 data to SQL Server 7/2000. This is not a simple trick. There will be a learning curve at least as hard as moving from macros to code. It is my opinion that if your bosses or customers will not give you enough time to overcome the learning curve, you are making a mistake if you go forward with the transition.

The reasons I wrote this document, is that after reading over 10 books that dealt with Access and SQL Server, I came to the conclusion that none of them dealt with working with an Access 2000 mdb and a SQL Server backend very well. There are a couple of good books that dealt with developing in an Access Project (adp) however it is my opinion that this is not the way you want to move when you are moving a legacy application, and my not be the way to go if you are starting an application from scratch.

Note: Client Server is a generic term. The two most popular brands of client server are SQL Server and Oracle Server.

#1 - Why you *should not* move to a Client Server Backend.

* If your database design is such that you bring back all the records in your tables each time the user opens your forms.

* Your main tables are under 50,000 records.

* You are using macros and do not want to move to code.

* You do not want to take the time or you do not have the time to learn SQL Server, ADO, and how to program Access to work with SQL Server data.

#2 - Why you may want to move to Client Server Backend and what are the options.

Why

* Have you ever had the backend of your Access application get corrupted because someone turn off their computer without shutting down Access?

* Has someone opened your mdb file with Word, Excel, Notepad, etc., and then saved it. So you know how well you backup strategy works. (See kb Q243895 "Database Password Appears Even Though it Was Never Set")

* Data is very important. If you see a need to protect your data moving to Client Server will do that.

* When you have an Access application on the local machine and the Data on the network, and have 100,000 records in a table, the following is what happens when you query on the table to get back 10 records.

In Access: The application sends a statement to the server and the server sends back 100,000 keys if you are querying on an indexed field, or 100,000 copies of each record if querying on a non indexed field. Your machine then decides the 10 records it wants and throws away the rest of the records.

A Well Programmed Access Application hooked to a Well Programmed Client Server Backend : The application sends a statement to the Client Server. The server processes the query and sends back the 10 records

What are the Options for the Front End

* Access, Visual Basic, and a 1,000 other programming languages. Why Access? If you pick Access you are in the minority and you need to be aware of it. VB is the most popular language used against Client Server. In my opinion Access is a better solution because of a couple of things. You can do anything with Access against Client Server that VB can do, but you can do it faster. Ten years ago we talked about Rapid Development. Access is that program. If you know what you are doing, you can develop your applications faster in Access and have them run just as fast as VB. If your customers are like mine, you do not write applications that sit for 10 years, you write applications that expand on a monthly basis. Businesses that succeed are constantly changing to stay competitive in their growing market place. You need a programming language that will allow you to make changes quickly in order to keep pace with your customers needs.

  In addition Access will give the intermediate user the ability to develop ad hoc queries and reports against client server. And most people agree that Access has one of the best report writers on the market.

* Why Access 2000? Access 2000 has some tools that work extremely well with SQL Server 7/2000. In addition, I have heard of some problems with ADO 2.5 with Access 97. You might want to check this out.

* If you are going to be using SQL 2000 with Access 2000 you need to read "Access 2000 and SQL Server Readiness Update" at http://office.microsoft.com/2000/downloaddetails/accsql.htm . There are some problems that can be solved.

Do you want to use an Access database (mdb) or an Access Project.

If you are moving from a legacy application, it would appear that an mdb is the only option. I need local tables for many purposes, I also love the ability to use local queries (I am still not real proficient in writing SQL for complicated queries, so it's nice to build a query by using Access grid and then changing the SQL for t-SQL.). Because of these facts, I use MDB, however I have an ADP hooked to SQL 7 so I can edit tables, views and stored procedures on the SQL database by just loading the Access Project.

Also some of my customers still want to build ad hock queries in an mdb to get at their data.

Figure 2-1

Differences between Access Database and Project - Access 2000

Objects	Access Database (mdb)	Access Project (adp)
Tables	Store data inside a mdb - Retrieve data using Jet. Link Tables from a mdb or SQL Server.	There are no tables in an Access project. You can see the tables that are in SQL Server (Which is where your data is stored). Inside the project, you can see the data in the table, and manage the tables.
Queries	This is where the power is. We can get data from one table or multiple tables (Select Queries), get cross tab data, Update data, Append data, and make a table with data. We can delete data in tables. Also Pass through queries.	Not Available
Views	Can be linked to an mdb. Have a view in SQL Server, link it to an MDB, and use it like a table.	Views are similar to select queries, they are stored in SQL Server, not in your Access Project. You can create, change, and look at data in views from within an Access Project.
Database Diagrams	Not Available	Database Diagrams are similar to Access table relationships, inside SQL Server, however can be managed in an Access Project.
Stored Procedures	Can be used in an mdb, through ADO code, or a pass through query.	See Stored Procedures later in this article. They are a very powerful part of SQL Server, but can be managed from an Access Project.
Forms	Stored inside mdb	Stored inside adp
Reports	Stored inside mdb	Stored inside adp
Pages	Data access pages are a special type of Web page designed for viewing and working with data	Same as MDB
Macros	Stored inside mdb	Stored inside adp
Modules	Stored inside mdb	Stored inside adp

What are the Options for Client Server Back Ends

* SQL Server (multiple versions), Oracle, and 20 other Client Servers.

* I have picked SQL Server 7 because it has some common tools that work well and are integrated with Access 2000. Most of what we discuss here will work with SQL 2000, and quite honestly if I had my choice I would work with SQL 2000. It works well and has some new features that make it easier to program.

* Oracle and other Client Servers will work, however I chose to work with SQL Server.

#3 - You Need to Learn SQL Server

* There are Database Administrators (DBA) who's entire job is managing SQL Server Databases. This is not programming to show data to the user, this is taking care of the SQL Server itself. When SQL Server 7.0 arrived, many of the administration duties were automated. So you can get by without a DBA if you learn the basics of SQL Server and have some contacts if something unexpected happens. (You need to be attending the HUG SQL Server SIG.)

* You need to have a SQL Server available to you. This might seem obvious, but I just wanted to point out that without SQL Server, it would be like trying to learn Access without a copy of Access. You can do your development with the Microsoft Data Engine (MSDE) (which is the same as SQL Server for development purposes) and an Access Project to manage it, however you will not have many of the tools that make the job easier. We will discuss managing SQL Server with an Access Project and the tools available in SQL Server later.

  See "Microsoft Access 2000 Data Engine Options" and "Creating and Deploying Microsoft Access Solutions with the Microsoft Data Engine (MSDE)" white papers at

  http://msdn.microsoft.com/library/techart/acsqlres.htm#acsqlres_mde

* Go buy a book about SQL Server that becomes your bible. I recommend "Teach Yourself Microsoft SQL Server 7.0 in 21 Days" by Richard Waymire and Rick Sawtell (They also have a book available for SQL Server 2000). It has 21 chapters that takes you from installing the server to the point where you have a good understanding of how it works and how to program in SQL Server. You will not be an expert at the end of the 21 days, but you will know you way around the server.

  Ok, I know you are going to accuse me of being crazy here, but what I am recommending is to set aside 2 to 4 hours a day for 21 future days where you can sit in front of SQL server and don't stop until you understand the chapter.

  Richard and Rick take you through doing all the processes through scripts/T-SQL code and through the Enterprise Manager. Skim the part where they show you how to do processes through T-SQL, what you need to learn is how to do the processes through the Enterprise Manager. The people who have been dealing with SQL Server for the last 5 years will prefer to manage through T-SQL, because that is how they learned to manage the Server. If you like managing your databases through GUI interface, and prefer to accelerate your learning, you want to learn management through the Enterprise Manager.

As you are reading the book, pay particular attention to Security, Roles, Tables, Views, and Stored Procedures. Don't worry too much about Triggers, Replication, and the Web at this point. You can come back and learn these later.

* **Tables** are similar to tables in Access

* **Views** are similar to select queries in Access.

* **Stored Procedures** are extremely powerful objects. I am going to describe them as a combination of Access queries and VBA code. The power is there, but trust me the ease of writing stored procedures for someone who is use to working with Access Queries, and VBA code is not there. There is no drop down lists, and you can't step through the code. You'll think your back in the DOS days to some extent. Stored Procedures:
  - > Can have Input parameters
  - > Can have Output parameters
  - > Can have parameters that are both input and output
  - > Can have 1 or more recordsets

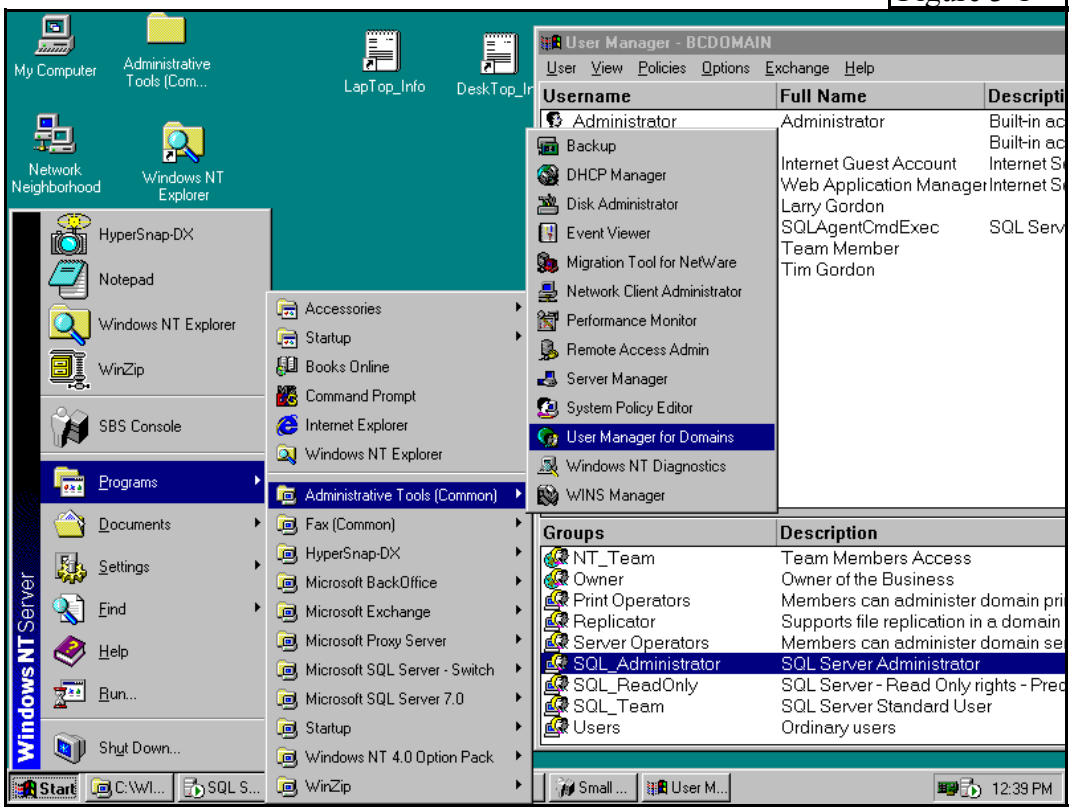
* **Security Users, and Roles**, are a very important part of SQL Server. There are basically 2 types of security, also referred to as authentication. "SQL Server and Windows NT" security, or "Windows NT only" (also known as "NT Integrated Security"). I use the combination so I can program my databases on my non NT machines (You can install SQL Server on Win 95/98/200 machines), however from my users point of view I use "NT Integrated Security", which means once they log into NT, their security is setup for SQL Server.

Figure 3-1

Security is a white paper of it's own, but I will cover some basics below.

Basic - NT Integrated Security

1) Set up users on your NT machine that host's your SQL Server. Go to "User Manager for Domains", then go to "User" on the menu and add a Group and put the appropriate people in the group. (*See figure 3-1).*



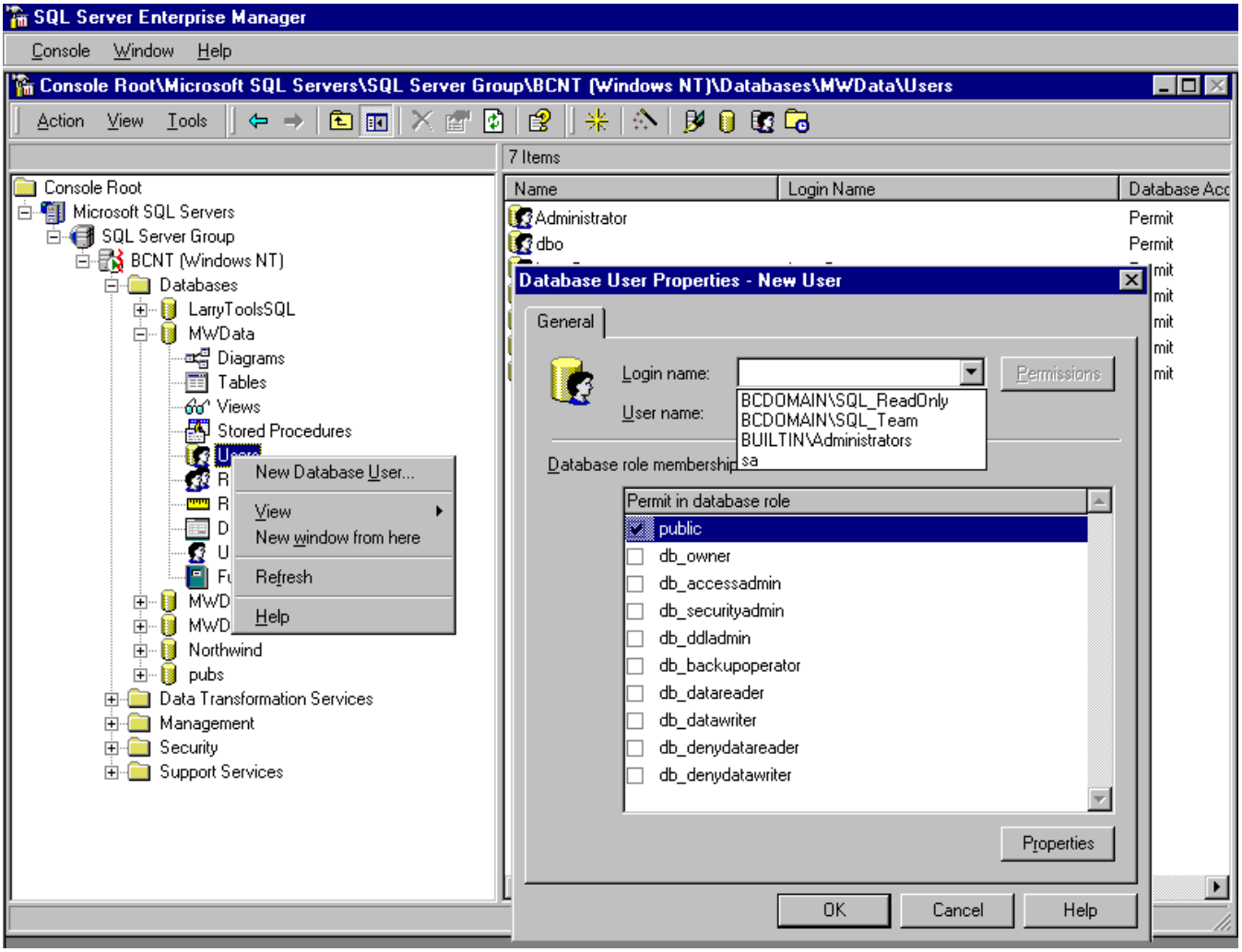
You might want to setup SQL_ReadOnly, SQL_Team, and SQL_Adminstrator groups. Then add the appropriate people to the groups.

2) Go into the SQL Server Query Analyzer and run sp_grantlogin system stored procedure for each group we just added. (Replace 'MyDomain' with the name of your server.)
Exec sp_grantlogin 'MyDomain\SQL_ReadOnly'

1) 3) Next, go to your database inside SQL Enterprise Manager. Right click on Users, pick "New Database User". This will bring up a property box in which you can pick the NT Group from the pull down list. Pick SQL_ReadOnly and click OK. The only permission should be public. Do the same for the other groups. (*See figure 3-2)*.

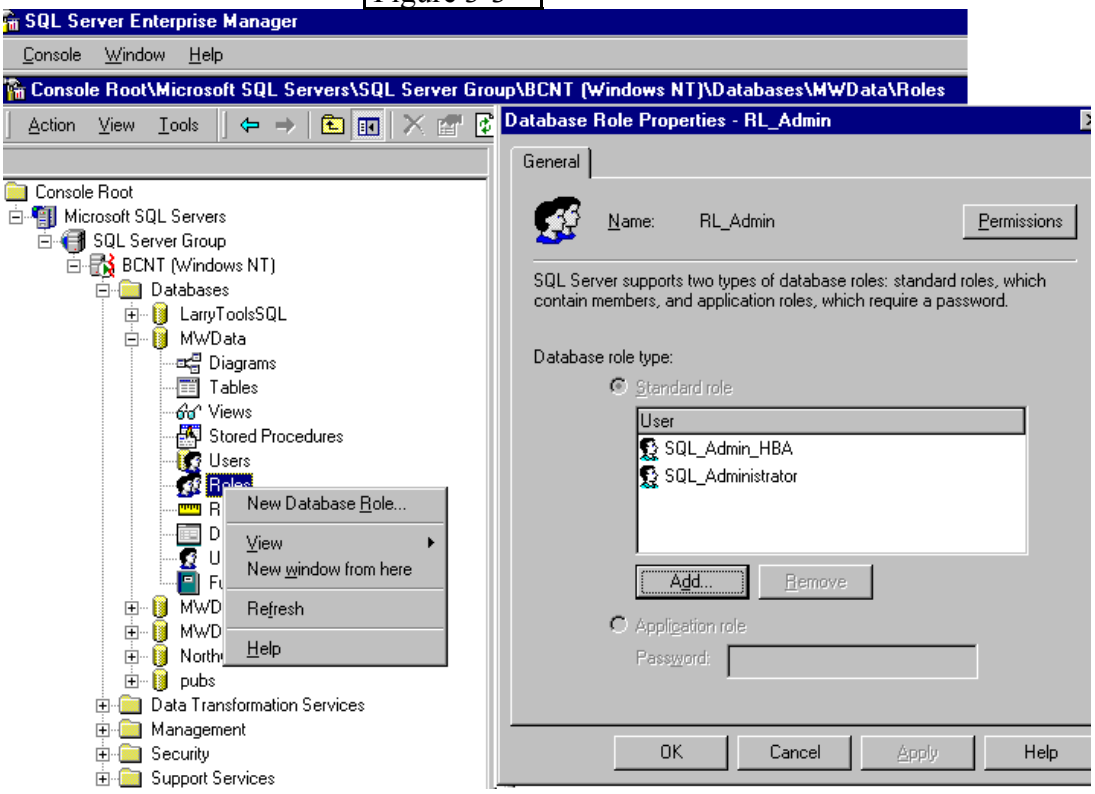
(Remember this is my way, read SQL Server 7.0 in 21 days for more complete information.)

Figure 3-2



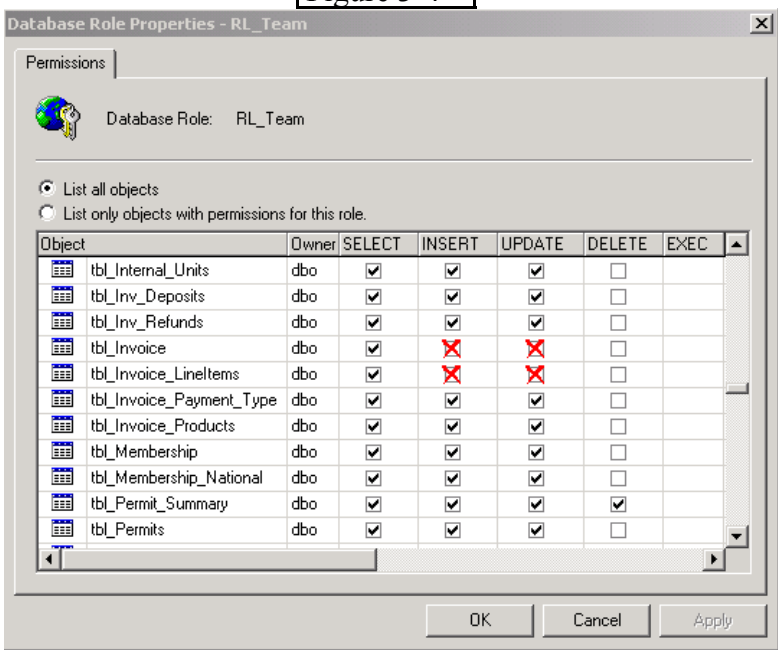
4) Next, go to your database inside SQL Enterprise Manager. Right click on Roles. Pick "New Database Roles", and create 3 database roles (for the sake of this article RL_Admin, RL_Team, and RL_ReadOnly). Each new role you add, add the approiate NT Group to the role. (eg SQL_Administrator to RL_Admin.).

Figure 3-3



5) Next, press the "Permissions" button on the appropriate SQL Server Role. Go to each object, and put a check mark on the appropriate permission you want to give your user.

Figure 3-4



* **Data Transformation Services (DTS)** is a very powerful way to move data to or from SQL Server.

* **Database Maintenance Plans** can optimize your database, check the integrity of your database, backup your database, and backup your logs and do it at times you set. (logs are a subject I'm not going to cover here).

* **Backups** are going to be a little foreign to us people who came from DOS to Windows, to Access. SQL Server can have many database connected to it. Each database has at least 2 physical files. MyDatabase.mdf which would be the database file, and MyDatabase_log.Ldf which is the log file (they can have different extensions, but mdf and ldf are the standard). If you copy the mdf file whether SQL server is running or not, you can not restore the database from the copied file.

  You have two choices, use SP_Detach_db stored procedure to detach the file, then you can make copies of it. And when you want to attach the copy you use Sp_attach_db Stored Procedure or Sp_attach_single_file_db. The other option is to do a backup of a database, and then a restore. With the SP_Detach_db everyone has to be out of the database (not out of SQL Server), with backup, people can keep working and no one knows the difference. I recommend doing a backup for backup purposes and moving the live database to a developer server.

  The book by Richard and Rick says in Day 7 Page 206 "SQL Server database backups are consistent at the point of the completion of a database backup. Therefore, if your backup finishes at 6 pm, you know that you have an image of all your data as of that time."

*** >>Tools that come with SQL Server that you need to learn.<<**

Enterprise Manager is the tool to manage SQL Server, the GUI enterface.

Books on Line is one of the best help systems I have ever seen. If you need to lookup a concept about SQL Server, it's there along with great examples.

Query Analyzer to some extent is like the Immediate Window in Access, it just much more powerful. ***Query Analyzer Could be a White Paper in Itself***

SQL Profiler allows you to see how your request for data is effecting the server.

*** >>A Tool you will want to have to manage your database.<<**

* **Moving Objects from one database to another** is something you will be doing if you are developing on a developers copy of the database, and moving changes to the live database. I recommend "SQL Compare" by www.red-gate.com. It will detectect the differences between the objects in 2 different databases, and script the updates. That way you can just bring a script file with you when you upgrade the live database, run the script, and your done. I wrote a review of the program and would be willing to send you a copy, or go to www.red-gate.com and get a 14 day full version trial.

#4 - You Need to Learn ADO.

* Your learning curve on ADO will be much faster than SQL Server. You can use your DAO code with SQL Server even transactions still work with DAO, however you do need to start moving to ADO. If you subscribe to one of the three Access publications, there are enough articles in the publications that you will be able to learn ADO by reading the articles on the subject in last 2 years.

Smart Access www.smartaccessnewsletter.com
Access VB SQL Advisor www.advisor.com
Office Pro (Use to be Office & Visual Basic for Applications Developer) www.msofficepro.com

* One of the best books on learning ADO is Alison Balter's book "Mastering Microsoft Access 2000 Development. Allison shows a lot of examples in ADO and if you are use to DAO, you will easily be able to follow Alison's book. (By the way, I don't like Allison's book on SQL Server & Access, she deals very little with Access mdb's.)

* Look at "Migrating from DAO to ADO" on Microsoft's web site. Print it out, put it in a binder, and read it. Then save it as a reference.
http://msdn.microsoft.com/library/techart/daotoadoupdate.htm

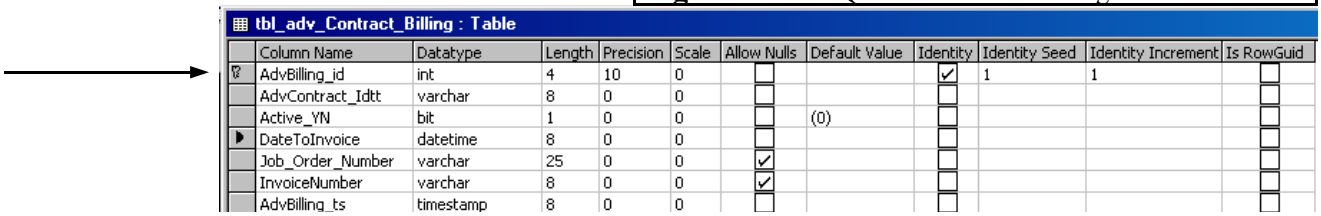
* "ADO 2.6 Programmer's Reference" by David Sussman is a great book. It's not easy reading but it has lots of examples, and as a reference it's great.

— Now - Let's Build some tables in SQL Server —

#5 - Some Facts - no matter how you move your data - or if you create tables from scratch

* Tables must have a primary index (key field) in order to update the table inside Access. You can have a table in SQL Server without a primary key, and set the primary key inside access as long as the field or fields are unique, however I recommend you set the field as key field inside SQL Server. (Note below, the primary key is an integer, we don't allow nulls, it's an identity column that we start with 1 and increment by 1)

Figure 5-1 SQL Server Table in design view



tbl_adv_Contract_Billing : Table

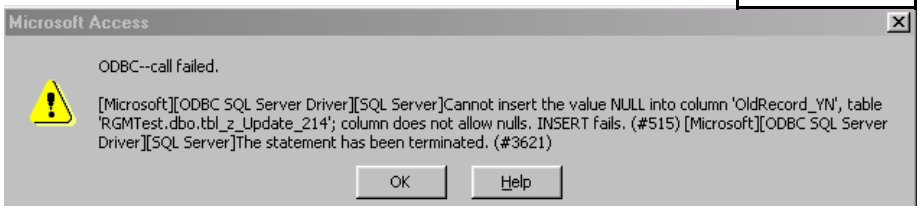
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment	Is RowGuid
🔑	AdvBilling_id	int	4	10	0	☐		☑	1	1	☐
	AdvContract_Idtt	varchar	8	0	0	☐		☐			☐
	Active_YN	bit	1	0	0	☐	(0)	☐			☐
▶	DateToInvoice	datetime	8	0	0	☐		☐			☐
	Job_Order_Number	varchar	25	0	0	☑		☐			☐
	InvoiceNumber	varchar	8	0	0	☑		☐			☐
	AdvBilling_ts	timestamp	8	0	0	☐		☐			☐

* It is recommended tables have a timestamp field if you are going to update the table through Access. The time stamp field allows SQL Server to know if anyone else has made a change to the record you are trying to update.

* You need to be careful with "Allow Nulls" check box. If you don't allow Nulls in a field, and you don't furnish a default value, you need to deal with it in access when you are updating your data. If you try to add a record to the table in a form (or anywhere else) without giving a value to the field, you will some times:
  a) Get an ODBC error that will make no sense to your user. See *Figure 5-2*
  b) Get no error message at all - the record will just not update or allow you off the record until you hit ESC 2 times.

So make sure you test for the value in the form, on the "before update" event and cancel the update if there is not a value.

Figure 5-2



* There are problems with true/ false fields. When you upsize or import to SQL 7, it makes

all true/false fields into bit fields (this is not a problem). All bit fields should >>not<< allow null and should have a default of 0 or 1. In my situation it was a real problem until I discovered the way to solve the problem.

If you have a bit field that allows nulls and you don't set the default value you are asking for problems in adding records. If you include the field in the query you are adding records to, and don't give a value to the field, Access will assign a value of 0 (false). However if the field is not included in the query you will get a message similar *Figure 5-2*.

Access 2000 has no problems internally interpreting a bit fields -
The bit field value of 0 = false, No, or off,
The bit field value of 1 = true, yes, on, or –1.
The above is true inside Access not inside SQL Server. Inside SQL Server views and stored procedures you must use 0 or 1.

#6 - Create a Database, Tables, and Views using the Upsizing Wizard

* The Upsizing Wizard does not work on all tables all the time. You may need to move your data other ways.

* The Upsizing Wizard is the only automated way I am aware of to move Access queries to views and stored procedures.

* Go get Microsoft's white paper on upsizing "Using the Access 2000 Upsizing Tools" at
http://msdn.microsoft.com/library/techart/acsqlres.htm
The paper goes into detail on how to upsize. Read it.

* There are problems with dates (My understanding is that they need to be between 1/1/1900 and 6/1/2079)

* If you tell the Upsizing Wizard to upsize to an Access Database, the wizard will only upsize your tables. You will want to move some or all of your queries over to SQL Server. The way to do that is to lie to the wizard. Tell the wizard you want to upsize to a project. When you upsize to a project it will bring all your tables, queries, forms, reports, macros, and code. That's more than we want to do, so see below.

* Move all the tables with the data, you want to upsize to a database with no other objects (We will refer to this database as A). After reading the instructions on upsizing wizard, try to upsize your tables. If you have problems, copy each table that has a problem to another database. Try to correct any problems with the tables, (Check out dates first). After you correct the problem, try upsizing the problem tables. If it works, move the corrected tables back the to database A.

* Next move all your queries Into database A. Fix your queries by taking out any references to forms, sorts, and the rest of the problems the above article talks about. Then run the Upsizing Wizard.

* The wizard, it will convert your text fields to nvarchar and your memo fields to ntext. If you think that this application will always be English, you probably want to change the fields to varchar and text. Then go in and change all you bit fields to <u>not</u> allow Null. I tend to move smallint to int.

#7– Moving data to SQL Server Using Data Transformation Services (DTS)

* DTS, can import data from almost any format into SQL Server or from SQL server to almost any format. This is a white paper to itself. We will deal with moving your data from Access to SQL Server here. It will not import queries.

* Decide what database inside SQL Server you want to create the tables in. (Or create a new database using the tools inside Enterprise Manager or through Access - File/ New/ Project (New Database)).

* Go to Enterprise Manager, right click on your database, and pick All Tasks, Import Data. The DTS wizard will appear.
    + Press the Next Button, you come to the page looking for the data source. This will be our Access database we will be importing our table structure and data from. Fill in the following:
        Source: Microsoft Access (The form will change - once you pick Microsoft Access)
        File Name: Push the 3 dot button to the right and pick your Access database.
        (Don't pick the program database with linked tables, pick the data database)
    —>Push the Next Button, unless you are using work group security on your database
        User Name: Person who has full rights to the tables in the database.
        Password: Password for the above person.
    —>If your System.mdw is the workgroup you are using for this database, Push the Next Button
        If you are using a different workgroup for this mdb, push the Advanced Button. Widen the Property column, and go to "Jet OleDB: System Database" Then to the Value Column and type in the path and the mdw to use. (eg C:\Windows\Access8\SYS_HBA.MDW).
    + The page after the data source, is the Destination page. Where do we want the data to end up.
        Destination: Microsoft Ole DB Provider for SQL Server
        Pick your database, security, and database. Press the Next Button.
    + Pick Tables. Press the Next Button.
    + Pick the Source Table(s) you want to transfer data from, type the Destination table name (it defaults to the source table name). The last column is Transformation and it has a button. If you push the button you can change field names, types, exclude fields, and decide what fields can be null. Press the Next Button.
    + Run the package immediately. You may or may not want to save the package.

* The DTS Wizard brings over the Primary Key data, however does not mark it as Primary Key. You will need to make sure all the tables have a primary index (key field) and a timestamp field in order to update the tables inside Access.

#8 - Create a SQL Server Database and Tables from scratch - or by Exporting Tables from Access.

* Create a new database using the tools inside Enterprise Manager or through Access - File/ New/ Project (New Database)

* You can Export an Access Database table to SQL Server (You will need a File DSN covered later in this paper). You export the table the same as exporting to Access, however you pick ODBC database instead of Access database.
* Use the field types on figure 8-1 to transfer you data type knowledge.

Figure 8-1 Data Types for Access and SQL Server

Access Data Type	SQL Data Type	Explanation Of SQL Data Type
Yes/No True/False	bit	Integer data with either a 1 or 0 value. Columns of type bit cannot have indexes on them. (It can be Null, but null can give you trouble later. I recommend you don't allow Nulls) Access stores True as –1 and False as 0 inside a Access table, however Access has no problems interpreting bit data - 1 = True and 0 = False.
Number (Long Integer)	int	Integer (whole number) data from -2^31 (-2,147,483,648) through 2^31 - 1 (2,147,483,647). About 2 billion minus to 2 billion plus
Number (Integer)	smallint	Integer data from 2^15 (-32,768) through 2^15 - 1 (32,767).
Number (Byte)	tinyint	Integer data from 0 through 255.
Number (Decimal)	decimal	Fixed precision and scale numeric data from -10^38 -1 through 10^38 -1.
	numeric	same as decimal
Currency	money	Monetary data values from -2^63 (-922,337,203,685,477.5808) through 2^63 - 1 (+922,337,203,685,477.5807), with accuracy to a ten-thousandth of a monetary unit.
	Small money	Monetary data values from -214,748.3648 through +214,748.3647, with accuracy to a ten-thousandth of a monetary unit.
Number (Double)	float	Floating precision number data from -1.79E + 308 through 1.79E + 308.
Number(Single)	real	Floating precision number data from -3.40E + 38 through 3.40E + 38.
Date/Time	datetime	Date and time data from January 1, 1753, to December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds.
	small-datetime	Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute.
	time-stamp	A database-wide unique number. A table can have only one timestamp column. The value in the timestamp column is updated every time a row containing a timestamp column is inserted or updated.
(Replication Id) Guid	uniquei-dentifier	A globally unique identifier (GUID).
	char	Fixed-length non-Unicode character data with a maximum length of 8,000 characters.
Text	varchar	Variable-length non-Unicode data with a maximum of 8,000 characters.
Memo	text	Variable-length non-Unicode data with a maximum length of 2^31 - 1 (2,147,483,647) characters.
	nchar	Fixed-length Unicode data with a maximum length of 4,000 characters.
	nvarchar	Variable-length Unicode data with a maximum length of 4,000 characters. sysname is a system-supplied user-defined data type that is a synonym for nvarchar(128) and is used to reference database object names.
	ntext	Variable-length Unicode data with a maximum length of 2^30 - 1 (1,073,741,823) characters.
	binary	Fixed-length binary data with a maximum length of 8,000 bytes.
	varbinary	Variable-length binary data with a maximum length of 8,000 bytes.
Ole Object	image	Variable-length binary data with a maximum length of 2^31 - 1 (2,147,483,647) bytes.

— How Do we Hook to SQL Server —

#9 - How to Hook an Access Project to Your SQL Database.

* I like to use SQL Server tools and an Access project to manage my SQL database. In an Access Project you have SQL Objects, which are not stored in the Access project. The Project just gives you a means to manage the SQL Objects. For example you *can not* add a local table to an Access Project, however you can create and manage the SQL tables.

The Access Objects can be used if you are building you database as an Access Project. In this paper we are talking about using the Project just to manage SQL Objects, so there will be no Access Objects.

SQL Objects seen in an Access Project	Access Objects
> Tables	> Forms
> Views	> Reports
> Database Diagrams	> Pages
> Stored Procedures	> Macros
	> Modules (Code)

* From the Menu in Access, click on File - New and *Figure 9-1* will appear. Click Project (Existing Database) and give your project a name. The Data Link Properties comes up (see *Figure 9-2)*. You then need to take the following steps.

1) Select the Server Name from the pull down list. Some times the pull down list is empty, you will then need to type in the server name.
2) Pick the type of security you are using, if you are not using NT integrated security, you will need to type in an user name and password.
3) Next select the database you want to manage.
4) Click OK, and your Project comes up with all your Tables, Views and Stored Procedures. *Figure 9-3*

Figure 9-1

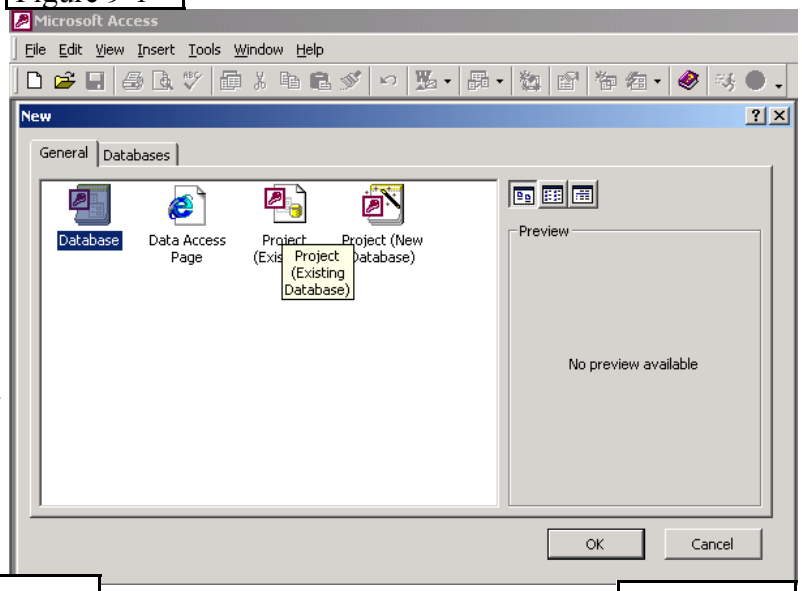


Figure 9-2

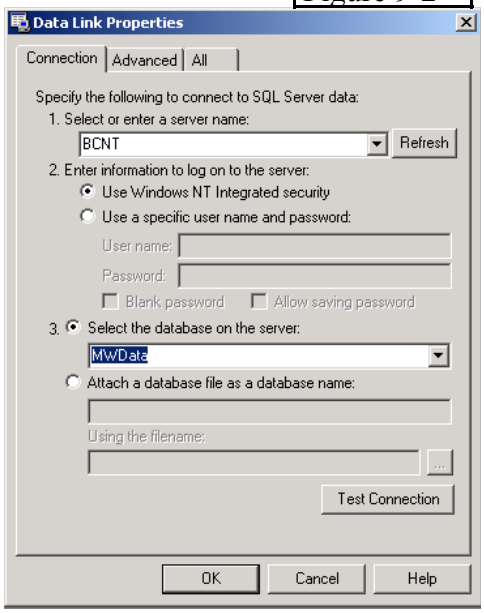
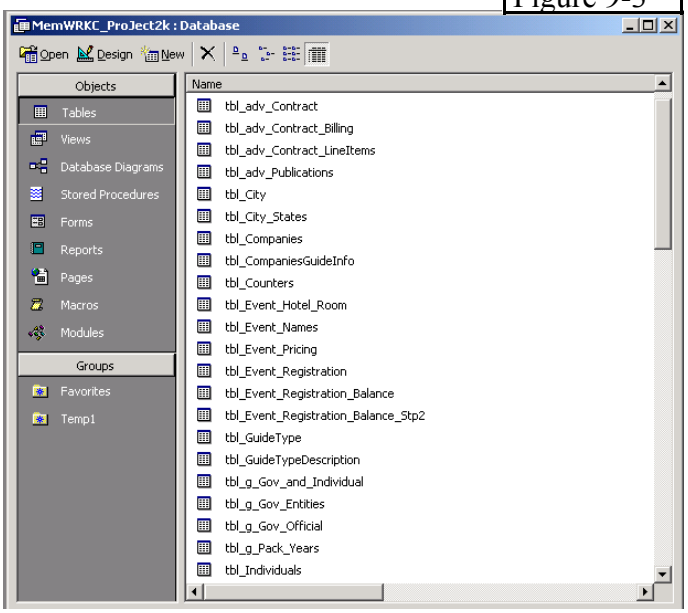


Figure 9-3



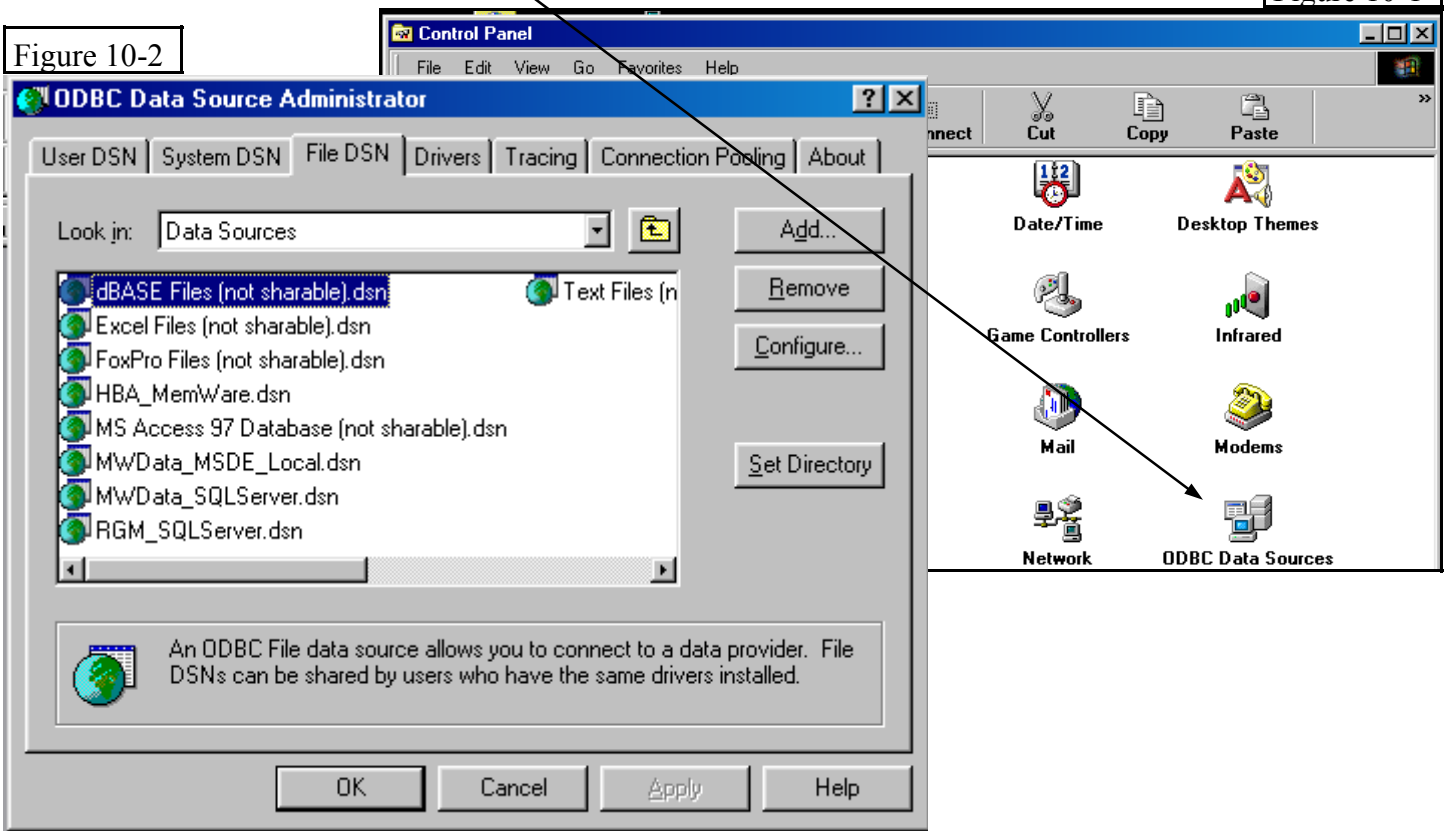
#10 - File DSN for Hooking to SQL Server through ODBC

* You will need to have a File DSN to link to tables and views in SQL Server. To set up a file DSN do the following steps.

1) Click your start button, Settings, Control Panel to bring up the Control Panel. (Win 2k go to Administrative Tools then Control Panel.)
2) Click the ODBC Data Sources icon. This will bring up the ODBC Administrator.
3) Go to the File DSN tab in the ODBC Administrator. (Figure 10-2)

Figure 10-1

Figure 10-2



4) Click the Add button.
5) Pick SQL Server, and then Next>. (Figure 10-3)
6) Enter the name you want to show in the list. Put a descriptive name you will remember. I usually put "SQL_" + the database name, and then Next>. (Figure 10-4)

Figure 10-3

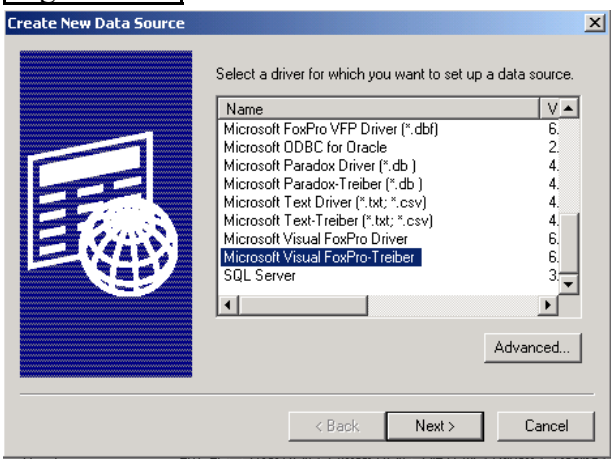
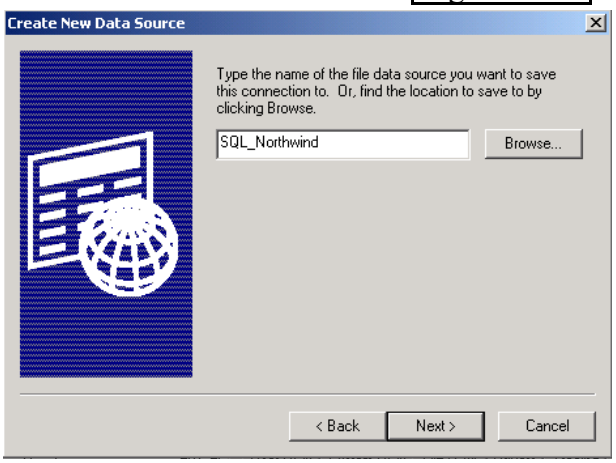


Figure 10-4



7) Click Finish. (By the way - you are not finished.) (Figure 10-5).

8) Put in a description if you want, then pick the SQL Server from the list. Some times it won't be in the list and you will have to type it in. Then Next>. (Figure 10-6)

Figure 10-5

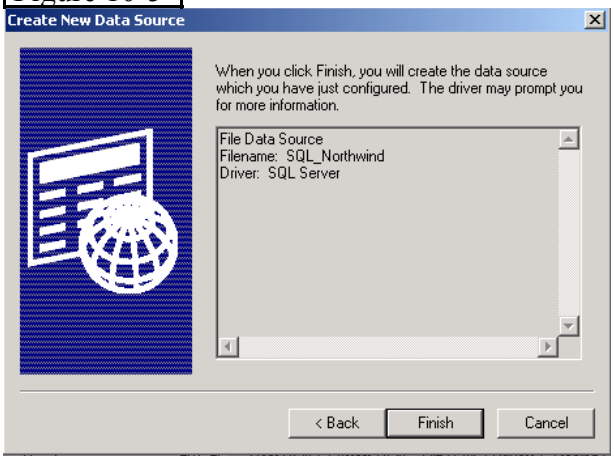
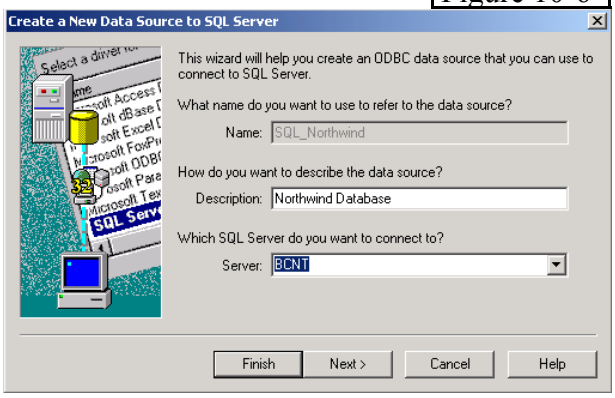


Figure 10-6



9) Pick the type of security. Then Next>. (Figure 10-7)
(I will be working with Windows NT authentication throughout this document.)

10) Put a checkmark in "Change the default database to:", and then pick your database. Then Next>. (Figure 10-8)

Figure 10-7

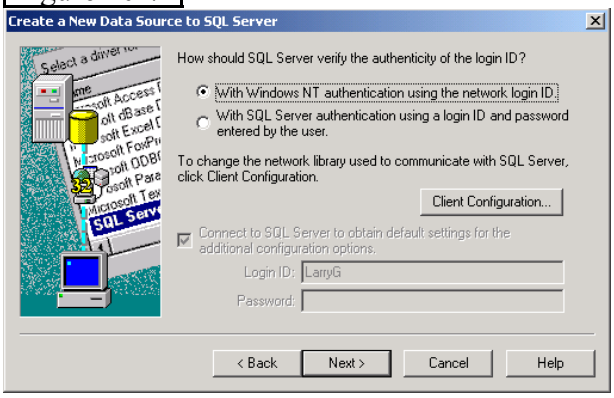
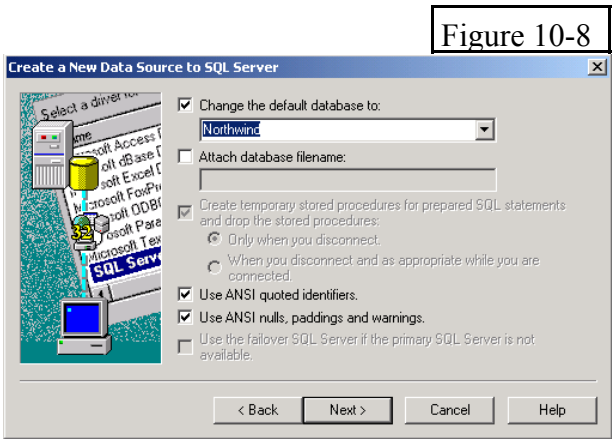


Figure 10-8



11) On Figure 10-9 click Finish. You are about finished.

12) On Figure 10-10 click Test Data Source to make sure you are connected. You are now finished.

13) You can now link tables from this database

Figure 10-9

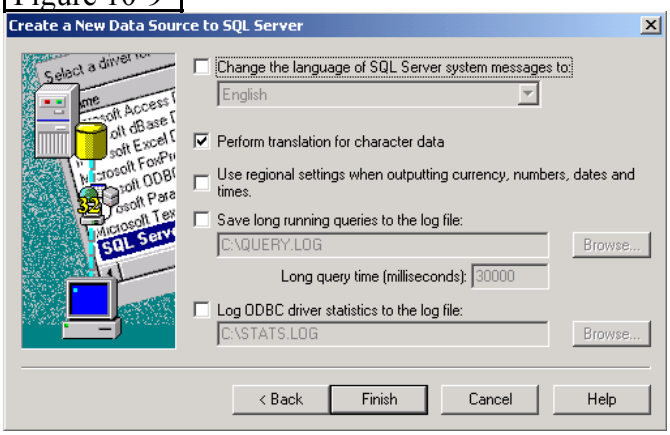
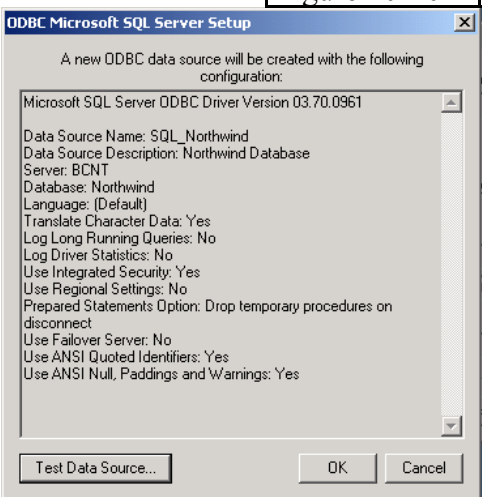


Figure 10-10



#11 - How to Link Tables and Views from SQL Server to an Access Database

* To link to tables once you have set up a File DSN, you do the following steps.

1) From the Access database window, go to the menu, File, Get External Data, Link Tables.
2) Pick ODBC Databases() from File of Type. (Figure 11-1)
3) Pick your DSN from the Select Data Source box, then OK. (Figure 11-2)

Figure 11-1

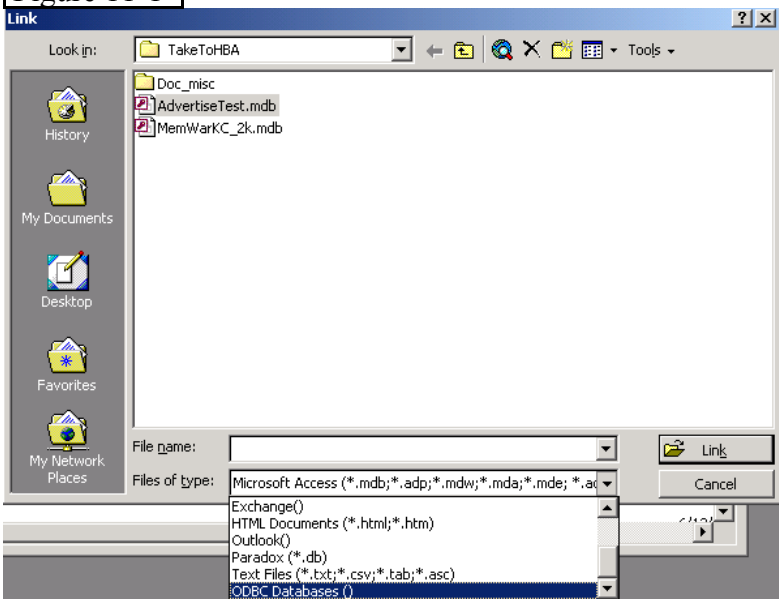
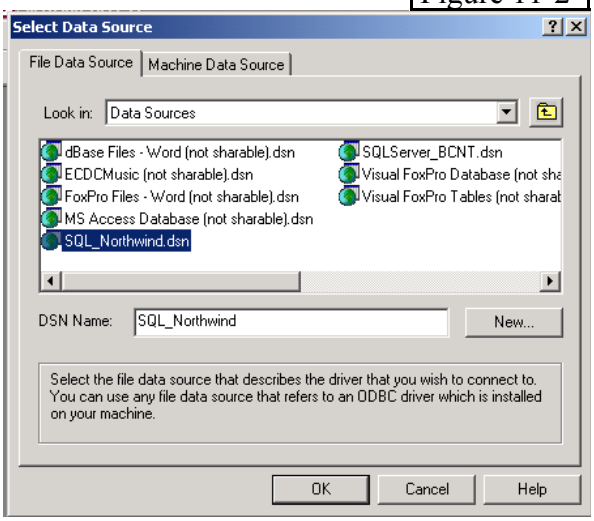
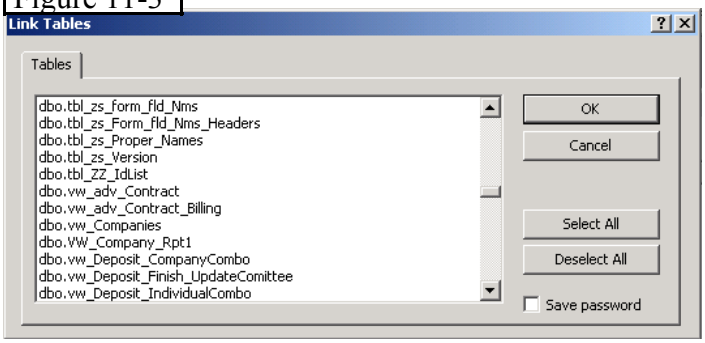


Figure 11-2



4) The Link Tables box comes up and you can pick your tables, and views you want to link. I usually rename dbo.tbl_Companies to tbl_Companies.(Notice that the owner of the object comes up. So tbl_Companies does not show as tbl_Companies, but dbo.tbl_Companies. This is because the owner of the object is "database owner". In SQL Server there could be a table dbo.tbl_Companies and one called Pete.tbl_Companies. It is generally agreed that this is bad programming to have duplicate objects with different owners, it can happen though.)

Figure 11-3



— How Do we Get Data from SQL Server —

When we talk about getting data from SQL Server, we need to divide the process into 3 categories:

1) Read data only (Give me the data, I don't need to edit it or add records)
2) Read, update, and add data. (we are looking at records in forms that we give the user the right to read, update and or add records)
3) What we in Access see as Action Queries (Insert, Append, Delete, and Make Table queries)

#12 - Reading Data From SQL Server

* REPORTS - are always read only. In Access you probably have some Access form that the user picks a particular report, and then possibly you give the user the ability to:

1) Open the report with only data they pick.
   DoCmd.OpenReport "Rpt_Customers_B", acViewPreview, , "tbl_Customers.Common_Name Like 'us*'"
2) The program does a lot of work in the background, creates a record set (table or query), and then changes the record source of the report, and then opens the report.

In all cases we are dealing with read only data. There are many ways to get to the data from SQL Server, some of them extremely fast. The following are some of the better ways.

1) Stored Procedures (SP) are the fastest way to get *read only* data from SQL Server (see more information on SPs in section #18. The following steps will move the data from SQL Server to your report.
   A) Create a SP in SQL Server, that returns a record set. Notice it has an Input Parameter @vcCompanyName. See *Figure 12-1*
   B) Set up a pass through query to bring the record set into Access. Notice it has an Input Parameter 'US'. See *Figure 12-2*
      (See section 16 on Access Queries, pass-through queries are covered at the end of the section)
   C) You can hook a report to a "pass through query" in the same way you hook a report to a select query. Just make the pass through query the record source of the report.
   D) Next, you setup a simple piece of code inside Access, to change the SQL statement of the pass-through query, and open the report. That's it, you just pass the correct parameter to the function when you want to print the report. See *Figure 12-3*
      Call PrintRpt_Customers_B("Smith")

Figure 12-1 Stored Procedure on SQL

ssp_CompanyRptB : Stored Procedure

```
Create Procedure ssp_CompanyRptB

@vcCompanyName as varchar(50)

As

set nocount on

SELECT tbl_Customers.Company_Idt, tbl_Customers.Active_Y_N,
       tbl_Customers.Common_Name
FROM tbl_Customers
Where tbl_Customers.Common_Name Like @vcCompanyName + '%'

return
```

Figure 12-2 Pass-Through Query (Design View) Property Sheet of the Query

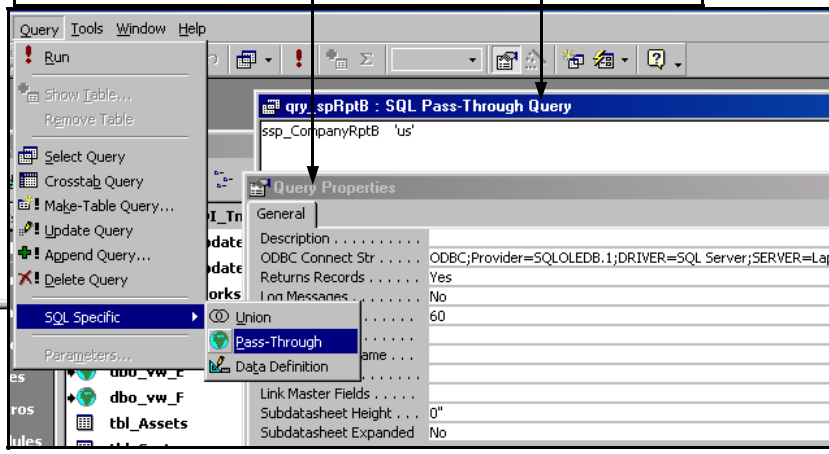


Figure 12-3

```
(General)                                   PrintRpt_Customers_B

Public Function PrintRpt_Customers_B(strCompanyName As String)

Dim db As Database
Set db = CurrentDb

  '--> Stored Procedure <---
   db.QueryDefs("qry_spRptB").SQL = "ssp_CompanyRptB '" & strCompanyName & "'"

DoCmd.OpenReport "Rpt_Customers_B", acViewPreview

ProcedureDone:

End Function
```

2) Views from SQL Server will give you acceptable speed. A below is a tiny bit faster than B, however you might want to test the results with your data.

A) The best way is to develop a View on SQL Server, and then call it with a pass-through query. You would pass a SQL statement with a where statement to the pass-through query using a similar method that is pointed out in *Figure 12-3.*

SELECT vw_Invoice.*
FROM vw_Invoice
WHERE vw_Invoice.Co_Alpha_Name Like 'w%'

B) A method that will give you similar speed to the above, is to link the view to Access (see section #11). Then you use the view the record source of the report. You can open the report in the same way you do your current reports today.

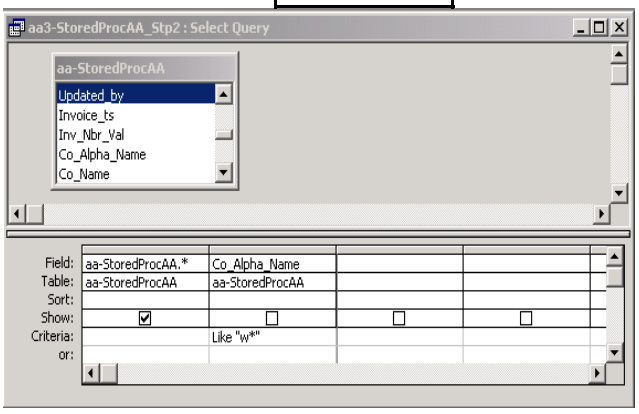
DoCmd.OpenReport "Rpt_Customers_B", acViewPreview, , "vw_Invoice.Co_Alpha_Name Like 'w*' "

C) You can bring a view into an Access query, do some calculations inside the query and make the Access query the record source of the report. You want to try to keep the above calculations out of the Where Statement, because they will slow down the process. You can open the report in the same way as B above.

3) Some additional ways to get data, that I don't recommend because they are slower than above, might be necessary in a very few instances, especially in a legacy application.

A) You can have a Stored Procedure, in most cases without a where and without parameters , that returns a record set (similar to *Figure 12-1)*. You hook to the proc with a pass-through query. Then you make an Access select query with the pass-through as the source. You can treat the final query in the same way you treat your current queries that are record sources for reports.

Figure 12-4



B) Use tables linked to SQL Server to make an Access query the way you do today. This is as fast as the views in #2 above if there is only one table involved. However if you have multiple tables with 10,000 records or more, this will be considerably slower in most cases, than using a view.
Remember >>>Never<<< use linked tables (or views) with local tables inside an Access query. This will slow down the query to unacceptable speed.

* Forms - We are talking about forms that the user is only reading data that comes from SQL Server. The Forms Record Source is the same as what we talked about with reports above. Below are some additional notes concerning Forms. By the way, they are all true in reports, however we see these situations in Forms more often.
    1) Please don't bring all the data into your forms, let the user limit it by specifying criteria – I use query by form. So however you do it, don't let the user bring back more records than they need/want to see. - limit the records so you are pulling 100 records from SQL Server, not 100,000.
    2) Never use combo or list boxes in a read only form. These take a record set behind the combo or list box. A second record set bringing data from SQL Server is going to slow down the process. If the user can't change the data, there is no need for the user having the ability to pick something.
    3) SubForms:
        A) You can't use a pass-through query as a record source for a subform if you use link master field and link child field. See MS article Q209116.
        B) What you will need to use as the record source of the subform is a Linked Table, Linked View, or an Access query hooked to one.
        C) Your form will be much faster, if you can hide your subforms, and have each sub form show when the user needs to see the data. Maybe they push a button, or click a tab, and you unhide the subform. Again we don't want to run a second record set if we don't have to.

#13 - Updating and Adding Records to SQL Server - Using Forms

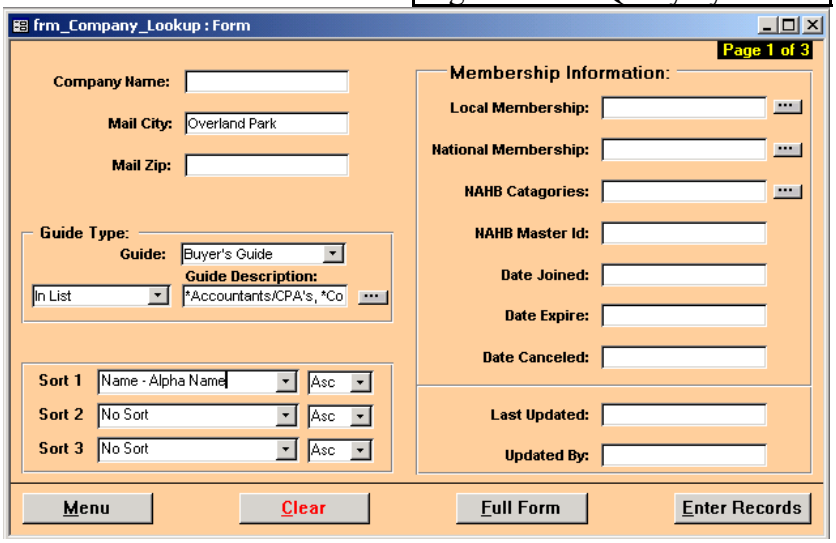
* Many experts on Access will tell you to create *unbound* forms when working with a client server database. I will not. Their concept is to call a function to get a stored procedure and with code, populate each control on the form. If the user changes a value on the form, they will use a stored procedure to update the data on the server. Each form is very time consuming to build, however the process works and works well.

One of the issues with Access is, that someone who knows Access and their data, can build *bound* forms that update themselves with very little development time involved. I believe this is extremely important if you work for organizations who constantly reengineering themselves and their databases to solve their problems and make themselves attractive to their customers and potential customers. The flexibility you will gain will greatly out weigh any of the benefits *unbound* forms may give. You might try each approach (bound & unbound) for yourself. SQL Licensing options might also be an issue. In this paper, I will cover bound forms.

* Record Source for the Form
    1) Stored procedures and pass-through queries are read only. You cannot use them for a record source of the form that you want to update or add data..
    2) Linked views are one of the best ways to deal with data in forms when you need to update and add records. If you have a view that is updateable and you can add records to it, you can use it by itself or inside an Access query. You then build your form in similar manor that you do today, based on the view as the record source.

Obviously you want the user to limit the data by specifying criteria. You would never load all the records from the table in the form. If you use query by form (similar to *Figure 13-1)* or other means to come up with a where statement, you can then set the forms record source to a new SQL statement. You might build the below SQL statement based on the linked view, vw_Companies, from the options the user picks.

Figure 13-1 Query by Form




```
SELECT * FROM vw_Companies
Where [Mail_Cty] Like "Overland Park*"  and
        ( Co_idT In(Select Co_idtt from  tbl_CompaniesGuideInfo where
        ( [Guide5] Like '*Computer Services*' or [Guide5] Like '*Accountants*' )))
ORDER BY [Co_Alpha_Name]  ;
```

If the variable, strRecSource , equaled the above SQL statement, you could then open the form with the below code.

```
dim frm as form
dim strCur_Frm_Nm as string

 strCur_Frm_Nm= “frm_Company”
set frm= forms!frm_Company

DoCmd.OpenForm strCur_Frm_Nm, acNormal, , , , acHidden
frm.RecordSource = strRecSource
DoCmd.OpenForm strCur_Frm_Nm, acNormal
```

The above works great. We can take tables with 100’s of thousands of records and return 200 records we are looking for in 1 to two seconds. (The first time a form with a lot of combo boxes and subforms opens, it’s going to take longer than 1 to two seconds) We can edit data in the form or add records to the form.

Some Good News about the above: Access working with SQL Server brings back about 50 records (Your user will never know) and lets you start working with the 50 records or so. You can go to the last record, or a particular record, but Access & SQL Server doesn’t automatically bring everything across the line. This increases the speed of showing the records to your user.

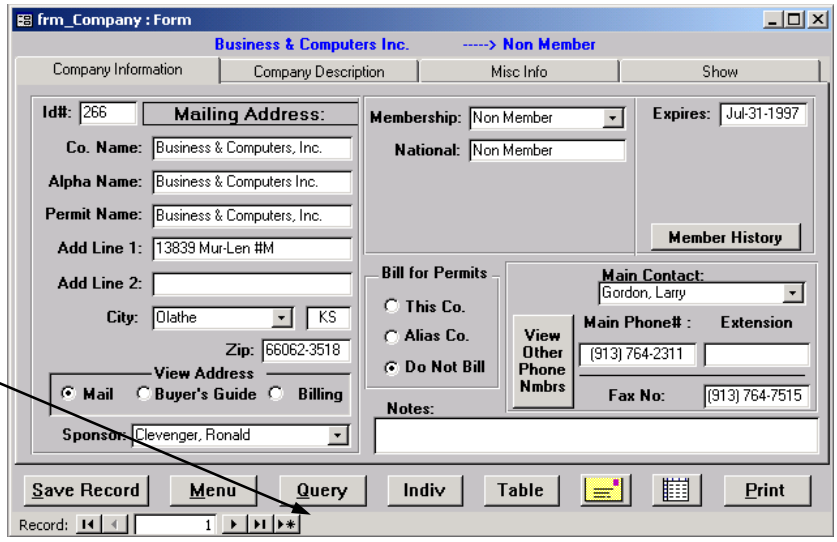
Some Bad News about the above: If you use this method, it will not show the total number of

records in the form, if more than about 50 records are returned. You can click the last record indicator and get the total number of records. (see *Figure 13-2)*

Figure 13-3
Form with data from SQL Server
You can view, update, and add records.

When using views in the above fashion, you do not get a record count when records are returned in the form.

You can click the last record indicator and get the total number of records for this query.



3) You can use ADO to set the record set property of a form to an ADO record set. You must use MSDatashape as the provider in order to have data you can edit and add records to in Access forms. The advantage of using SQL Server views is that you can use an SQL statement and it works rather fast. When you compare it to using a view in #2 above, these are the differences. Access with SQL Server will show the total number of records in the form, however it will take longer to see the first record.

This can be against a table (however you don't have to use the MSDatashape when using a table), or view which can but does not have to be linked to SQL Server. I link the view to Access, so I can use the view to build the form (Again rapid development). Below is a sample procedure (*Figure 13-3)* to use ADO and MSDatashape. You can call it with the following:

```
OpenFormWithShape("Where [Mail_Cty] Like 'Overland Park%'   ORDER BY
                  [Co_Alpha_Name] ", fm_Company,
                  " select * from vw_Companies ",  "tbl_Companies")
```

```
Public Function OpenFormWithShape(ByVal strWhere As String, ByVal strFormName As String, _
                    strSQLView As String, strTableUpdating) As Boolean

' Purpose: Open Form with a SQLView and Where Statement

' Example: Call OpenFormWithShape(Where_string, Cur_Frm_Nm, strSQLView, strTableUpdating, rs1)
'         See MemWareKC form frm_Company_Lookup for example
'----
' Parameters:  strWhere = Where string + Sort (If Where String then the word 'Where' is at the start of the string)
'             strFormName =The form we are going to open
'             strSQLView  = The View from the SQL Server we are using
'             strTableUpdating = The table in the above view we might be updating or adding records to.

'----
' Returns:
'----
    Dim cnn As ADODB.Connection
    Dim cnnShape As New ADODB.Connection
    Dim RS1 As ADODB.Recordset
'----
On Error GoTo HandleError
```

Figure 13-3 Continued

```
On Error GoTo HandleError
If Programming_Mode Then On Error GoTo 0

'Check the Connection - If no connection try to make one
   ‘If Not CnnCheckConnection() Then GoTo ProcedureDone

      Set cnn = New ADODB.Connection
      cnn.ConnectionString = "Provider=SQLOLEDB.1;" & _
                             "DRIVER=SQL Server;SERVER=bcnt;" & _
                             "Trusted_Connection=Yes;UID=;PWD=;" & _
                             "DATABASE=MWData;"
       cnnTemp.Open

  'Change to MS DataShape
   cnnShape.Provider = "MSDatashape"
   ‘cnnShape.ConnectionString = "Data " & cnn.ConnectionString
   cnnShape.ConnectionString = "Data " & cnn
   cnnShape.Open

   Set RS1 = New ADODB.Recordset
   Set RS1.ActiveConnection = cnnShape

   With RS1
      'if instr(strwhere,"Order by") between 2 and 4 then
      .Source = strSQLView + " " + strWhere
      .CursorType = adOpenKeyset
      .LockType = adLockOptimistic
      .CursorLocation = adUseClient  'adUseClient  '    adUseServer
      .Open
      'Debug.Print .RecordCount
   End With

   If RS1.RecordCount > 0 Then
     'Open Form Read / Write
      DoCmd.OpenForm strFormName, acNormal, , , , acHidden
      Set Forms(strFormName).Recordset = RS1
      If Not IsMissing(strTableUpdating) Then
         Forms(strFormName).UniqueTable = strTableUpdating
      End If
      OpenFormWithShape = True
   Else
      OpenFormWithShape = False
      MsgBox "No Records Match the Criteria You Entered.", 48, "No Records Found"
   End If
 '----

ProcedureDone:
   On Error Resume Next
   RS1.Close
   Set RS1 = Nothing
   cnnShape.Close
   Set cnnShape = Nothing
   Exit Function

HandleError:
   Msgbox Err.Description,
   Resume ProcedureDone

End Function
```

4) SubForms:
   A) Use a linked table or linked view, or an Access query hooked to one for subforms.
   B) Your form will be much faster, if you can hide your subforms, and have each sub form show when the user needs to see the data. Maybe they push a button, or click a tab, and you unhide the subform. Again we don't want to run a second record set if we don't have to.
5) Combo Boxes and List Boxes – It's A challenge
   A) You can use local tables with no problem for Combo or List boxes. The problem comes in keeping the data current/Up to date in the local table. There are many ways to keep it up to date, but there is not room for the details in this paper.
   B) I have found that using stored procedures along with pass-through queries are almost as fast as local tables. You need to check your own system and data.
6) A View Got Ya: If you have calculated fields in Views the calculation will not change until the next time you query the View. (Quantity * UnitPrice = Amount) If you change the quantity from 6 to 7, the amount won't change until the next time you query the View. Solve this by putting all calculations on the form or Access query.

#14 - What we in Access Call Action Queries

* We talk in a little more detail in the Stored Procedure in section #18.

* >Append Query< is much faster in a stored procedure inside SQL Server. It also works with well with a pass-through query, however it will work satisfactory with an Access append query in most cases.

* >Update Query< Updating SQL table 1 into SQL table 2 seems to have acceptable speed using an Access update query. Using a stored procedure with ADO or a pass through query works much better. (will be covered later). Using a SQL statement with ADO command object works great too.

* >Make Table Query< is either in a pass-through query or a stored procedure inside SQL Server. You can not use an Access make table query to make a SQL table.

* >Delete Query< Don't use an Access delete query against SQL table – way too slow. See stored procedures and ADO command object.

#15 - Linked Tables (SQL Server Tables Linked to Access)

* Linked tables work OK. You need to read Mike Gunderloy's article in Feb-2000 Smart Access "Access Answers: Client-Server Efficiency". (Mike talks about adp's, however the same is true with Mdb.) Basically the data for the table is not all sent across the wire, only the first X # of records. This is helped by the fact that Access saves the table schema and indexes locally.

  If you double click on a linked table (linked to SQL7) in the Access user interface, you immediately see a screen full of data. You do not see how many records in the table unless there are a very few records. The reason you don't see the record count is that Access & SQL Server work together, to only bring you approximately 100 or fewer records, and then stops the process. Later, in the background they might go out and fetch the rest of the records if the 2 computers are not busy. Or if you click the button for the last record, it will show you the last page of data, and the record count. You would not want to use tables from the database window like this, but these facts will enter into decisions you make in building your database.

* Make sure your are aware that if you are going to Update/Add data to tables, you will need a primary index (Primary key field) on the table and it is recommended you have a timestamp field (*See Section*

#5). When you create the table inside SQL Server you should create a primary key field. If the table doesn't have a Primary Key field, when you link it to Access you will get the dialog from Access *figure 15-1*. Access wants you the pick a primary key field.

If you change the connection string of your tables (Note I do this all the time – I hook to my customers SQL Server, my SQL Server, and on my laptop I hook to my MDE) the primary index goes away. One way to reset it is to have an Access query as follows. (see *figure 15-2)*
CREATE UNIQUE INDEX [Index Name] ON [View Name] ([Field Name]) WITH DISALLOW NULL
See the note below on resetting the connect property in code. You need to reset any primary keys that are not set on tables inside SQL Server, with this code.

Figure 15-1

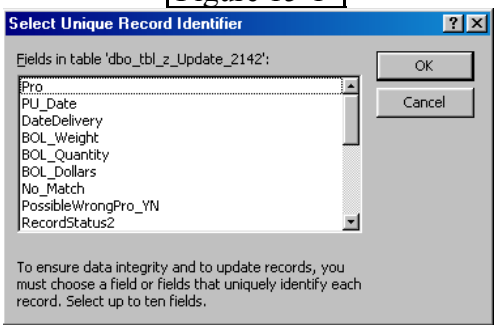
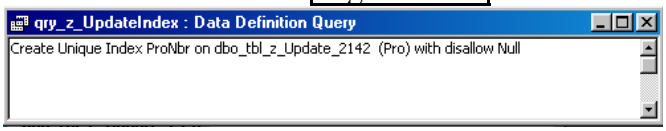


Figure 15-2



This Could be a White Paper in Itself

* If you change a table design (also View design) on SQL Server, because Access saves the table schema and indexes locally, you need to do one of the following for any linked tables or linked views. If you don't do one of the below, you will not be able to read the table/view inside Access:
  1) Refresh the links for all tables and views (Just the table/view that was changed would do the job)
  2) Relink the tables/views
  3) Reset the Connect property of the tables, and views.

  The best thing is to set up code to reset the connect property for you. You might want to hook your mdb to your SQL server, or your client's SQL server, or the MSDE on your laptop at different times. The connect property will allow you to change between different servers or MSDE. If you write this code you need to deal with linked tables, views, pass-through queries, and primary key on views. If there is an interest in my code to do the re-linking, I would be willing to put together a sample at a later date.

* If you look at fields as having 3 options, be aware of a potential problem.
  1) Null Values allowed
  2) Default Value
  3) Neither of the above 2

* If #3 above is true, make sure you check for a value in the field before you allow the user to update the form. You will try to go to the next record and Access will just sit there and do noting (or other crazy things) if it is expecting a value in the field. (Some day they might put in an error message – however that would make the programmers job too easy.) Access & SQL Server do not always communicate well - it might say "ODBC – Call Failed", no error message, or any other error message.

#16 - Access Queries

* If you are using an Access backend (the data is stored in Access) you would use a local query with tables linked inside a query. You can do this with small tables, or where there is only one table in the query, however you don't want to have a query with 3 or more SQL Server tables, in an Access query. You want to use Views inside SQL Server linked to Access. Under the right circumstances, you can

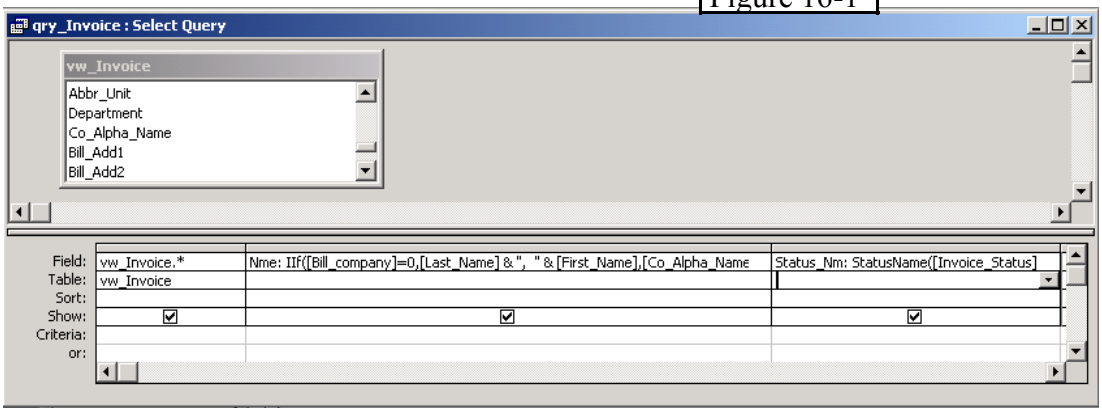
add data to Views and edit data in views. By having the View in SQL Server, only the data you want is moved. The work is done on SQL Server.

* You >>Never<< want to have an Access query with a mix of local Access tables and Linked SQL Server tables. This mix will slow the query down to "Get a cup of coffee" time. In some circumstances it will time out the server.

* If you have written functions that you use in the query, you can't use it in a View. Also, you can't use a IIF statements in views. The way you solve this is one of two way. You can move your query to a stored procedure, or you can use a combination of a view in an Access Query with one or more functions or IIf statements.

  If you use an Access query with a linked SQL view with IIf statements and/or functions, you need to make sure you don't use criteria with the IIf statements and functions. If you do use criteria, or use them in a Where statement, SQL Server has to send over every record to Access from the table, and then access decides what records it wants to keep. This is a very time consuming process. If you need this capability, use a stored procedure.

Figure 16-1



* You can also use a pass-through query with a stored procedure along with an Access Select Query. This give you a lot of flexibility. In *figure 16-2* you see a pass-through query with parameters. We will only bring back the data we want. Then in *figure 16-3*, we have an Access select query based on the pass-through query. If gives us a lot of flexibility.

Figure 16-2

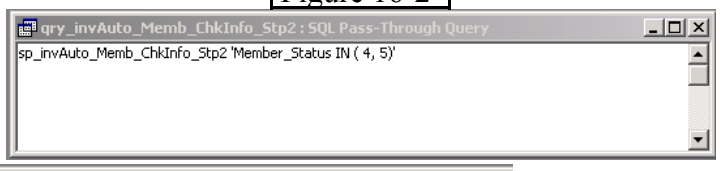
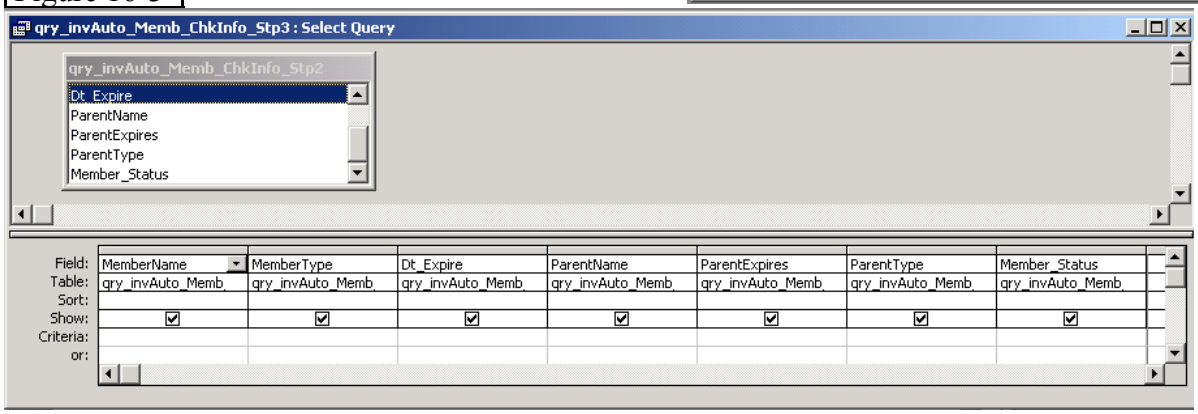


Figure 16-3



* Pass-through queries in Access give you a lot of power. You can hook to stored procedures, views, or just send an SQL statement. Access doesn't interpret anything you put in a Pass-through query, it just sends it to the server. (Remember Pass-through queries are always read only.)

  To build a pass-through query, you need to go to the property sheet on put in the ODBC Connect String. Your connection string will be different, however it will be similar to *figure 16-4*. Once you build the first pass-through query, just copy it and change the SQL statement. That way you won't have to re-type the Connect String each time.

Figure 16-4

```
ODBC;Provider=SQLOLEDB.1;
DRIVER=SQL Server;
SERVER=bcnt;  <—— (Your Server' s Name)
Trusted_Connection=Yes;
UID=;
PWD=;
DATABASE=MWData <-(Your data base Name)
```

Figure 16-5 Pass-Through Query (Design View) Property Sheet of the Query

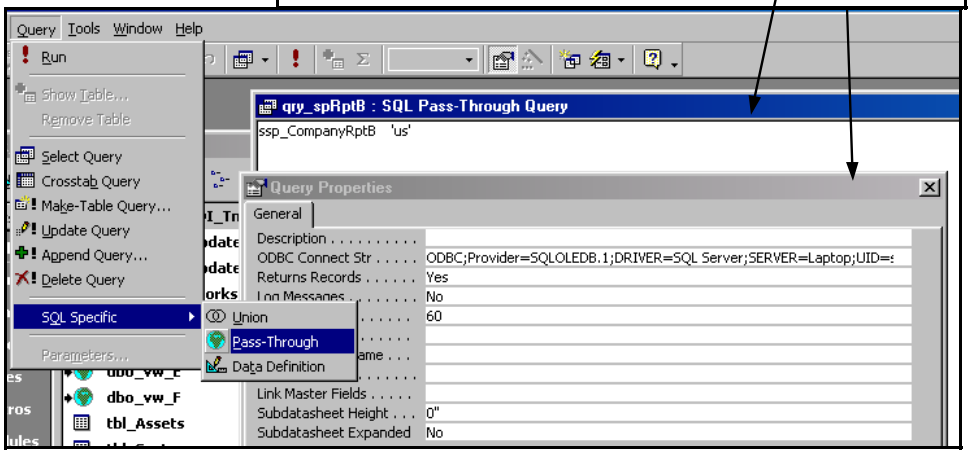


Figure 16-6

Some ways to use pass-through queries.	**Example**
Stored Procedure	ssp_CompanyName
Stored Procedure - with parameters	ssp_CompanyReport_C 'Builder Member', 'Olathe'
SQL statement using a table or a View	select * from tbl_Invoice where Date_Due < '6/1/01' and invoice_status =3
Deleting records, appending, and updating	delete tbl_City where City_Id=159
Many other things I will not cover in this paper	

* Access Select Queries.
  1) Keep to 1 table or use SQL Server View
  2) Access interprets anything you put in a select query, and tries to translate it to SQL Server. It does a good job of this.
  3) Don't link Access Table with SQL Server Table – You have to bring all the records over the line in order to link them. This increases time dramatically.

* See section #14 for information on Access Action Queries.

#17 - Views – Moving Access Queries to SQL Server Views

* SQL Server Views are basically select queries. If you have more than 1 table (it depends on the tables – check out the speed) in your Access select query, you probably want to move the query to a SQL View, it will be faster most of the time. What you would be doing is moving an Access query that was hooked to 3 tables linked to SQL Server, to a SQL Server View, and link the View to Access.

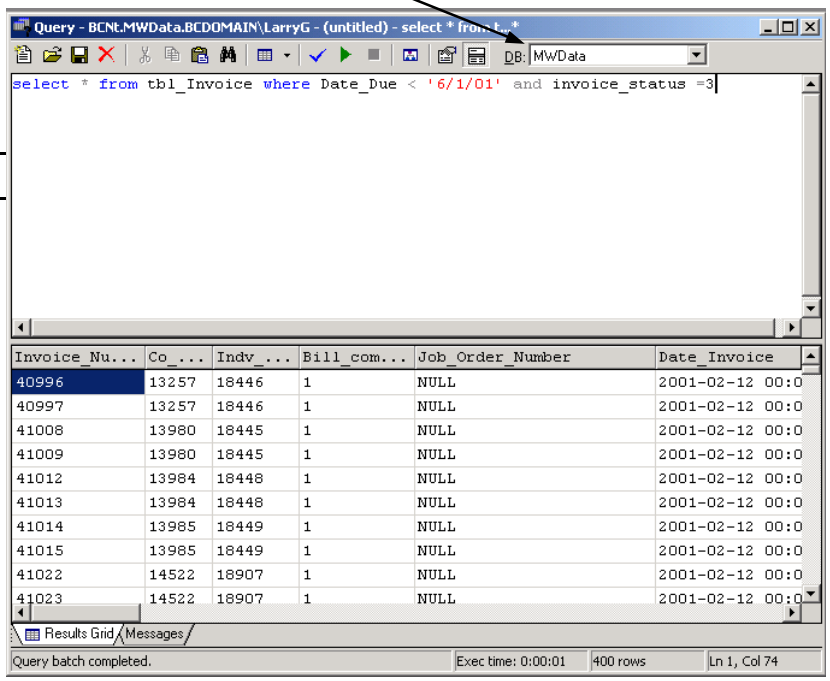
This Could be a White Paper in Itself - along with a function to help with this process.

In the below, I presume the names for the tables in Access are the same name in SQL Server.

* To build a SQL View from an existing Access select query you need to:

1) Go to the SQL view of the Access query.
2) Copy the SQL statement of the query. (Get rid of the semicolon at the end.)
3) Put the SQL statement in the SQL Query Analyzer (One of the SQL Server Tools). Make sure you pick your database in the top right corner of the Query Analyzer.
4) Get rid of the Order By statement - SQL Server Views do not support sorting.
5) Run the query, if it runs, go to step # 12.

Figure 17-1



6) If you have written functions that you use in the query, you must get rid of them. (Consider going to a stored procedure. Also see the notes in section #16 on using functions Access Queries connected to SQL Tables & Views)
7) If you have used IIF statements, you must get rid of them.
   (Consider using the Case statement in a stored procedure. Also see the notes in section #16 on using IIF Statements in Access Queries connected to SQL Tables & Views)
8) If there is a built in function and it fails on one record - the entire View will fail in Access.
   (e.g. If you have a varchar field called Idt that is normally all numbers (123, 456, 789) and you have one record with numbers and text (123ab) and you built a function around this field in a SQL view convert(int, Idt) , you have problems. Once Access hits the record where there is a problem, all fields in all records of the view come back with #Name?.)

9) The "From Statement" is usually OK, however you will need to work with the "Select" and the "Where Statement".
   If you refer to an Access form or control in the where - get rid of it.
   You will need to make changes if you use any of the built in functions, delimiters, etc. referred to in figures 17-2, 17-3, and 17-4.
10) Run the query in the SQL Query Analyzer, if it runs, go to step #12.
11) If it doesn't run - you need to find where the statement is giving you trouble.
12) Copy the SQL statement to a View in SQL Server, and save the View. You can do this in an Access Project hooked to your database or through SQL Server's Enterprise Manager.
13) Link the View, like you link a table.

Figure 17-1 **>>Numeric Functions<<**

Access	SQL	Explination
int(7.234)	Floor(7.234)	Convert to integer
Round(765.4321, 2)	Round(765.4321, 2)	select Round(765.4321, 2) returns 765.43
Val("123 Main")	Nothing available in SQL Server	Val() returns a numeric value at the beginning of a string Val("123 Main") returns 123 Val("Main 123") returns 0
	IsNumeric(Zip)	ISNUMERIC returns 1 when the input expression evaluates to a valid integer, floating point number, **money** or **decimal** type; otherwise it returns 0. A return value of 1 guarantees that *expression* can be converted to one of these numeric types.
24 **mod** 5	select 24 % 5	Used to divide two numbers and return only the remainder. Always uses an integer. If you use non integers in Access, Access turns them into integers, SQL Server returns an error

Figure 17-2 **>>Type Conversions<<**

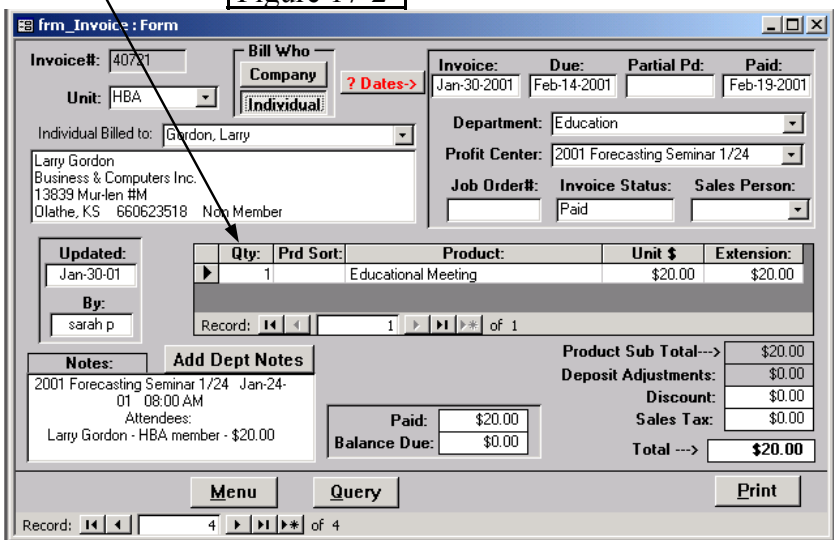
Access	SQL	Explination
Cint(X)	Convert(int, X)	Convert to Integer CInt("876.54") equals 877
Cdbl(X)	Convert(float, X)	Convert to Double Precision
Ccur(X)	convert(money, X)	Convert to Currency
Cstr(strX)	convert(varchar, X)	Convert to String
CDate(X) or CVDate(X)	Convert(DateTime, X)	Convert to Date/Time

Figure 17-3 **>>Misc. Information<<**

Explanation	Access	SQL
Date Delimiter Access-> Between #1/1/01# and #12/31/01# SQL->Between '1/1/01' and '12/31/01'	#	'
String Delimiter Access -> "Gordon" & ", " & "Larry" SQL -> 'Gordon' + ', ' + 'Larry'	"	'
Concatenation Operator	&	+
Wildcard Character (Any *one* character) Where Last like "Gor?don" Access -> select last_Name from tbl_Individuals where last_name like "Gor?on" SQL -> select last_Name from tbl_Individuals where last_name like 'Gor_on'	?	_
Wildcard Character (Any *group* of characters) Access -> select last_Name from tbl_Individuals where last_name like "Gord*" SQL -> select last_Name from tbl_Individuals where last_name like 'Gord%'	*	%
True/Yes	True, Yes, On, -1	1
False/No	False, No, Off, 0	0
In versions of Access prior to Access 97 when you created a query, Access created a SQL statement like Select DistinctRow Company_Id, CompanyName … I recommend to get rid of DistinctRow except when needed (Look up DistinctRow in Access Help for further explanation)	DistinctRow	Distinct

* A View GotYa: If you have calculated fields in Views the calculation will not change until the next time you query the View. (Quantity * Price = Amount) If you change the quantity from 6 to 7, the amount won't change until the next time you query the View. Solve this by putting all calculations on the form or Access query.

Figure 17-2



=Update/Add Data Considerations=

* If you link the View, Access will ask for a unique key, you must give the View an unique key, or you will not be able to Update/Add data. You can make a field the primary key in the same way as with a table. See section 15 and *figure 15-2*.

* If there is a Where statement in a view you are looking at a possible problem when you add or edit a record. If you have a Where statement hard coded in the view, which states Where tbl_Individual. LastName Like 'S%', everything will work fine if you add/update LastName to Smith, however if you add add/update to Jones, the record will truly be added/updated, but the view will show #Deleted for each field in the record.

* If a View has a concatenated field, you will not be able to Update/Add data. (eg Cty_St= [City] + ', ' + [State])

#18 - Stored Procedures (SPs) Inside SQL Server.

This Could be a White Paper in Itself

* Stored Procedures are one of the most powerful pieces of *programming* you will ever see. When you

Figure 18-1

z_sp_In_Out_Parameters_Simple : Stored Procedure

```
Alter Procedure z_sp_In_Out_Parameters_Simple
        @vcCo_Idt as varchar(10),
        @vcOutPut1 AS   varchar(100) output
As

set nocount on

/*      Input   @vcCo_Idt =266
        OutPut  @vcOutPut1 = Business & Computers, Inc */

   select  @vcOutPut1 =Co_Alpha_Name from tbl_Companies where Co_IdT = @vcCo_Idt

return
```

start out you will see them as a way to return a record set, or do some small update on your data. As you learn more about SPs you will understand why there are entire books written on the subject. SQL Server compiles the Proc so that when you run it, it runs as fast as possible. Once you write a couple of complicated SPS, you will be convinced. I will only cover some of the minor factors in SPS, you will need to read considerable more.

* SPS can have Input, output, parameters and have parameters that are both input and output. They also can have one or more record sets. For an example of a simple SP with input and output parameter, see *figure 18-1*. In the SP we input a company Id (@vcCo_IdT) and return the company name in the output parameter. We run the SP with ADO Code. (see *figure 18-2)*

Figure 18-2

```
Public Function ex_SP_In_Out_Parameters_Simple_2()

'On Error GoTo HandleError

Dim Cmd1 As ADODB.Command
Dim lngRecordsAffected As Long
Dim cnnTemp As ADODB.Connection
Set cnnTemp = New ADODB.Connection

cnnTemp.ConnectionString = "Provider=SQLOLEDB.1;" & _
                           "DRIVER=SQL Server;SERVER=bcnt;" & _
                           "Trusted_Connection=Yes;UID=;PWD=;" & _
                           "DATABASE=MWData;"
cnnTemp.Open

'----
  'Open Command Object
   Set Cmd1 = New ADODB.Command
   Cmd1.ActiveConnection = cnnTemp

'---
   With Cmd1
      .CommandText = "z_sp_In_Out_Parameters_Simple"
      .CommandType = adCmdStoredProc
      .Parameters.Refresh
      .Parameters("@vcCo_IdT").Value = 266
      .Execute , lngRecordsAffected, adExecuteNoRecords
   End With

   Debug.Print Cmd1.Parameters("@vcOutPut1").Value

   Set Cmd1 = Nothing

ProcedureDone:
Exit Function

HandleError:
   Debug.Print Err.Number, Err.Description
   Resume ProcedureDone
```

Figure 18-3

cc : Stored Procedure

```
Alter Procedure cc
As
set nocount on
        SELECT Co_Alpha_Name,  Mail_Cty,
        State =
          Case Mail_St
         When 'KS' then 'Kansas'
         When 'MO' then 'Missouri'
          else Mail_St
        end
    FROM vw_Companies where not (Mail_St  is null)
return
```

Figure 18-4

cc : Stored Procedure

Co_Alpha_Name	Mail_Cty	State
Welcome Home Properties	St Charles	Missouri
Carpet Alliance	Lee's Summit	Missouri
Valley Floor Covering	Lee's Summit	Missouri
Gibson Companies Inc The	Kansas City	Missouri
Remodelors Council	Kansas City	Missouri
Burke Construction	Lee's Summit	Missouri
Leader Mortgage	Lenexa	Kansas
RARE Enterprise	Kansas City	Missouri
Blind Guy, The	Harrisonville	Missouri
Veronicas Construction	Kansas City	Missouri
Witkin Custom Homes Inc	Englewood	CO
Gach Mark Construction Inc	St Joseph	Missouri
Keach & Grove Real Estate	Bedford	IN

Record: 1 of 9181

* One of the questions you will come up with is, what is this "set nocount on" you see in some SPS. If you don't set "set nocount on" in a SP, when you run the SP in the Query Analyzer, you will get back a message "X records were affected". By setting nocount on, it stops SQL Server from doing some work, that you don't care about. This will cause the SP to run just a little faster.

Figure 18-5
Append data from 1 table to another table

z_sp_Qry_Append : Stored Procedure

```
Alter Procedure z_sp_Qry_Append As

INSERT INTO tmpCity  (City, County)
SELECT  City, County FROM tbl_City
Where City_id<>1
```

Figure 18-6
Update the date (dt_expire) by 1 year

z_sp_Qry_Update : Stored Procedure

```
Alter PROCEDURE z_sp_Qry_Update  AS

update tmptbl_Companies
set dt_Expire= dateadd(yy,1,dt_expire)
Where Co_Alpha_Name like 'b%' and dt_Expire is not null
```

Figure 18-7
Update 1 table using data in another table

z_sp_Qry_Update : Stored Procedure

```
update tbl_Permits
        SET tbl_Permits.Invoice_No = tbl_za_PermitInvoicing.Invoice_Number,
            tbl_Permits.Billed = tbl_za_PermitInvoicing.Billed,
            tbl_Permits.Quarter = tbl_za_PermitInvoicing.Quarter
    FROM tbl_za_PermitInvoicing  JOIN  tbl_Permits ON
             tbl_za_PermitInvoicing.Pmt_Idd = tbl_Permits.Pmt_Nm
```

Figure 18-8
Make a Table

z_sp_CreatePerminetTable : Stored Proce...

```
SELECT Co_IdT, Co_Alpha_Name
into AAA
FROM tbl_Companies
WHERE (Co_Alpha_Name LIKE 'c%')
```

* If you are like me and use the "IIf" statement in Access queries, you are going to want to know what you can replace it with in SQL Server. There are no replacements in Views, however in SPS you can use the case statement. In *figure 18-3* we have a SP that looks at the field Mail_St which is a 2 character field for the state. If it = KS we substitute Kansas, if MO we use Missouri, otherwise we use the actual value in the field Mail_St. You can see how it comes out in *figure 18-4*.

Figure 18-9
Delete records in a Table

dd : Stored Procedure

```
Create Procedure "dd"
As
set nocount on

delete tbl_City where City_Id=99

return
```

Figure 18-10	>>String Functions<<	
Access	**SQL**	**Explanation**
Asc(X)	Ascii(x)	Returns the ASCII value of a character Asc("A") will return 65
Chr(X)	Char(x)	Returns a character associated with the specified character code. Chr(65) will return A
Instr("XYZ", "Y")	CharIndex('Y', 'XYZ')	Find a position of a particular string Instr("XYZ", "Y") returns 2
Lcase(x)	Lower(x)	Change to lower case0 SELECT Lower('THIS IS HOW THE MAIN FRAME PROGRAM-MERS USE TO DO IT')
Left("ABCDE",2)	Left('ABCDE', 2)	Left characters of a string Left('ABCDE', 2) returns AB
Len(X)	Len(X) or DataLength (x)	select LEN('This is a test') returns 14
Ltrim(x)	Ltrim(x)	Trim the spaces off the Left of a string Ltrim(" SQL") returns "SQL"
Mid("Test This",6) Mid(Expression, Start, Length)	Substring("Test This",6, 20) Substring (Expression, Start, Length)	In SQL Server you have to put the length, however in Access you are not required to have the length. The secret in SQL Server is to put the maximum length it could ever be (if it's greater than string length, that's not a problem).
IsNull([Dt_Join]) Not IsNull([Dt_Join])	([Dt_Join] IS NULL)) Not ([Dt_Join] IS NULL))	Check to see if a value is null select * from tbl_Companies where not ([Dt_Join] IS NULL)
nz([Price], 0) nz([Price], "Free")	IsNull([Price], 0) IsNull([Price], 'Free')	If the price is null, return 0, else return the Price If the price is null, return Free, else return the Price
Replace("aabbccdd", "bb", "xx")	Replace('aabbccdd', 'bb', 'xx')	Replace all 'bb' in the original string with 'xx'
Right("ABCDE",2)	Right('ABCDE', 2)	Right characters of a string Right('ABCDE', 2) returns DE
Rtrim(x)	Rtrim(x)	Trim the spaces off the Right of a string Rtrim("SQL ") returns "SQL"
Space(X)	Space(X)	Give you X number of spaces e.g. Select Space(22) + 'aabbccdd'
Str(X)	Str(X)	Converts a number to a string Str(1234) returns "1234"
	Stuff(X,12,4,Y)	Stuff('Now is the time', 12, 4, 'Place') Returns - Now is the Place
Ucase(x)	Upper(x)	Change to UPPER case
Val("123 Main")	Nothing available in SQL Server	Val() returns a numeric value at the beginning of a string Val("123 Main") returns 123 Val("Main 123") returns 0
VBCRLF	Char(13)	Line Feed

* See *figure 18-10* and *18-11* for built in functions used in Access that have a similar function in SQL Server. You will want to use these functions in Stored procedures. In some cases, you can use the function in a view.

Figure 18-11 **>> Date/Time Functions <<**

Access	SQL	Explanation
Now()	Getdate()	SQL Server returns 2001-05-24 10:37:09.043 Access returns 5/24/2001 10:37:09 AM
Date()	Getdate()	Date() in Access returns the date with no time. (Technically it returns the date with the time of midnight, but displays only the date) GetDate() Gets Date & Time - See "Style in Date Convert" below.
Style in Date Convert	Convert(*data_type* [(*length*)], *expression* [, *style*])	**In SQL Server** select date_Invoice, convert(varchar, date_Invoice, 1) as x from tbl_invoice Returns: 2001-04-12 00:00:00.000 4/12/01 2001-04-04 00:00:00.000 4/04/01 Style Date Style Date Style Date 1 4/12/01 101 4/12/2001 2 01.04.12 7 Apr 12, 01 107 Apr 12, 2001 0 Apr 12 2001 12:00AM select convert(varchar, getdate(), 8) returns hh:mm:ss 13:02:57
Format(expression, format) mmm-dd-yyyy = Apr-12-2001 mm-dd-yyyy = 4/12/2001 mm-dd-yy = 4/12/01	See "Style in Date Convert" above	**In Access** select date_Invoice, format(date_Invoice, "mmm-dd-yyyy") as x from tbl_invoice Returns: 4/12/2001 Apr-12-2001 4/4/2001 Apr-04-2001
DatePart("M",#5/22/99#)	DatePart(M, '5/22/99')	Get a part of a date - Note the Quotes in Access not SQL Select DatePart(M, '5/22/99') returns 5 ***Notice the quote marks in Access around "M" and not in SQL.***
DateAdd("M",2, #5/22/99#)	DateAdd(M, 2, '5/22/99')	Does Date addition and subtraction DateAdd(interval, number, date) Interval - see the constants below The number can be a positive or negative number
DateDiff("M", #1/1/00#, date())	DateDiff(M, pubdate, getdate())	Get the difference between 2 dates DateDiff(interval, number, date) Interval - see the constants below select date_Invoice, DATEDIFF(d, date_Invoice, getdate()) as x from tbl_invoice
Constants used in date functions		
q	q, qq	Quarter
m	m, mm	Month
y	y, dy	Day of Year
d	d, dd	Day
ww	ww, wk	Week
	dw	WeekDay
h	hh	Hour
n	mi, n	Minute
s	s, ss	Second
	ms	millisecond
yyyy	yy, yyyy	Year
	dw	Day of week (SQL Server considers Sunday as first day of the week) DATEPART(dw, GETDATE())

Figure 18-xxx	>> SQL Server Internal Variables <<
SQL	**Explanation**
@@DATEFIRST	Returns the current value of the SET DATEFIRST parameter, which indicates the specified first day of each week: 1 for Monday, 2 for Tuesday, and so on through 7 for Sunday. Sunday is the default first day of the week for SQL Server select DATEPART(dw, '3/29/03') returns 7 ('3/29/03' is a Saturday) You can reset the first day of week -- set first day of week to Friday --> SET DATEFIRST 5
@@ERROR	Returns the error number for the last Transact-SQL statement executed. Must be used before the next line of code is executed. You might want to select @intRecords= @@rowcount, @intError=@@Error
@@IDENTITY	Returns the last-inserted identity value. INSERT INTO jobs (job_desc,min_lvl,max_lvl) VALUES ('Accountant',12,125) SELECT @@IDENTITY AS 'Identity'
@@MAX_CONNECTIONS	Returns the maximum number of simultaneous user connections allowed on a Microsoft® SQL Server™. The number returned is not necessarily the number currently configured. SELECT @@MAX_CONNECTIONS
@@NESTLEVEL	Returns the nesting level of the current stored procedure execution (initially 0).
@@OPTIONS	Returns information about current SET options.
@@PROCID	Returns the stored procedure identifier (ID) of the current procedure.
@@ROWCOUNT	Returns the number of rows affected by the last statement. select @intRecords= @@rowcount, @intError=@@Error
@@SERVERNAME	Returns the name of the local server running Microsoft® SQL Server™.
@@TEXTSIZE	Returns the Maximum length, in bytes, of text or image data that a SELECT statement returns.
@@TOTAL_ERRORS	Returns the number of disk read/write errors encountered by Microsoft® SQL Server™ since last started
@@TOTAL_READ	Returns the number of disk reads (not cache reads) by Microsoft® SQL Server™ since last started SELECT @@TOTAL_READ AS 'Reads', @@TOTAL_WRITE AS 'Writes', GETDATE() AS 'As of'
@@TOTAL_WRITE	Returns the number of disk writes by Microsoft® SQL Server™ since last started.
@@TRANCOUNT	Returns the number of active transactions for the current connection.
@@VERSION	Returns the date, version, and processor type for the current installation of Microsoft® SQL Server --> SELECT @@VERSION

#19 - ADO - Here is a note, however *This Could be a White Paper in Itself*

* ADO is a great way to read tables direct. If you need to write to the tables, it works great too. If you are using a SQL View (with multiple tables), the only way I have found to write data is to use the ADO Shape Command.

Conclusion

This is just a few of the things I have learned. If there is an interest in sharing what we know about Access 2000 to SQL 7, I would be delighted to share my code and participate in the learning experience.