Database Structure

Training & Setup Guide

Last Updated: January 2008

About this Guide

This SedonaOffice Database Structure Training Guide is for use by SedonaOffice customers only.   This guide is to be used in conjunction with an approved training class provided by SedonaOffice, and is not meant to serve as an operating or setup manual.

This training and setup guide is for experienced SedonaOffice users who have knowledge of the database setup.   While this guide will review some of the basic setup necessary, this guide is not intended to teach Database Structure basics and assumes the user has knowledge of SQL and of the SedonaOffice application.

SedonaOffice reserves the right to modify the SedonaOffice product described in this guide at any time and without notice.   Information in this guide is subject to change without notice.   Companies, names and data used in examples herein are fictitious unless otherwise noted.   In no event shall SedonaOffice be held liable for any incidental, indirect, special, or consequential damages arising out of or related to this guide or the information contained herein.   The information contained in this document is the property of SedonaOffice.

This guide will be updated periodically, be sure to check our website at www.sedonaoffice.com for the most current version.

Copyright 2008

Table of Contents

Overview

This guide is intended to teach you how to access data from a SedonaOffice database. Data extracted from a database can be used for many different purposes both internally and externally for an organization. While this guide will review a variety of different techniques, it is impractical to detail each and every type method that can be used to extract data.

Each Company is a Database

Each SedonaOffice company is its own unique database within the SQL server. In addition to the various company databases, there is an additional database that helps to controls access to the company databases. This access control database is named SedonaMaster.

SedonaMaster contains list of company names and the database associated with each name. All other data about a company is contained within the company database. All of the setup information, names, addresses, part numbers, service tickets, etc., for a company, are stored within the same database. The structure of the database will remain the same for all companies. The differences in how companies operate are contained in the setup tables. If a feature of SedonaOffice is not used, the data structure will still exist but may be empty of data.

Databases Contain Tables, Views, and Stored Procedures

The main structures in a database are tables, views and stored procedures. Tables contain the raw data, the actual names, addresses, etc.

Views are premade queries that will return sets of data automatically. If there is a set of data you are going to regularly extract, you may want to think about making a view. SedonaOffice uses several views in supplying data to the client. Do NOT alter these or your system may cease to operate correctly[1].

Stored procedures are routines containing SQL code. They can be created to act as a view but are usually used to manipulate the data. Stored procedures also can take parameters, values that modify how the stored procedure will operate. Most of the business logic in SedonaOffice is handled by stored procedures. They are encrypted and locked for safety and security. Do NOT delete or replace a stored procedure or your system will cease to operate correctly[2].

Tables Contain Fields

Fields contain your actual data. They are different types:

- Text including varchar, nvarchar and char. The length of the field in characters (including spaces and punctuation) is defined when the field is created.
- Numeric including integer, double and money. What range and if a fractional decimal amount is supported is defined when the field is created.
- Datetime. Microsoft SQL server does not contain a field type for date or one for time, all date and time related fields are Datetime fields.

---

[1] Unless directed to by a SedonaOffice support person.
[2] Unless directed to by a SedonaOffice support person.

Linking Tables

Tables are linked via fields that end in Id. In each table, the first field is the Identity field for that table. Identity fields are not editable nor should you try. Identity fields are unique. This number is automatically assigned by the SQL server. Once assigned, a number is never reused, not even if it was previously deleted. This Identity field is the "Address" of the record. Other tables that point to this table will have an Id that matches the "Address" of the record. IE Customer_Id in the AR_Customer_Bill record will point to the Customer_Id field in the AR_Customer table. The Customer_Id in the AR_Customer table is the Identity or "Address" of that record. Notice that the name of an Id is the same as the table name in our example. This is true of all ID's with very few exceptions.

Link Types

Table links are defined by the relationship of records in one table to the records in another table. There are three basic link models.

One to one: Each record in one table matches to exactly one record in the other table. IE AR_Customer and AR_Customer_Userdef

One to many: Each record in one table matches to many records in the other table. IE AR_Invoice and AR_Invoice_Item

Many to one: Many records in one table match to one record in the other table. IE AR_Customer and AR_Branch

Customer Structure

Invoice Structure

```
AR_Customer_          AR_Customer          AR_Customer_Bill

                      AR_Invoice    AR_Category   GL_Account   AR_Term   GL_Register   AR_Invoice_Description

                                    OE_Job   SV_Service_Ticket   IN_Warehouse   AR_Taxing_Group

AR_Deposit_Check_Detail    AR_Invoice_Item           AR_Item    IN_Part

AR_Advance_Deposit_Detail    AR_ACH

                                    GL_Deferred_Income

AR_Unapplied_Cash_Detail    AR_SalesTax

                                    IN_Journal

AR_Credit_Detail    AR_Cancel_Queue
```

Cash Structure

Vendor Structure

Vendor Bills Structure

AP_Invoice →	GL_Account	AP_Vendor	AR_Term	AR_Category
	AR_Branch	GL_Register	IN_Warehouse	OE_Job	SV_Service_Ticket

AP_Invoice_Parts →	IN_Unit_Of_Measure	OE_Job	GL_Register	IN_Journal
	IN_Part	AR_Customer	AP_Purchase_Order_Parts	IN_Repair_Order_Parts	SV_Service_Ticket

AP_Check_Invoices →	AP_Invoice	GL_Register

AP_Invoice_Expense →	AR_Customer	SV_Service_Ticket
	GL_Account	GL_Register	AR_Category	OE_Job

AP_Credit_Detail →	AR_Branch	GL_Register	GL_Account	AP_Credit

Check Structure

```
AP_Check

  GL_Account   AP_Vendor   SY_Employee   AR_Category

  GL_Register   AR_Unapplied_   AR_Credit   AR_Advanced_   AR_Branch
                Cash                        Deposit

  IN_Warehouse   OE_Job   SV_Service_   AR_Customer
                          Ticket


  AP_Check_      IN_Part   IN_Unit_Of_   AR_Customer   IN_Journal
  Parts                    Measure

                 SV_Service_   AP_Invoice_   GL_Register   OE_Job
                 Ticket        Parts

  AP_Check_
  Invoices       AP_Invoice   GL_Register   AP_Credit

  AP_Check_
  Credit_Detail

  AP_Check_      GL_Account   GL_Register   AR_Customer
  Expense
                 SV_Service_   OE_Job   AP_Invoice_   AR_Category
                 Ticket                 Expense
```

Job Structure

OE_Job				
AR_Customer	AR_Customer_Site	OE_Job_Status	OE_Job_Type	OE_Job
SY_Employee	SY_Department	SV_Service_Tech	OE_Install_Company	AR_Taxing_Group

AR_Credit

OE_Job_Task	OE_Job_Schedule	OE_Job_Log	OE_Job_TimeSheet	OE_Quote	AR_Customer_Equipment
OE_Job_System	OE_Job_Recurring	OE_Job_Parts	OE_Job_Issue	OE_Job_Commission	AR_Customer_Recurring
AR_RMR_Tracking	GL_Register	IN_Journal	IN_Requisition	IN_Requisition_Parts	AR_Customer_System
Ap_Check	AP_Check_Expense	AP_Check_Parts	AP_Credit	AP_Credit_Expense	AP_Credit_Parts
AP_Invoice	AP_Invoice_Expense	AP_Invoice_Parts	AP_Purchase_Order	AP_Purchase_Order_Expense	AR_Invoice

Service Ticket Structure

Inventory structure

```
                    AR_Item      IN_Product_     IN_Unit_Of_     AP_Vendor     IN_Manufacturer
                                    Line           Measure

   IN_Part

                    AP_Credit_    AP_Check_      AP_Invoice_     AP_Purchase_
                      Parts         Parts          Parts        Orders_Parts

                    AR_Credit_    AR_Customer_   AR_Invoice_     CS_Customer_
                       Item        Equipment        Item       System_Access_
                                                                    Card

                    IN_Alternate_ IN_Cost_Layer  IN_Inventory    IN_Journal
                       Part

                    IN_Part_      IN_Part_Unit   IN_Physical_    IN_Repair_
                     Supplier                       Part        Order_Parts

                    OE_Job_       OE_Pricebook_  OE_Quote_       OE_Sales_      OE_Sales_
                      Parts          Part          Part        Package_Part    Part_Part

                    SV_Service_   SY_Package_
                    Ticket_Parts    Parts
```

General Ledger Structure

```
GL_Register

AR_Customer    SS_Register_     AP_Vendor    SY_Employee    AR_Branch    GL_Register
               Type

OE_Job         GL_Accounting_   GL_Account   SV_Service_    AR_Category
               Period                        Ticket

               AR_Customer      AR_Customer_  OE_Job_
                                Site          Status

               OE_Job_Type      SY_Employee   SY_Department
```

—

—
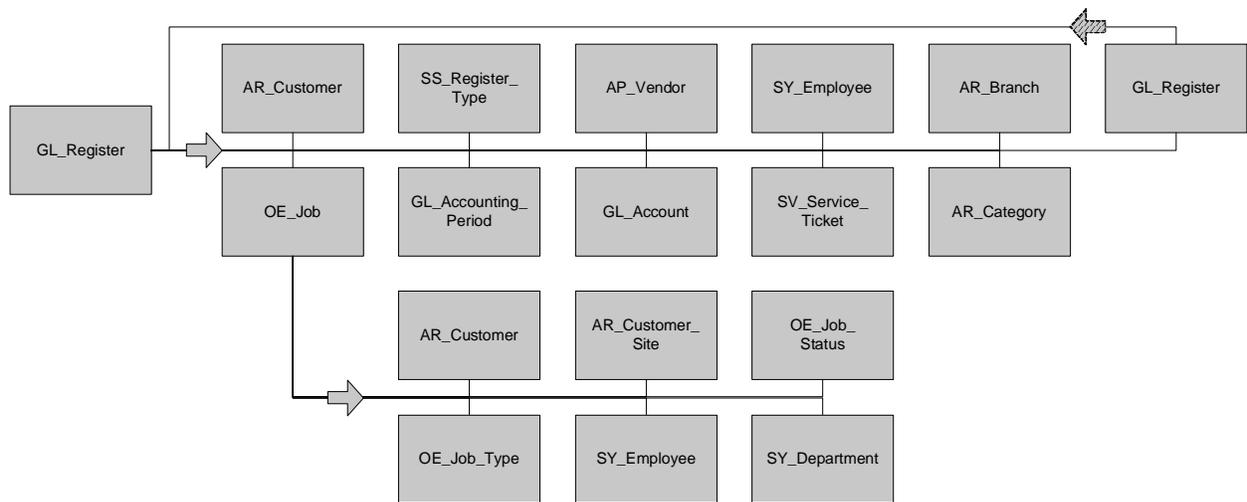
SedonaOffice Database Structure Training & Setup Guide

Open Database Connectivity (ODBC)

Open Database Connectivity is the methodology created by Microsoft for different applications to talk to different kinds of databases. With ODBC you can connect Excel to Microsoft SQL server or MS Word to Excel for example. The first step in connecting any application to your Microsoft SQL database is to create an ODBC connection. There is a utility for setting up ODBC connections. It is located in the Control Panel under ODBC. Many applications though contain an implementation of the ODBC Data Source Administrator. In our example we are going to use Excel to create an ODBC connection.

Creating an ODBC Connection with Excel

Let's now review how to import Data into Microsoft Excel.   In this example we are going to use the feature in Excel to Query an External Data Source using Microsoft Query.   This feature is available in most recent versions of Excel.
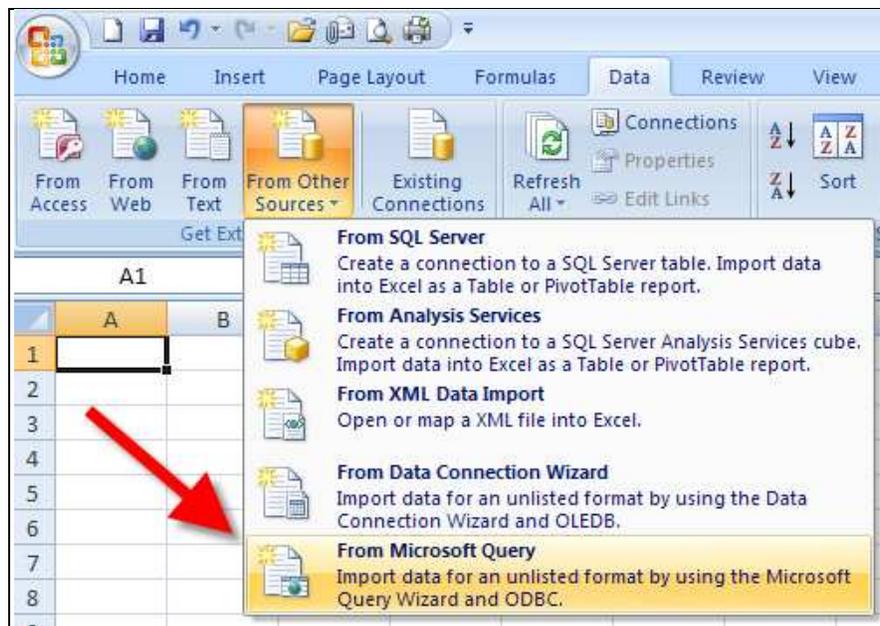
If you have not already done so, you will need to create a Data Source connection to your SedonaOffice database.
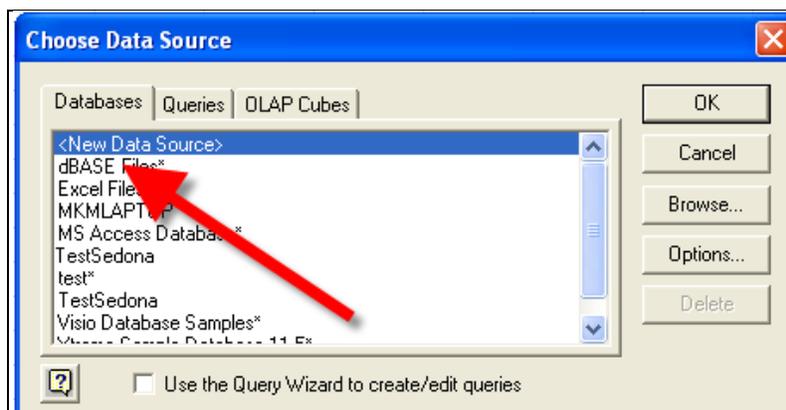
To create the new Data Source:

1) Name the data source appropriately (Here we are using "SedonaOffice GL Data" but the same connection can be used for all of your queries so you might want a more general name.)
2) Select 'SQL Server' as the driver to connect to the database
3) Press the Connect button
    a. On the SQL Server Login Screen select the name of the SQL Server for SedonaOffice
    b. Use "SedonaReports" as the Login ID, no password is needed
    c. Select the Options tab and select the name of your production SedonaOffice database
4) Press OK
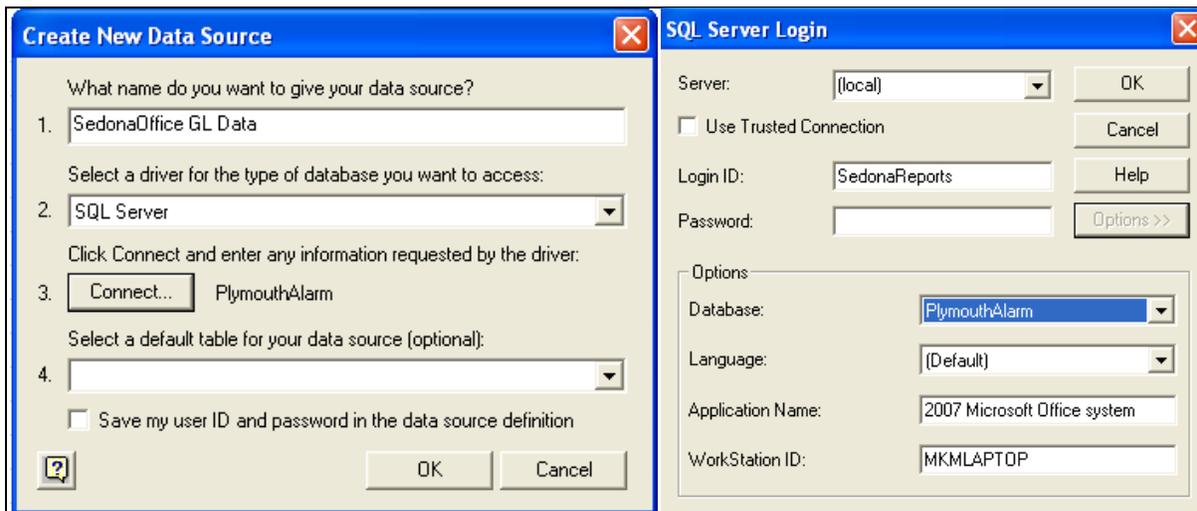
You now have an ODBC connection to your database.

## Building a Query Using Excel and MS Query

Select the data source you have previously created to create the Query.   Uncheck the 'Use the Query Wizard..." this will take you directly to Microsoft Query to create the Query.
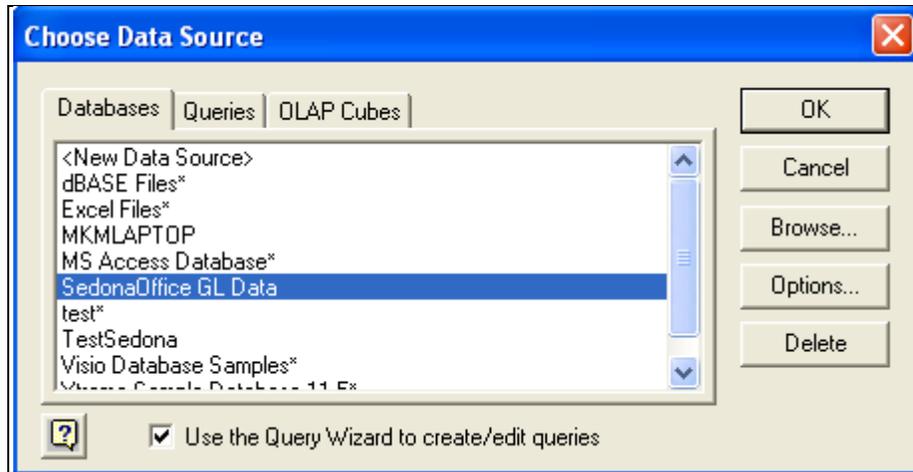
To begin with you need to select the Table file to use in the Query.   Select the "SO_Complete_GL_Total_YTD" table.   Then click Close.
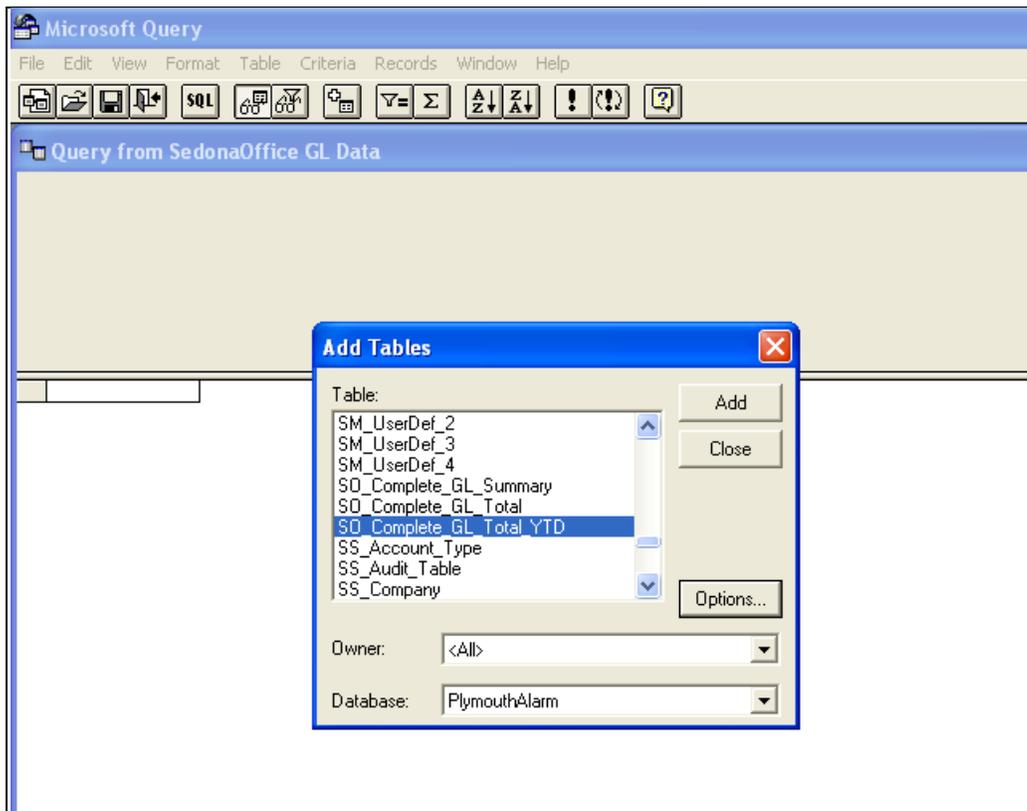
The next step is to select the data fields and criteria for the data to be returned. Select all the data elements in the Table. While it doesn't really matter what order to display the data fields, using the order as shown below will be more logical when viewed with Excel.
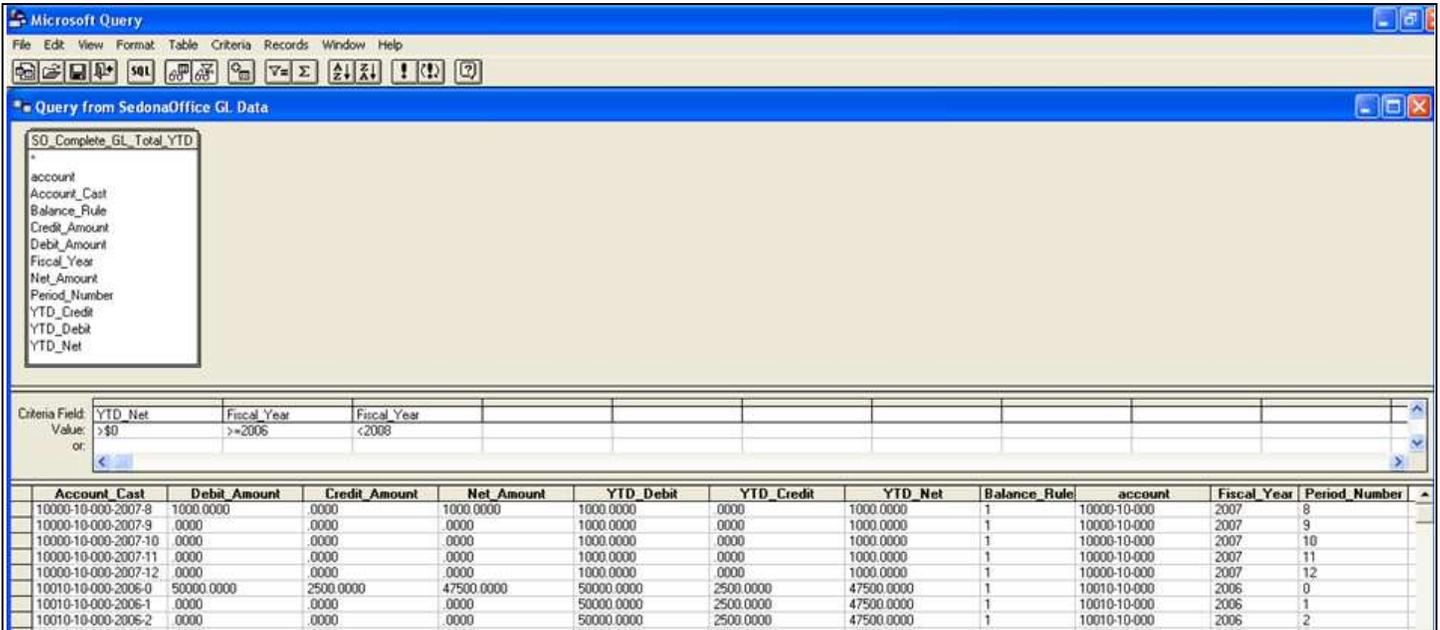
Since this table can contain thousands (hundreds of thousands of records) it is best to use some criteria to limit the data that returns.

Criteria Selections:

1) YTD_Net <> $0 – By selecting this option only data with values will be returned.
2) Fiscal Year >= 2006 – In this case only years 2006 and 2007 are needed so limit the data to only these fiscal years.
3) Fiscal Year < 2008 – In this case since 2008 has been created we can remove these entries since were still reporting on 2007.
4) Net_Amount <>$0 – This is included as an 'OR' selection. This is necessary to return the Retained Earnings account (more on this later).

Criteria Field:	YTD_Net	Fiscal_Year	Net_Amount	
Value:	<>$0	>=2006		
or:	$0	>=2006	<>$0	

Now that we have completed the Query, click the Return Data icon, and the GL Data will be returned to Excel.
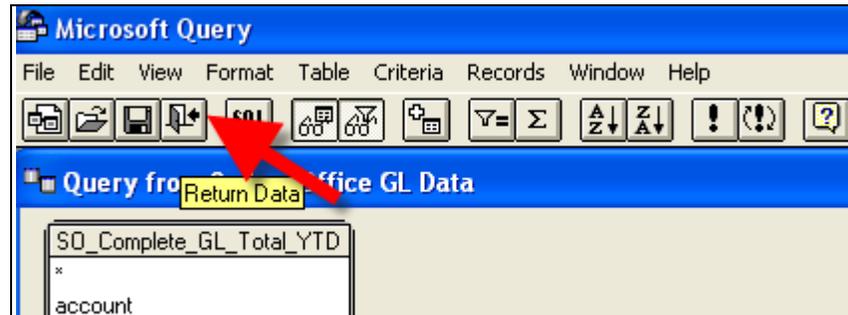
Your data will be returned to Excel.

	A	B	C	D	E	F	G	H	I	J	K
1	Account_Cast	Debit_Amount	Credit_Amount	Net_Amount	YTD_Debit	YTD_Credit	YTD_Net	Balance_Rule	account	Fiscal_Year	Period_Number
29	10010-10-000-2007-9	0	19744	-19744	50100	22364	27736	1	10010-10-000	2007	9
30	10010-10-000-2007-10	0	0	0	50100	22364	27736	1	10010-10-000	2007	10
31	10010-10-000-2007-11	0	0	0	50100	22364	27736	1	10010-10-000	2007	11
32	10010-10-000-2007-12	750000	0	750000	800100	22364	777736	1	10010-10-000	2007	12
33	10010-20-000-2007-9	750000	980	749020	752425	5325	747100	1	10010-20-000	2007	9
34	10010-20-000-2007-10	0	0	0	752425	5325	747100	1	10010-20-000	2007	10

Using Microsoft Access to Review Your Data

Why use Access instead of Excel? Excel has a row limit. The maximum number of rows you can have in a spreadsheet varies with what version you use from as little as 32,767 rows for older versions to 1,048,576 rows in Excel 2007. This may seem like a lot of rows and for most queries it will be sufficient. But, queries involving the GL_Register for a company that has several years of history can easily exceed these limitations.

Excel treats all fields containing only numeric characters (0-9) as numbers unless prefaced with a ' character. By treating things like postal codes as numbers postal codes starting with a 0 are truncated. Thus a postal code of 01234 becomes 1234.

Finally, Access has a built in report generator. With Access you can make complex reports with groups, subtotals, totals, etc.

*** Caution *** ONLY use SedonaReports for an ODBC connection to Access. Otherwise changes you make in Access can change your SQL Server data and corrupt your database.

Connecting Access via ODBC

When using an ODBC connection with Access you have two options on how to connect the data, Import or Link.

When you Import data into Access, you create a copy of the data stored within the Access database. This allows you to review the data when not connected to the database. Like Excel, you have to periodically refresh the data to keep it up to date.

When you Link data to Access, the data remains in the SQL Server but Access can use it in queries and reports. This method is constantly refreshes as the data in the SQL server changes but it will not function if it is disconnected from the SQL Server.

Choose the External Data tab. Then choose More. Finally choose ODBC Database.
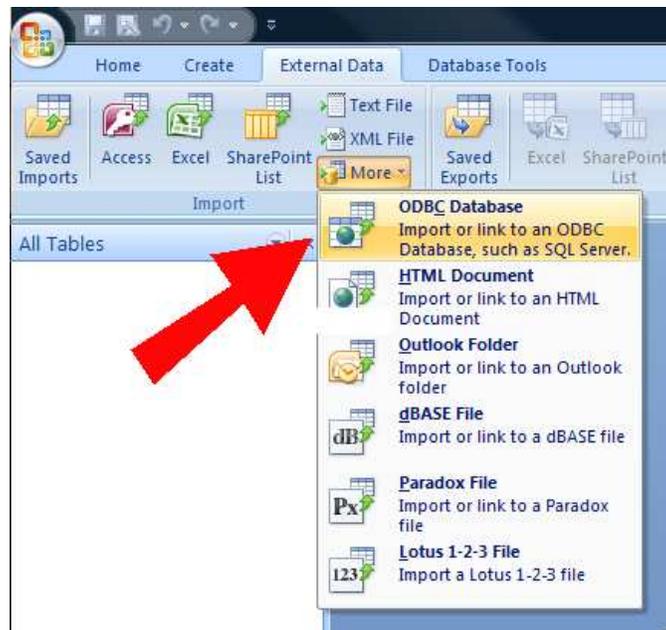
Choose Import or Link and then click OK.
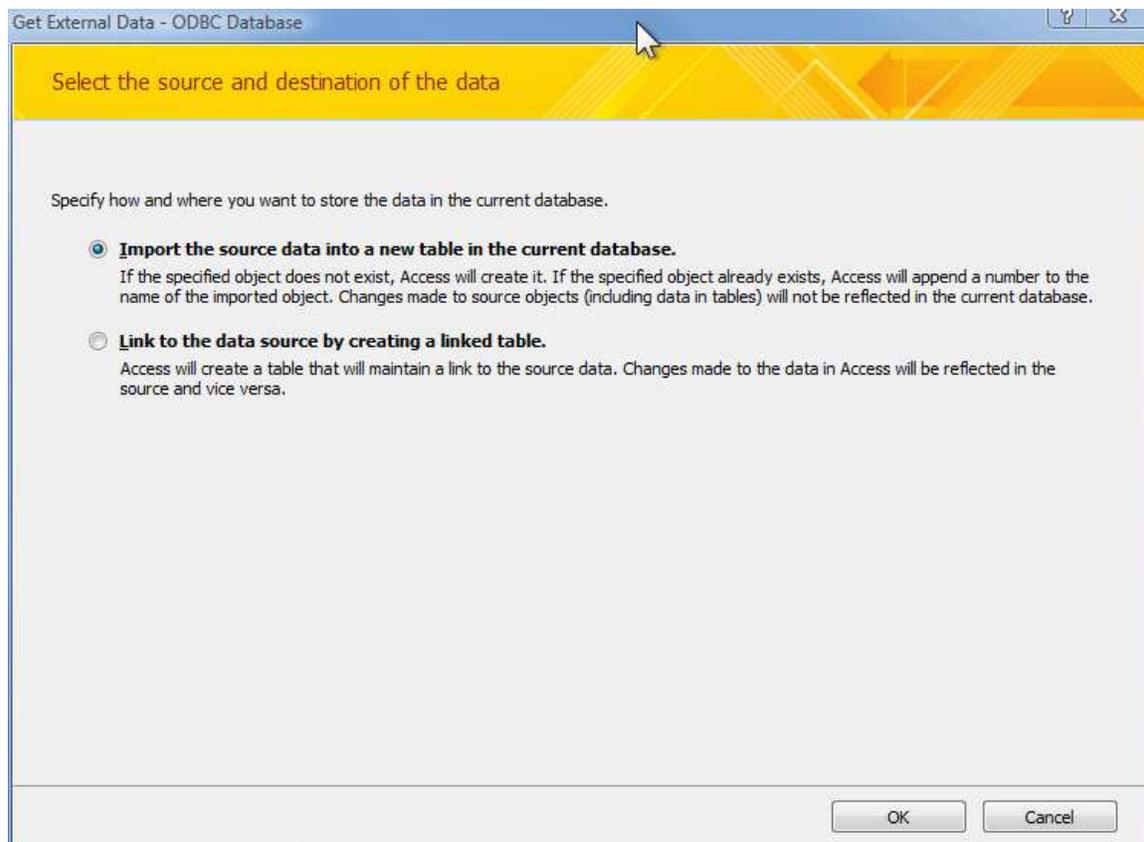
Choose your Data Source.
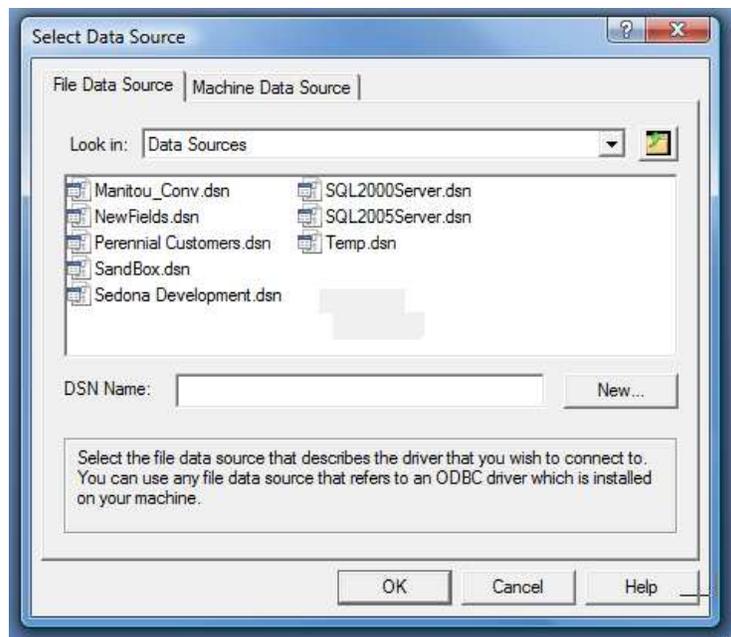
Then choose the tables you wish to Import or Link and click OK. You can choose multiple tables but do not select all. Access is not as large or as powerful as SQL Server. Choosing all will probably crash Access.
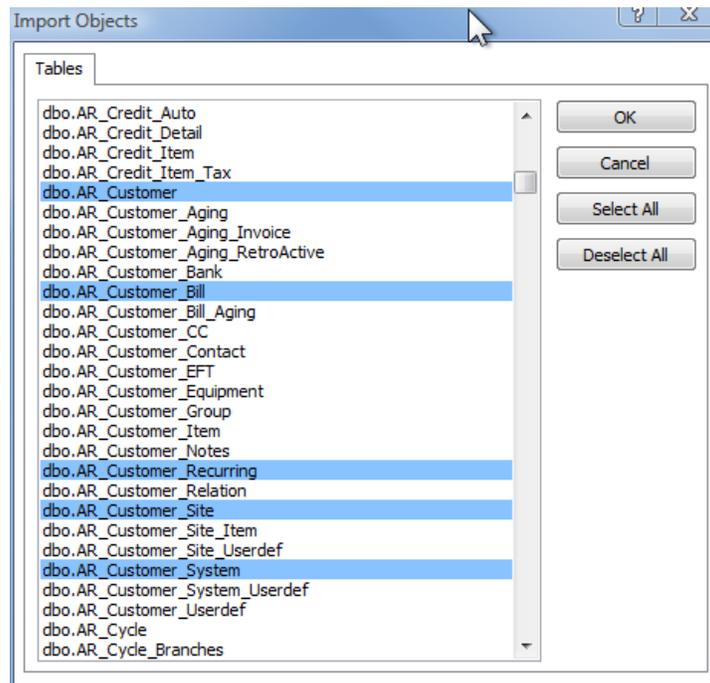
If you chose to Link you will be asked to Select Unique Record Indicator. This is always the top item in SedonaOffice.

If you chose Import, when the operation is complete a window will be displayed showing the success of the operation. Here you can also Save the steps you just did so refreshing the data will be easier.
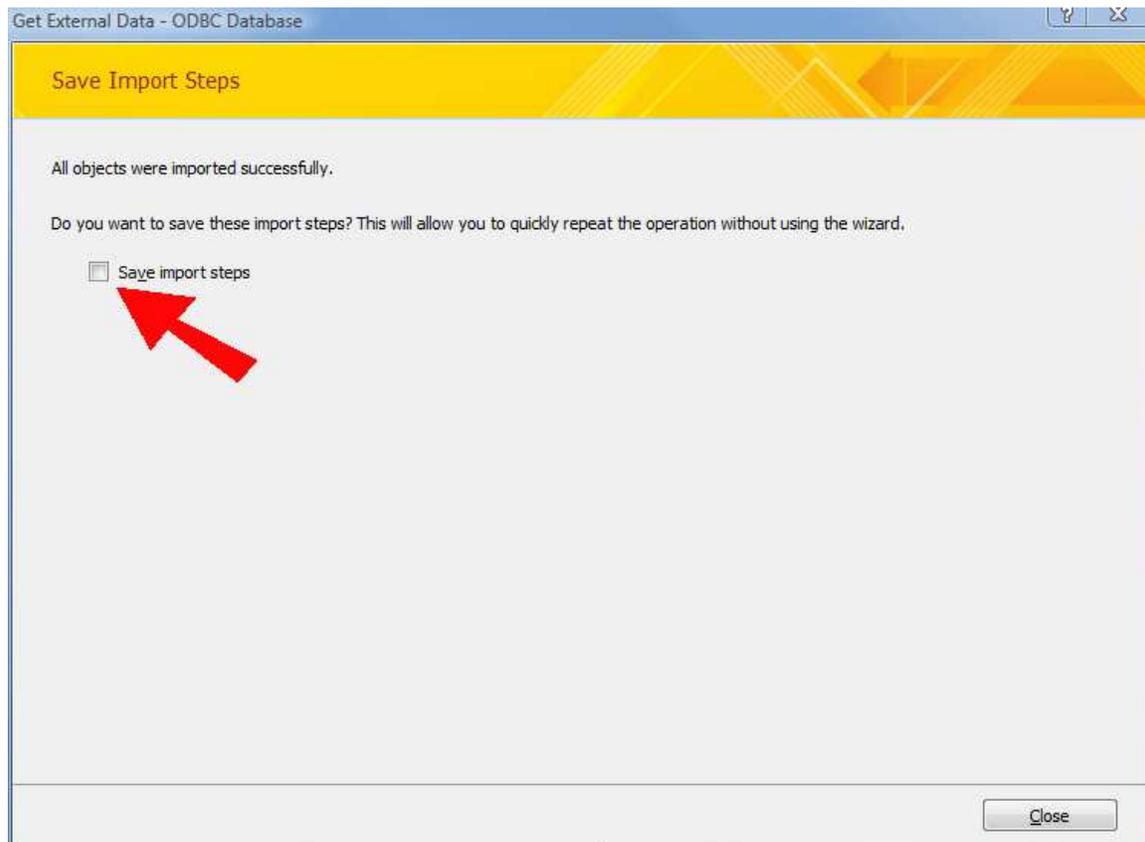
Your tables will then be accessible in Access. You may mix Import and Link in the same Access database. In the example I have Imported several customer tables and linked the branch table. Notice the different icons for imported versus linked tables. The highlighted table is the linked branch table.
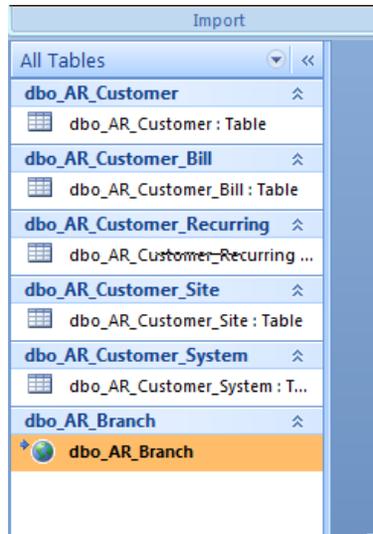
Writing a query with Access

Click on the Create tab and then on Query design.

Choose the tables you wish to include in your query. A table can be selected more than once if you need to join it to more than one Id. For our example we are going to choose all of the tables.
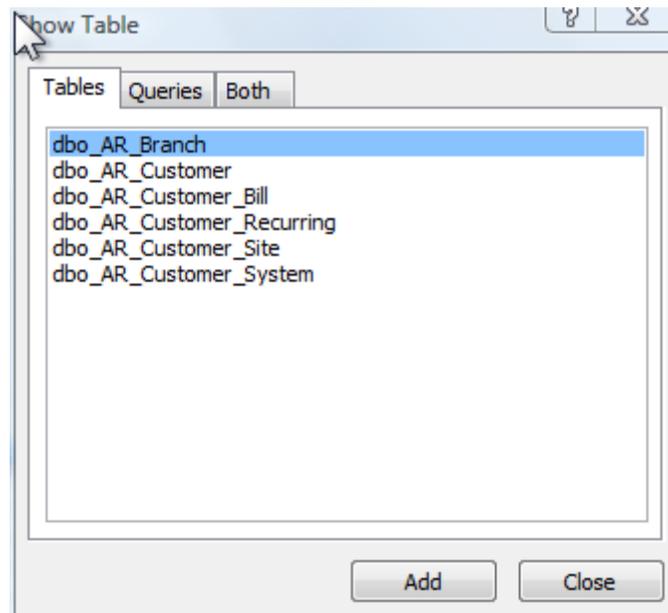

Delete all of the joins that access automatically creates.
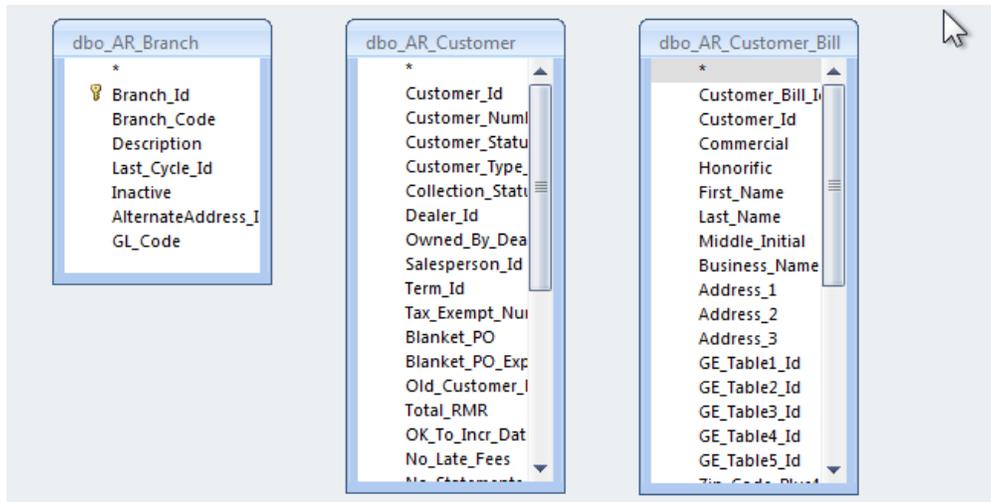
Create the joins according to the structure of SedonaOffice. In this case AR_Branch.Branch_Id to AR_Customer.Branch_Id and AR_Customer.Customer_Id to AR_Customer_Bill.Customer_id.

We are going to create a mailing list so we need to drag the name and address information to the lower pane. We are also going to drag down the branch code so we can sort on branch.

Informatio n we don't want to
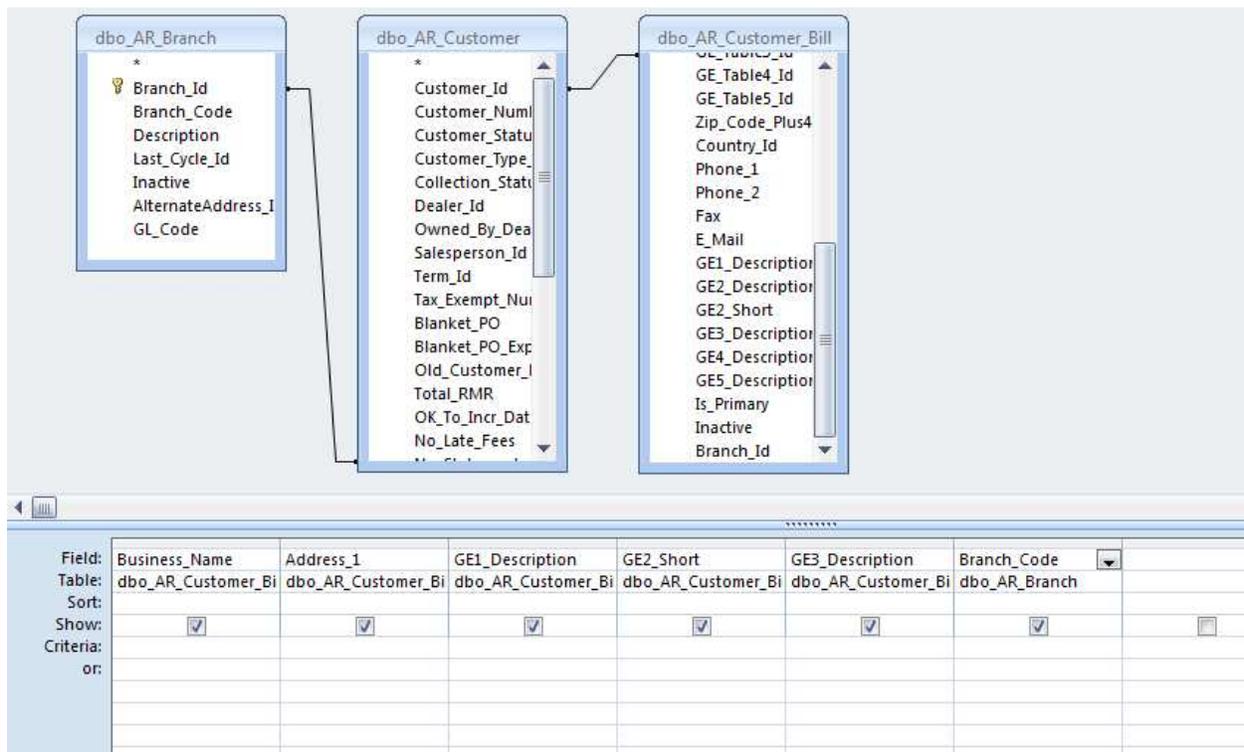
Address_1 may not be all of the address information needed but if there is no address_2 we don't want to add a blank line. So we create a formula. Click in the Address_1 cell and then click 0n the formula button.
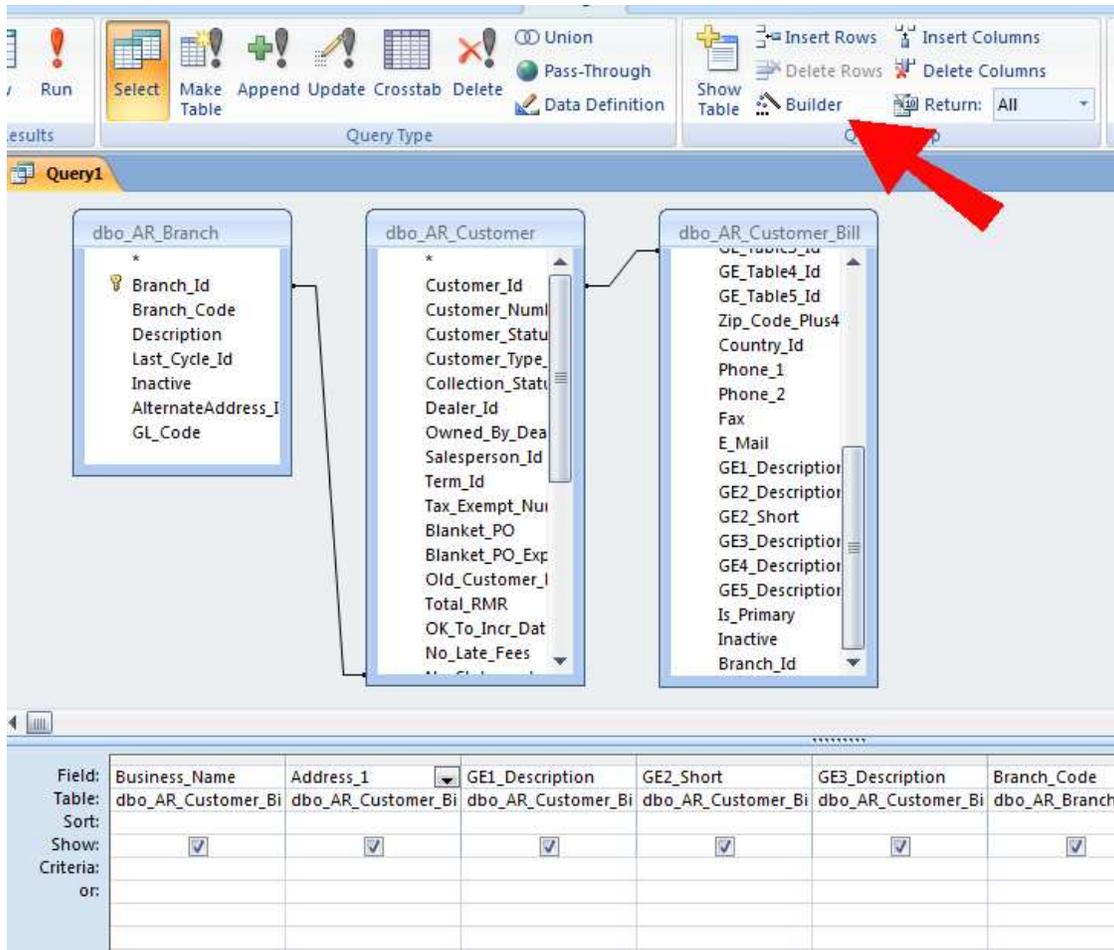
Enter the following into the builder window.

Address: [dbo_AR_Customer_Bill]![Address_1] & IIf([dbo_AR_Customer_Bill]![Address_2]=
"","",Chr$(13) & Chr$(10) & [dbo_AR_Customer_Bill]![Address_2]) &
IIf([dbo_AR_Customer_Bill]![Address_3]= "","",Chr$(13) & Chr$(10) &
[dbo_AR_Customer_Bill]![Address_3])
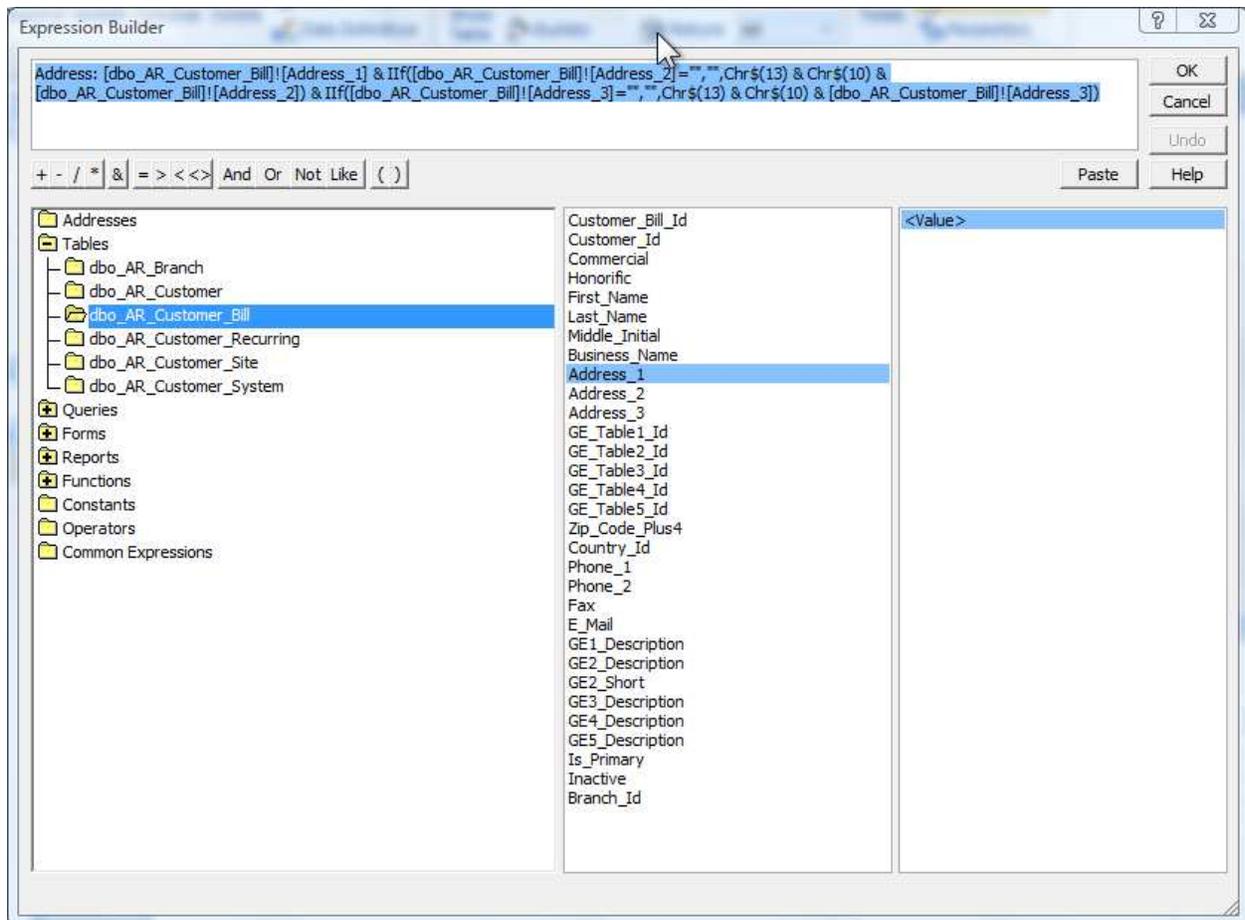
Click view to test our results.
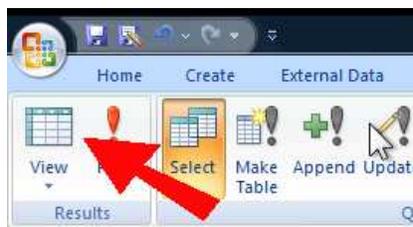
| Business_Name	Address	GE1_Description	GE2_Short	GE3_Descrip	Branch_Cod
N/A		N/A	N/A	N/A	Main Division
George Washington	1234 Mount Vernon Lane	Colorado Springs	CO	80919	Major
John Adams	5411 2nd Street	Colorado Springs	CO	80919	Major
Mega Mart	2154 Mountain Springs Road	Colorado Springs	CO	80919	Major
Mega Mart #200	7415 Union Blvd	Colorado Springs	CO	80919	Major
John Wayne	4521 Mountain View Terrace Apartment # 315	Colorado Springs	CO	80919	Major
Roy Rogers	7411 Bullet Lane	Colorado Springs	CO	80919	Major
Gene Autry	7466 Carter Vall. Road	Colorado Springs	CO	80919	Major
Clint Eastwood	12445 Happy Acres Drive	Colorado Springs	CO	80919	Major
Andrew Marriott	123 Main Street	Colorado Springs	CO	80919	Major
Rocky Mountain High School	421 Falcon Way	Colorado Springs	CO	80919	Major
TELUS	1234 Main Street	Colorado Springs	CO	80919	Major
Win-Pak	421 Windchime Pl	Colorado Springs	CO	80919	Major
Dealer X	1234 Main Street	Colorado Springs	CO	80919	Major

Now, let's remove the N/A row and add a method to select which branch we want.

Under business_Name add <>"N/A". Then under Branch_Code add =[Select Branch].

Business_Name dbo_AR_Customer_Bi	Address: [dbo_AR_Cu	GE1_Description dbo_AR_Customer_Bi	GE2_Short dbo_AR_Customer_Bi	GE3_Description dbo_AR_Customer_Bi	Branch_Code dbo_AR_Branch	
✓	✓	✓	✓	✓	✓	
< > "N/A"					=[Select Branch]	

Now when we return the results we are asked to select a Branch.

Enter Parameter Value

Select Branch

[ ]

OK   Cancel
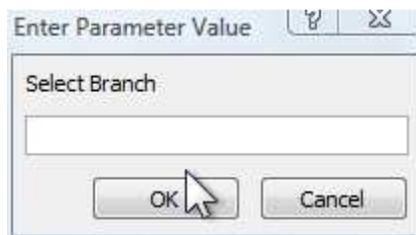
Entering a branch we get results with no N/A.

| Business_Name	Address	GE1_Description	GE2_Short	GE3_Descrip	Branch_Cod
TELUS	1234 Main Street	Colorado Springs	CO	80919	Major
Dealer X	1234 Main Street	Colorado Springs	CO	80919	Major
George Washington	1234 Mount Vernon Lane	Colorado Springs	CO	80919	Major
John Adams	5411 2nd Street	Colorado Springs	CO	80919	Major
Mega Mart	2154 Mountain Springs Road	Colorado Springs	CO	80919	Major
Mega Mart #200	7415 Union Blvd	Colorado Springs	CO	80919	Major
John Wayne	4521 Mountain View Terrace Apartment # 315	Colorado Springs	CO	80919	Major
Roy Rogers	7411 Bullet Lane	Colorado Springs	CO	80919	Major
Gene Autry	7466 Carter Valley Road	Colorado Springs	CO	80919	Major
Clint Eastwood	12445 Happy Acres Drive	Colorado Springs	CO	80919	Major
Andrew Marriott	123 Main Street	Colorado Springs	CO	80919	Major
Rocky Mountain High School	421 Falcon Way	Colorado Springs	CO	80919	Major
Win-Pak	421 Windchime Pl	Colorado Springs	CO	80919	Major

Creating a Report with Access

Displaying the results on the screen is useful but Access allows us to create reports. The report we are going to create will be to create mailing labels.

First make sure the new query you created is selected and then launch the label wizard.
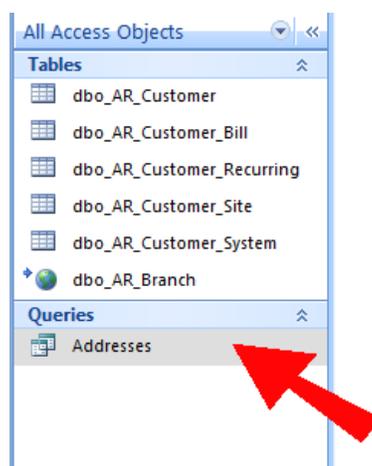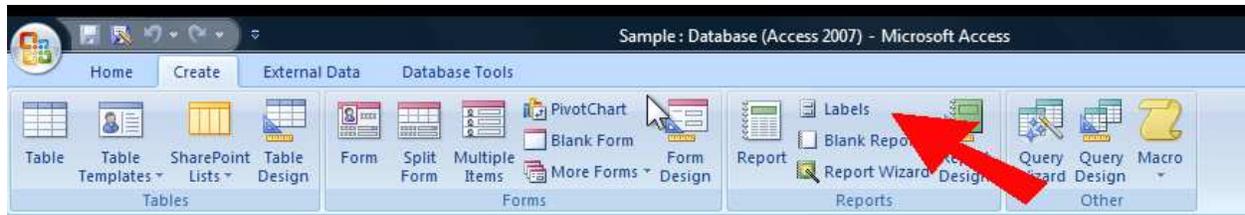
Choose your label. You can choose by the form number if you bought labels from a major manufacturer or just choose a label of the same size as the ones you are using.
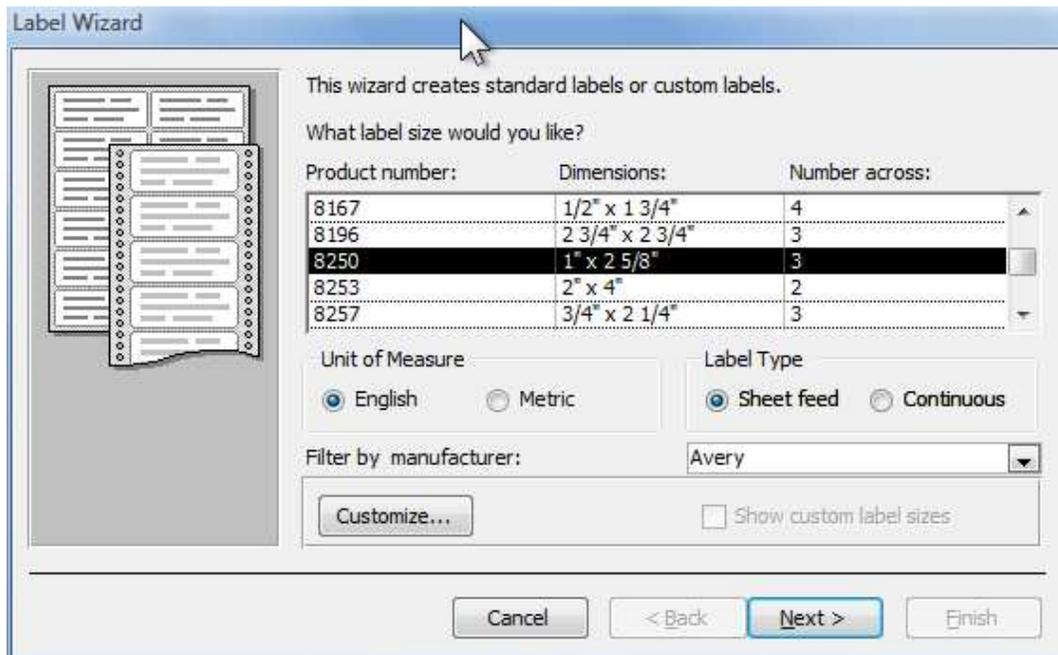

Choose your font.
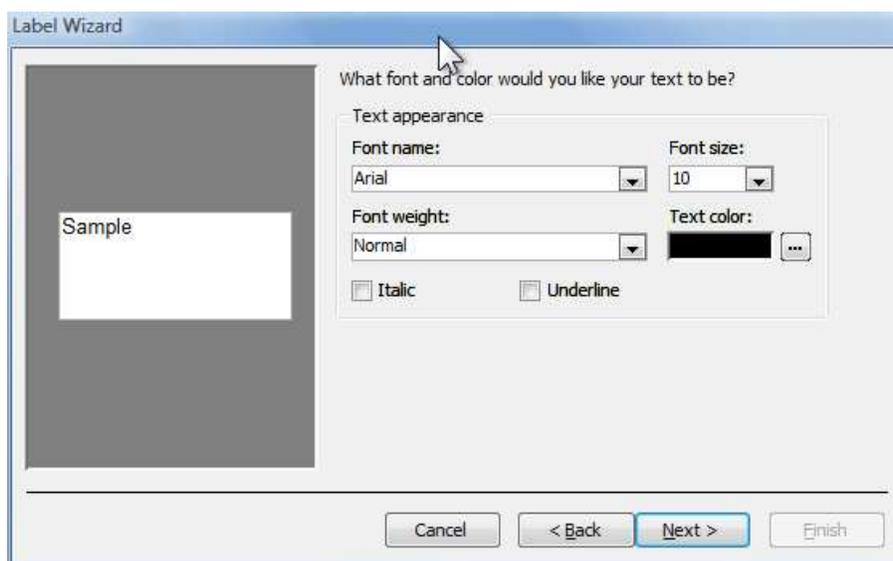
Setup the fields how you want them to appear on the label.
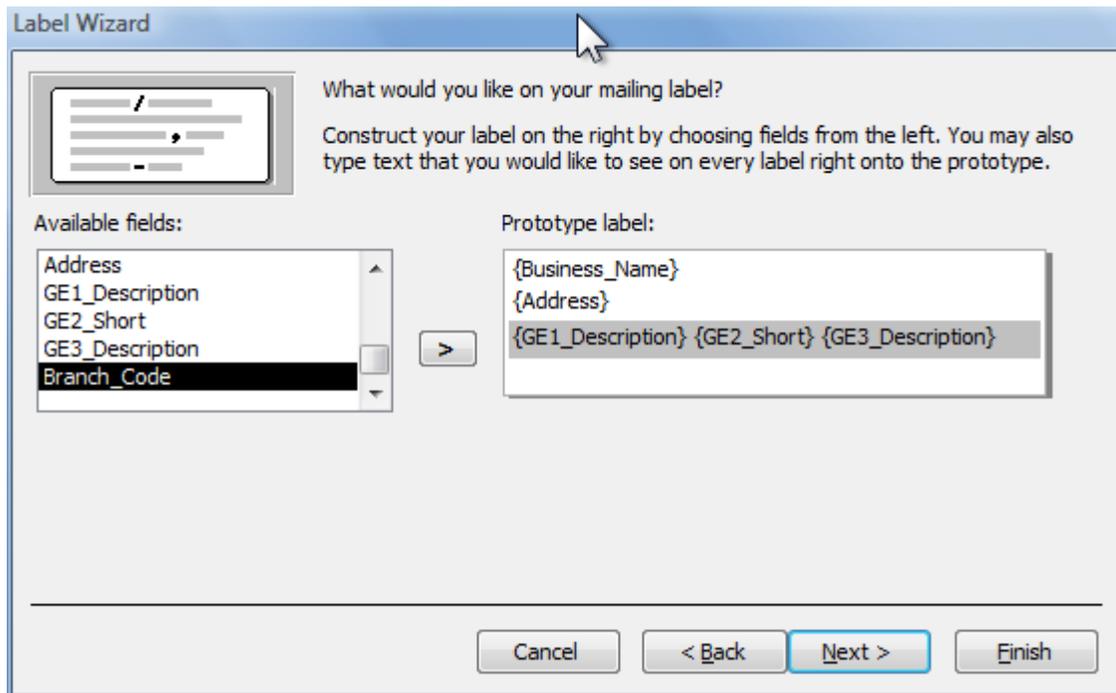
Select any fields you want to sort on. Here I've selected the GE3_Description so we can get a presorted discount from the post office.
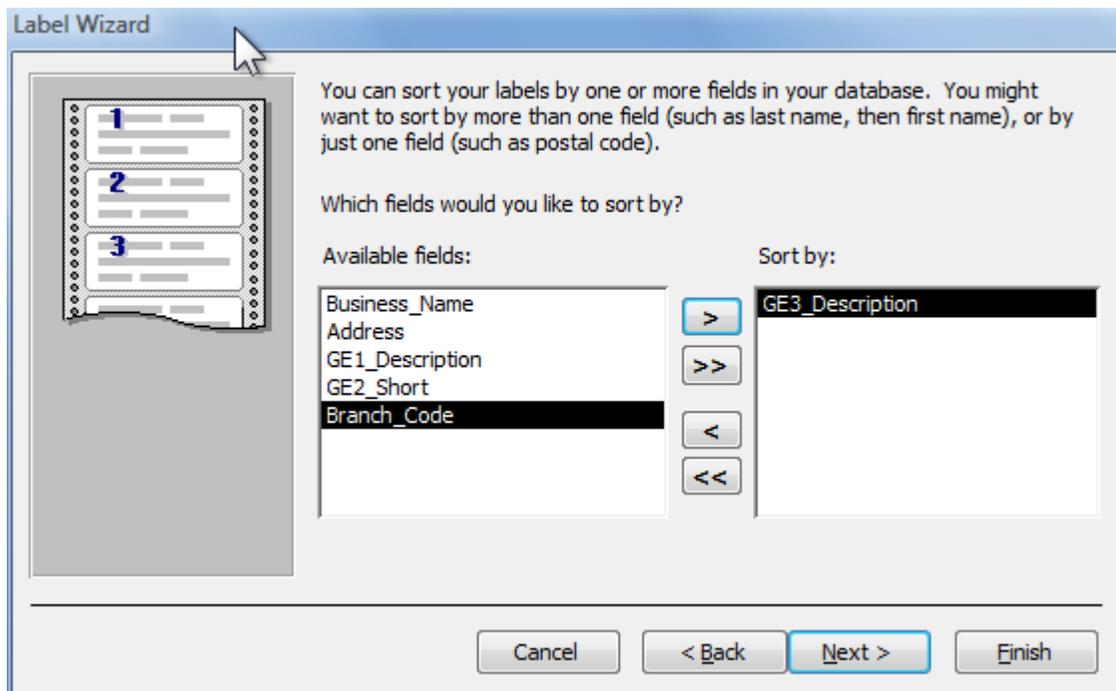
Give your report a name and save it.
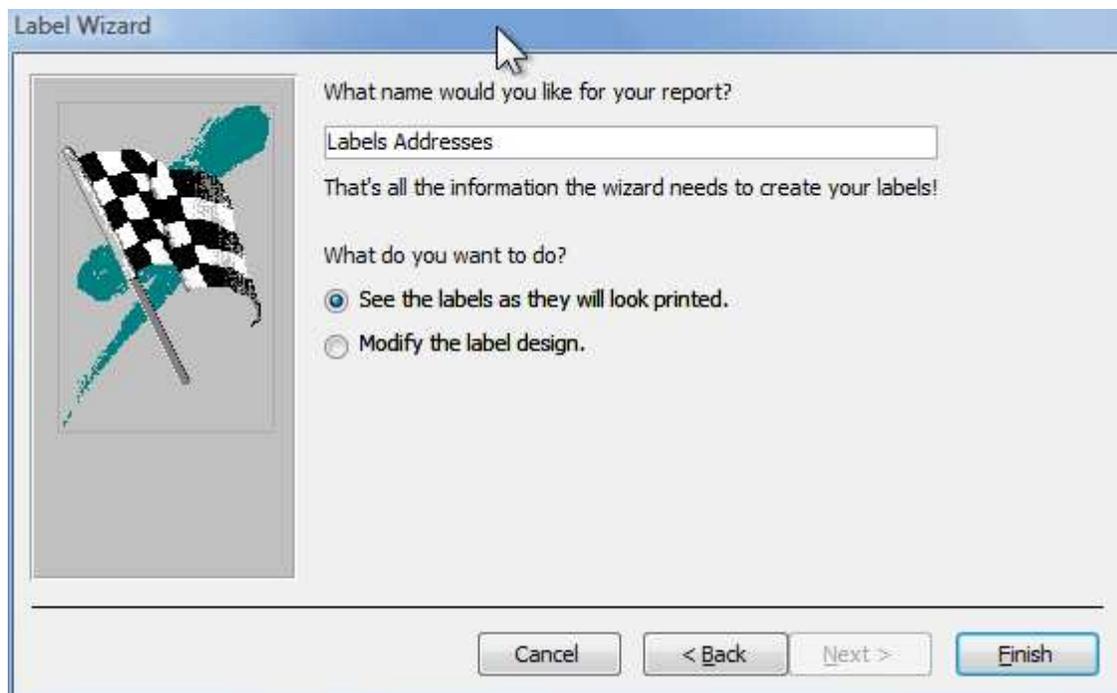
Click Finish and see a preview.

Win-Pak	Rocky Mountain High School	Andrew Marriott
421 Windchime Pl	421 Falcon Way	123 Main Street
Colorado Springs CO 80919	Colorado Springs CO 80919	Colorado Springs CO 80919

Clint Eastwood
12445 Happy Acres Drive
Colorado Springs CO 80919

Gene Autry
7466 Carter Valley Road
Colorado Springs CO 80919

Roy Rogers
7411 Bullet Lane
Colorado Springs CO 80919

John Wayne
4521 Mountain View Terrace
Apartment # 315
Colorado Springs CO 80919

Mega Mart #200
7415 Union Blvd
Colorado Springs CO 80919

Mega Mart
2154 Mountain Springs Road
Colorado Springs CO 80919

John Adams
5411 2nd Street
Colorado Springs CO 80919

George Washington
1234 Mount Vernon Lane
Colorado Springs CO 80919

Dealer X
1234 Main Street
Colorado Springs CO 80919

TELUS
1234 Main Street
Colorado Springs CO 80919

Notes: