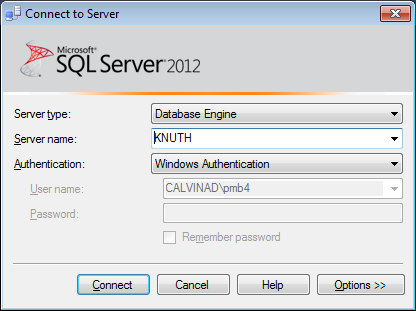
Goal: This lab will introduce you to the SQL Server RDBMS environment and the SQL Server Management Studio (SSM) tool. At the end of this lab you should be able to do the following:

* Use SQL Server Management (SSM) studio to connect to a database
* Understand the purpose to each of the system databases
* Implement the database example available from the author.
* Know how to open a table using SSM
* Navigate with SSM to find database objects
* Create and save simple queries in SSM
* Connect and create queries using the command line tool sqlcmd and Powershell

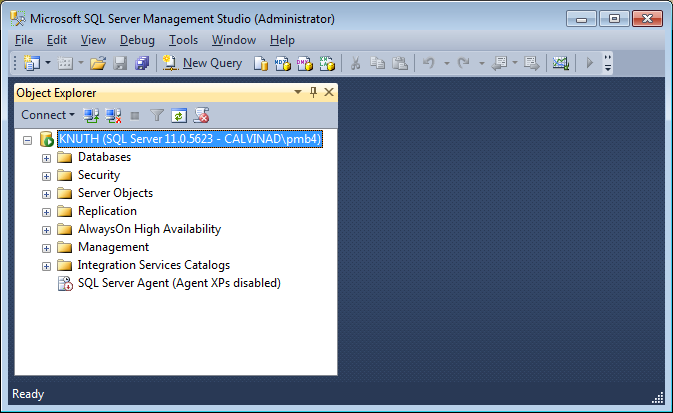
Read: Chapter 1 and 2 of Murach’s SQL Server 2012 for Developers

Approach: First, be ready to record your answers so that they can be directly placed into Moodle in the assignment labeled “SUBMIT LAB RESPONSES TO QUESTIONS HERE”. Please use a separate MS Word document for your responses. Note the step number for the response.

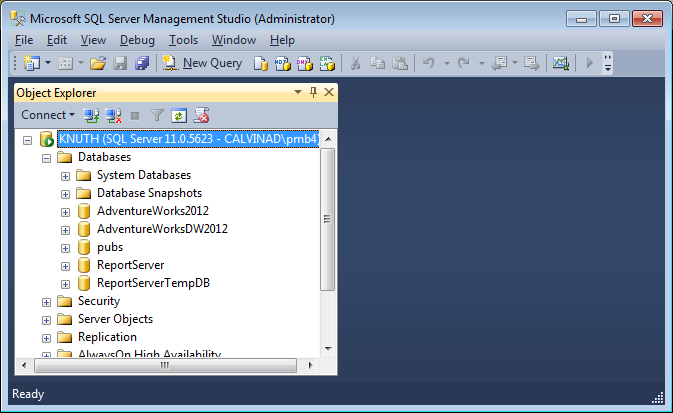
1. Login in to Microsoft Windows. After you log in, start SQL Server Management Studio available through Windows 10 under the menu selection for Microsoft SQL Server 2016…Make sure you start it by right clicking on it and choose “Run as Administrator”
2. Verify that you are connecting to the local server. The “Server Name” should match the name on your work station (e.g. maroon12) and the Authentication should show “Windows Authentication” similar to the illustration below.



1. Now select the “Connect” button. After a pause, SQL Server Management Studio will be connected to the server. Your display should look similar to the following:



1. Note the name of the server appears on the left-hand side, it is at the very top. The list on the left-hand side is the Object Explorer.
2. Keep in mind that SQL Server Management Studio (SSM) itself is only a front end that allows you to connect to a database. It does not have any special access to the database. Any information it reveals to you about the database is based on sending queries to the database. The list in Object Explorer is the list of objects the database returned based on a query SSM sent to the server. You are observing the client-server model and SSM is the client in this scenario.
3. In the object explorer, expand the Databases folder (click on the plus [**+**] sign). Your display should look similar to the following:



1. The server can have multiple databases. Two of the listings you see (AdventureWorks2012 and pubs) are application databases. Those will be explored in a moment, but for now, you will see an entry for System Databases. Expand the entry for System Databases. Record the names of the databases you see after expanding it:

* Master
* Model
* Msdb
* tempdb

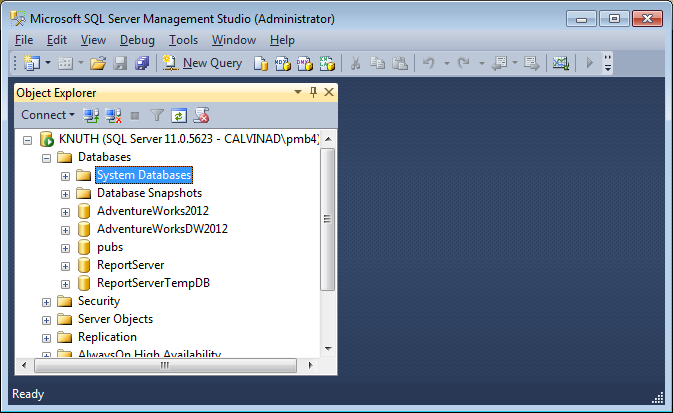
1. Go to the following website: <https://technet.microsoft.com/en-us/library/ms178028%28v=sql.110%29.aspx> It is Microsoft’s documentation describing the System Databases. On the left-hand side, you will notice a menu list that includes the other three as well. For the four System Databases listed above, read the descriptions about their purpose.
2. Of those databases, which is the one that you should avoid altering directly? Explain why?

* From what I can see, you should never change the master database, as it records all the system level information. Aside from holding all system, level information, the master data base also holds records of all other databases and the location of those data base files.

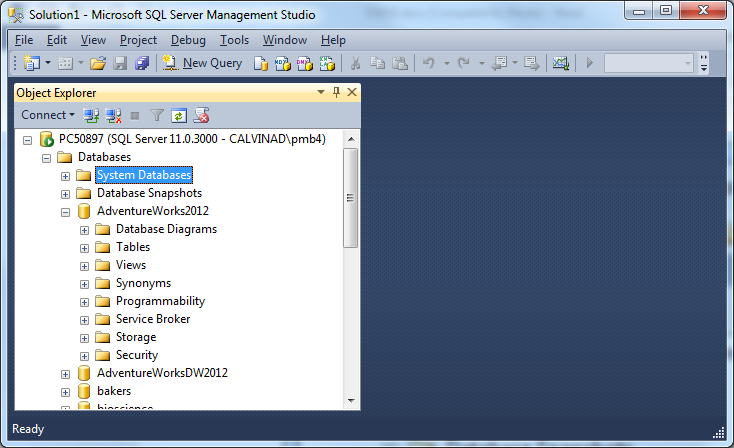
1. You will notice that the documentation makes reference to the “Resources Database” Explain the purpose to that database.

* The purpose of a resource database allows for easier upgrading version of sql server and faster procedures. The resource db contains all system objects (which includes upgrades and updates), which when upgrading you can just simply copy the single resource db.

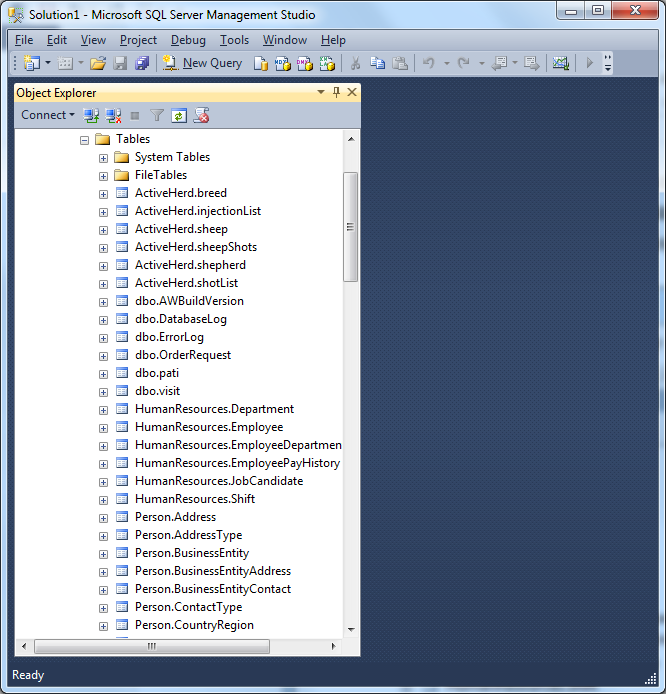
1. Collapse the System Databases folder. Your display should look similar to the following:



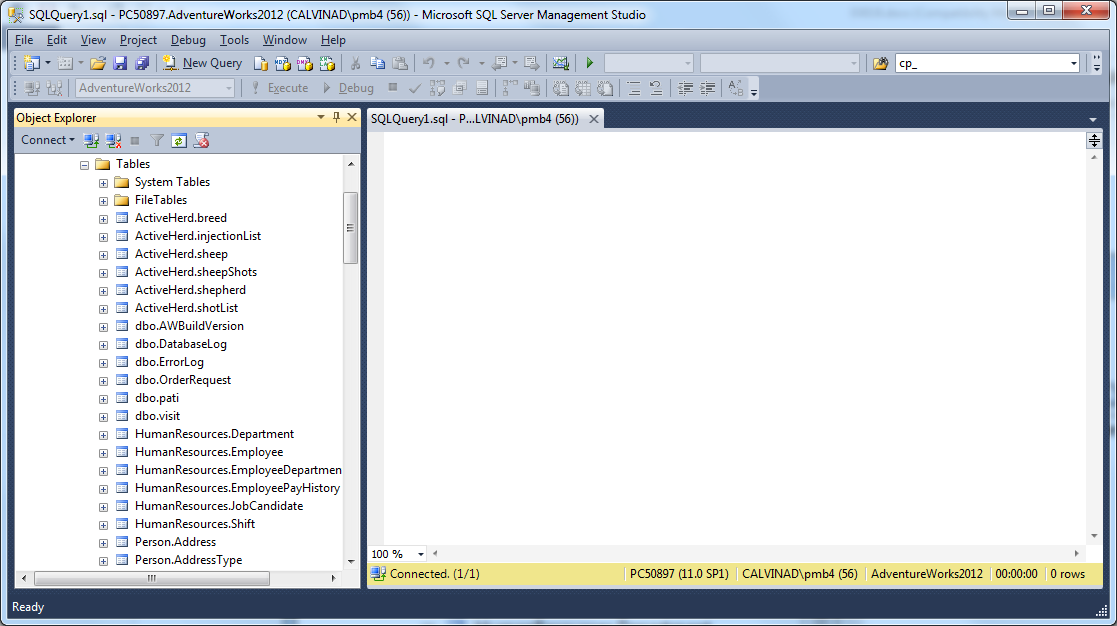
1. Expand the database entry for AdventureWorks2012 by clicking on the plus (+) sign next to it. Your display should look similar to the following:



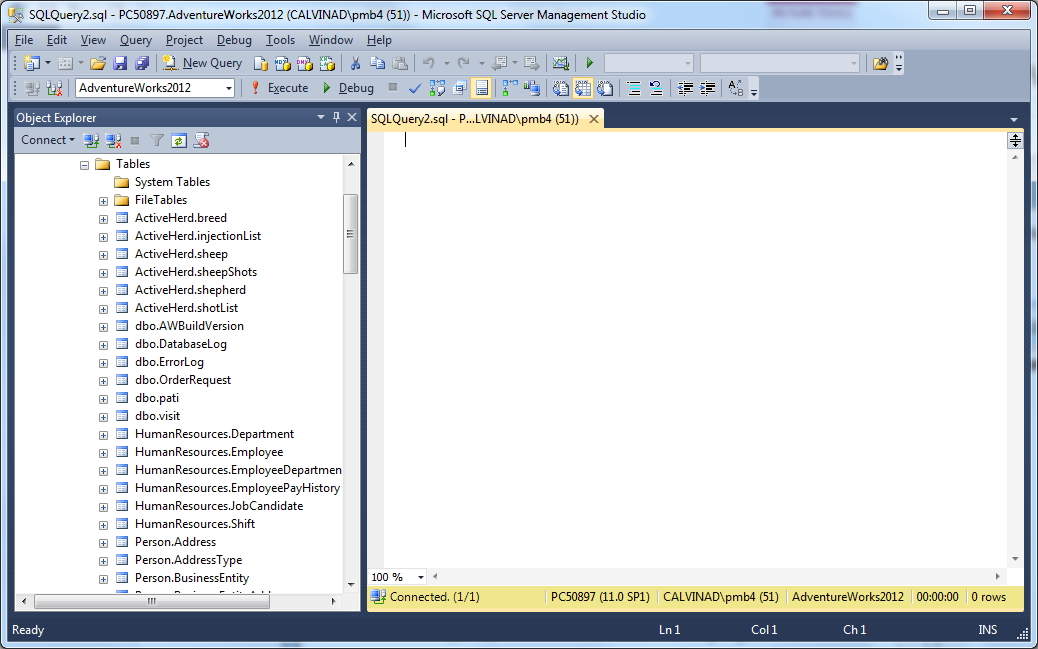
1. AdventureWorks 2012 is a tutorial database provided by Microsoft. The list beneath its entry includes the folders to group the objects in the database. We’ll look at a few of the folders that are of immediate concern to us.
2. As you know from the reading and our lecture, a relational database system organizes its data into consistent tables. For the database, the applicable tables are under (of all places) the Tables folder. Expand that folder by either double clicking on it or selecting the plus (+) sign. Your display should look similar to the following:



1. You’ll notice another folder named “System Tables.” In our setup, that folder is empty, and we are not concerned with it. Observe the tables which have a light blue icon next to them. The AdventureWorks2012 database has several tables representing a typical manufacturing organization. They are organized by schema names. A schema name provides a logical grouping to tables. You might think of it as a namespace (Note: in other DB’s and previous versions of SQL Server, the schema also directly referred to the owner of the table. More on that later.) Generally, you should get in the habit of using the full schema and table name combination. You don’t have to for those named “dbo” (short for database owner), but it’s recommended that you should.
2. We will now open up a query window so you will understand how to use SSM as a front end for ad hoc queries using SQL.
3. In the tool bar of SSM, you will see the option “New Query”. It is above the Object Explorer and right under the File option. Click on that. After a moment, you will see the right-hand side of SSM change to a text editable box similar to the following:



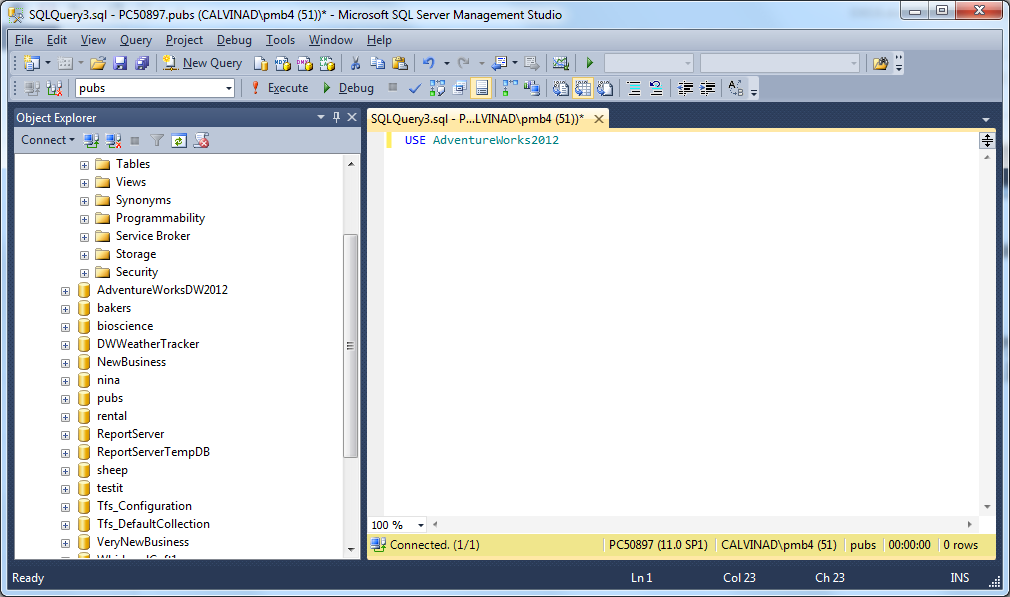
1. The query window allows you to submit SQL commands to the server for the current database context. Your current database context was set because the last folder you were in under Object Explorer was the folder for the AdventureWorks2012 database. Look in the lower right-hand portion of your display (like the one above). You should see “AdventureWorks2012” in the third box from the right. *If for some reason, it has something other than AdventureWorks2012, don’t worry. We’re about to go through how to change database contexts.*
2. **Changing Database Context:** First, make sure you click on the query window on the right hand side. If you want to change the context to one of the other databases, you have two ways of doing so. First, SSM provides a menu driven approach. Above the object explorer, you will see a drop down menu. It is here



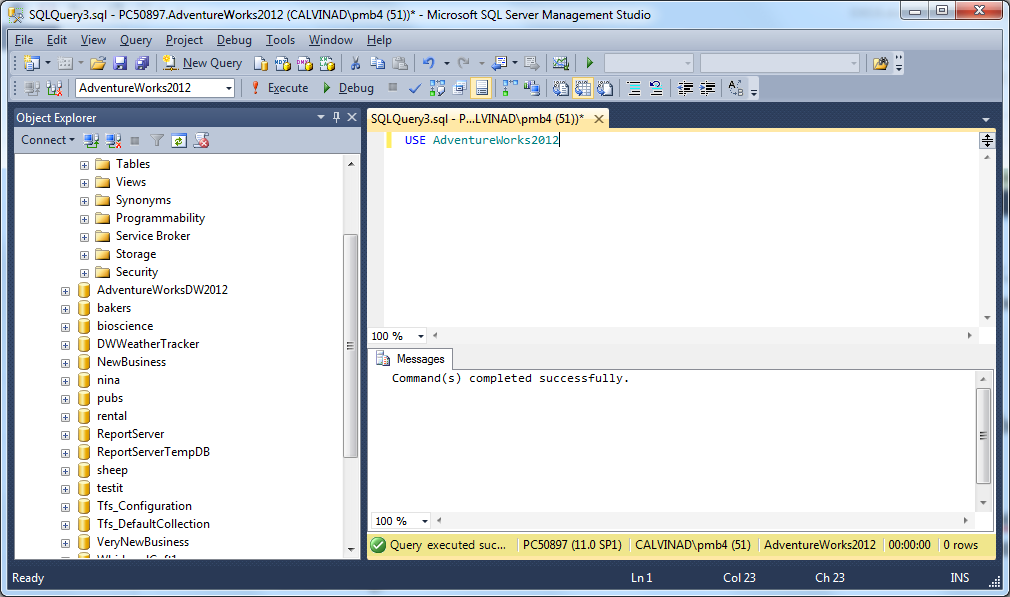
1. Click on the down arrow, and you will see the other databases available. Select the **pubs** database. You should notice the context name changes in the lower right-hand corner.
2. The second way to change database context for a query window is by issuing the “USE” command to the server. To change back to the AdventureWorks2012 database, click on the query window. Then key in the following:

USE AdventureWorks2012

Your display should look similar to the following:



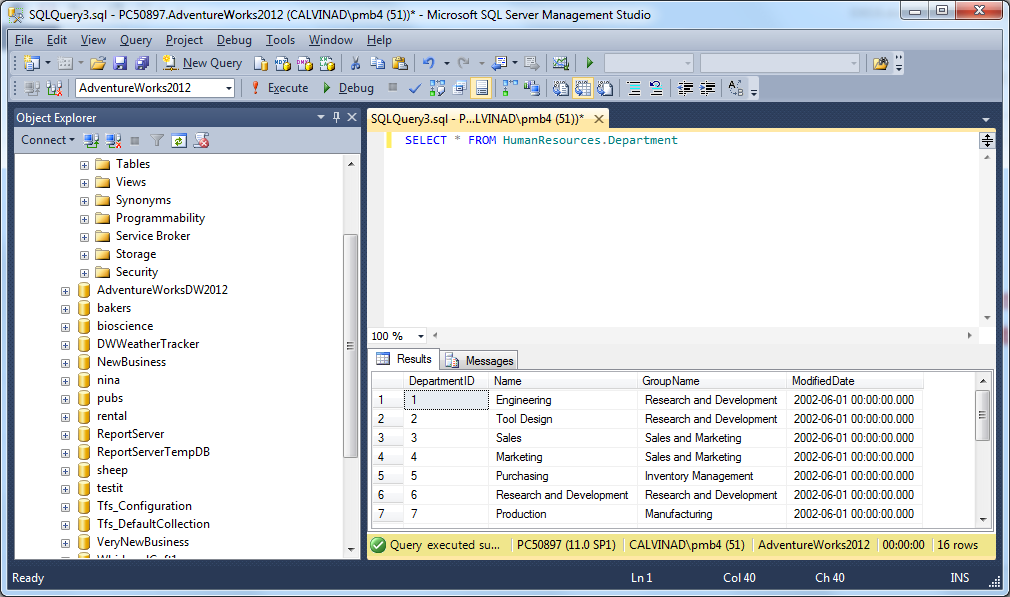
1. To send the command to the server, you can press either the **F5** key, or you can click on the “Execute” tool (it has a red exclamation mark next to it.) After doing so, your display should look similar to the following:



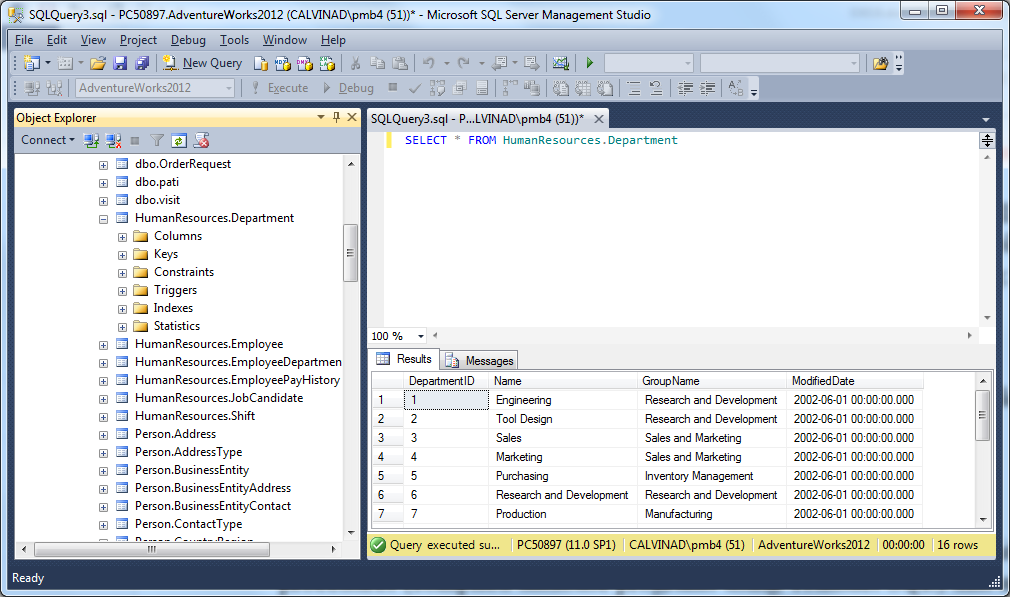
1. Note, that you have an additional window on the right hand side, labeled messages. This is letting you know your command worked if your display has the same message as the one above.
2. Now, we’ll look at how to do a simple query. Using the HumanResources.Department table, we’ll do a simple SQL SELECT statement. In the query window on the right, delete the previous command using appropriate editing tools and replace it with the following:

SELECT \* FROM HumanResources.Department

Then, press the F5 key or select the Execute tool. Your display should look similar to the following:



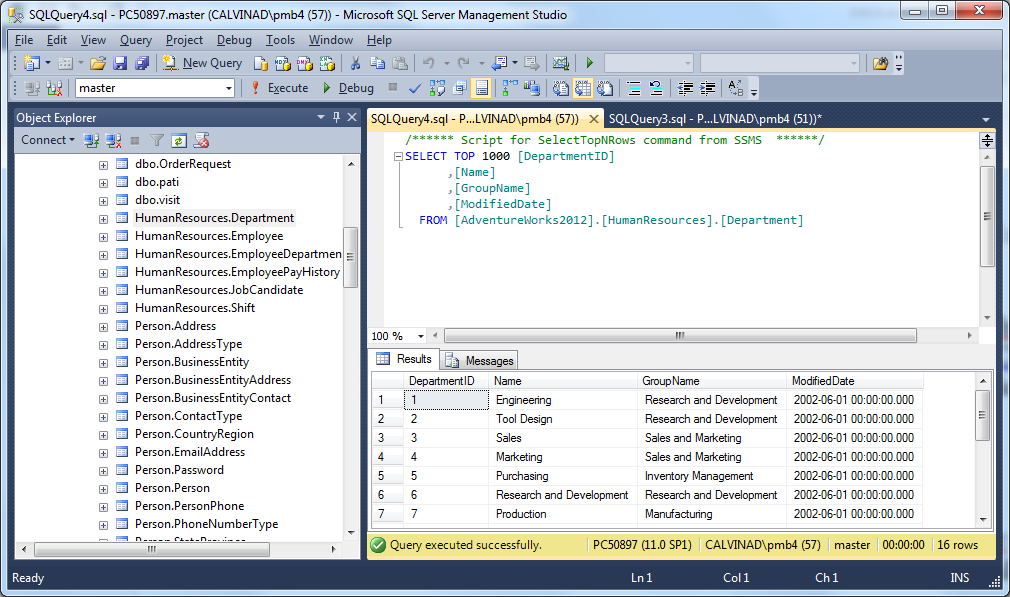
1. Notice, you have another tab below your query window. This includes the results. If you click on the messages tab, you should see a message indicating how many rows were returned. **On the response document note the message you saw in the Messages tab: ( 16 rows affected)**
2. You will often need to save your queries. To save a query, first make sure the query edit window has focus (click on its tab or somewhere in it if it is visible). Then click on the save tool (i.e. the outdated diskette icon) or select File->SaveSQLQuery1. You can select Save or Save As if this is your first time doing so. Another way of saving a query is to click on the Close (X) tool of the query window. SSM will ask you if you want to save the query. **Whatever option you choose, please save this most recent query as “myfirst.sql” to a convenient folder.**
3. Next, we will examine detailed information about the tables, we will use the HumanResources.Department table again. This time, move your mouse to object explorer (on the left) and expand the table information for HumanResources.Department (again, click on the plus sign). Your display should look similar to the following:



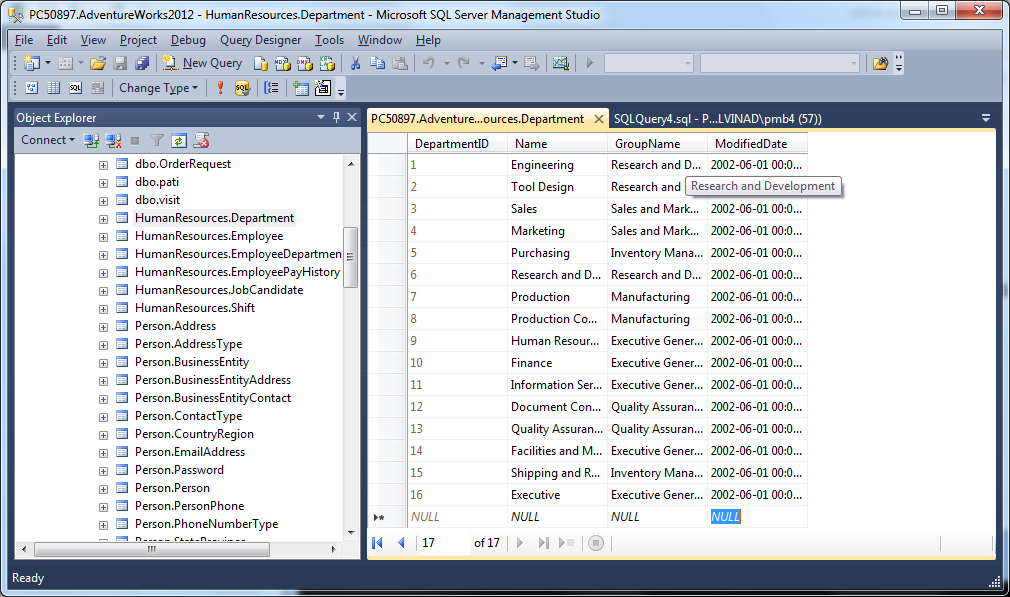
1. You will see the sub-folders for the table’s objects. For now, double click on Columns. Then, double click on Keys. Explain what you believe is being displayed:

* It’s self-explanatory. They are the column names for the HR.Dept table with the constraints.

1. The folder labeled constraints contains the formal definitions of limitations on values. The Triggers folder contains the stored procedures (compiled database programs using Transact SQL) that are “programs” called when specific events such as adding or deleting records happen. Triggers are determined and created as needed by the developer. The Indexes Folder lists those indexes defined for this specific table. The Statistics folder provides named reports on the usage of objects associated with the table, and these are used to evaluating and improving performance.
2. You can directly open the table using Object Explorer. Using the HumanResources.Department table, right click on it in Object Explorer. You will see the menu option “Select Top 1000 Rows” Choose that option. Your display should look similar to the following (If you get a Properties box displayed also, click on its Close tool (X):



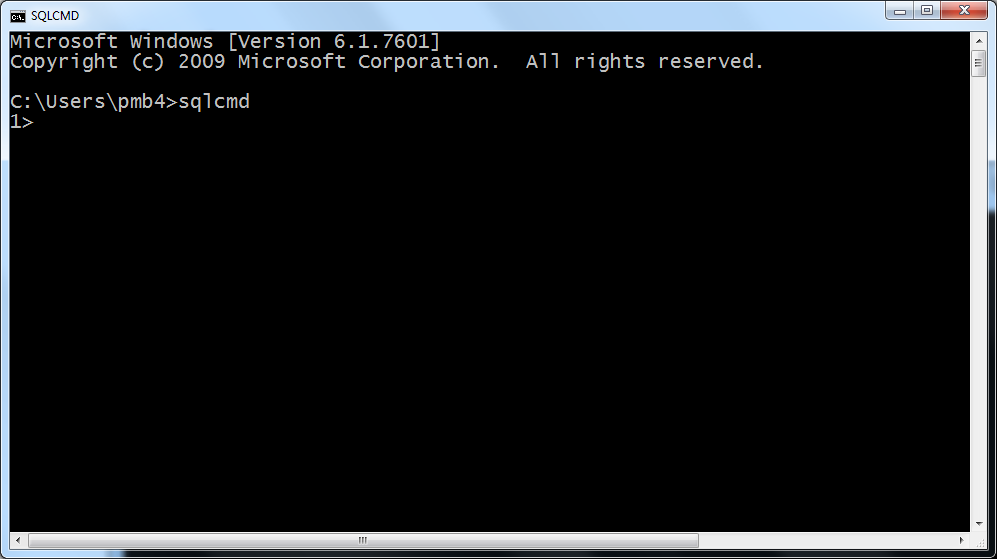
1. You can edit data interactively for theHumanResources.Department table. Right click on HumanResources.Department and select "Edit Top 200 Rows", which will give you all 16 rows. It will look similar to the display below:



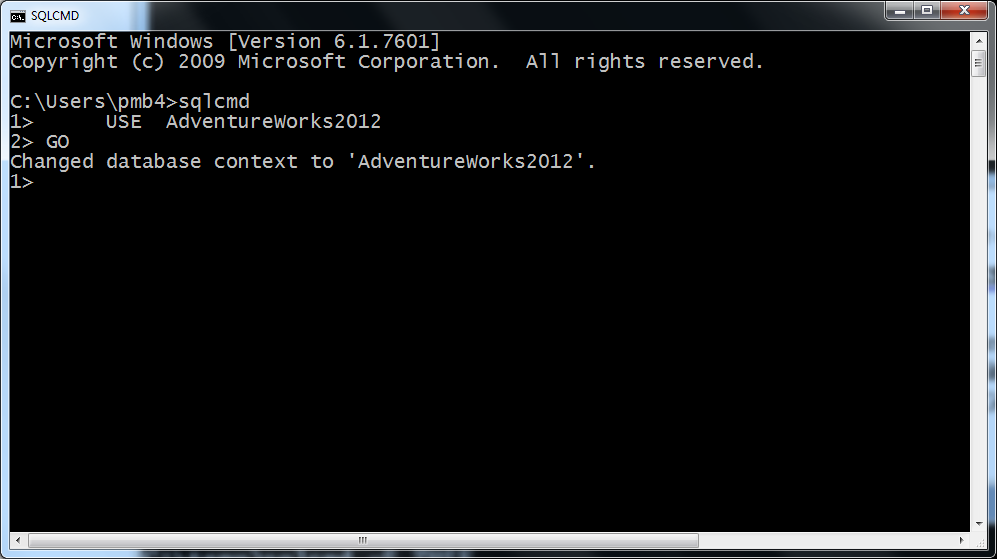
1. Go ahead and try to edit a few on your own.
2. Get ready for step 39.
3. We will stop using SSM now, and you will have a brief introduction to the command line tool **sqlcmd.**
4. To start sqlcmd, open up a command line window in Windows. You can also use Powershell, but again, make sure you start it as administrator.
5. At the command prompt, enter the following:

**sqlcmd**

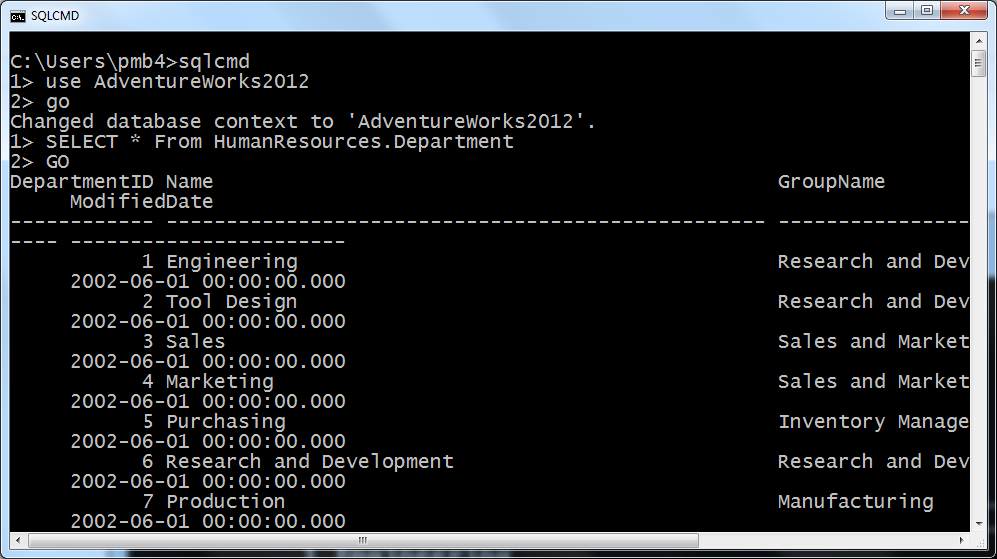
Your display should look similar to the following:



1. The “1>” is the command prompt for the sqlcmd tool. Right now, your database context is probably the “master” database. You want to use the AdventureWorks database. Enter **USE AdventureWorks2012** and press the enter key.
2. Hmmmmmm….nothing happened, right? All you got was a “2>” which indicates that sqlcmd wants more. What it wants is for you to enter “GO” on the next command line. “GO” is for the sake of sqlcmd – **it is not part of SQL**. “GO” merely tells sqlcmd that you are ready to send your command to the server. “Go” ahead and enter “GO” now! Your display should look similar to the following:



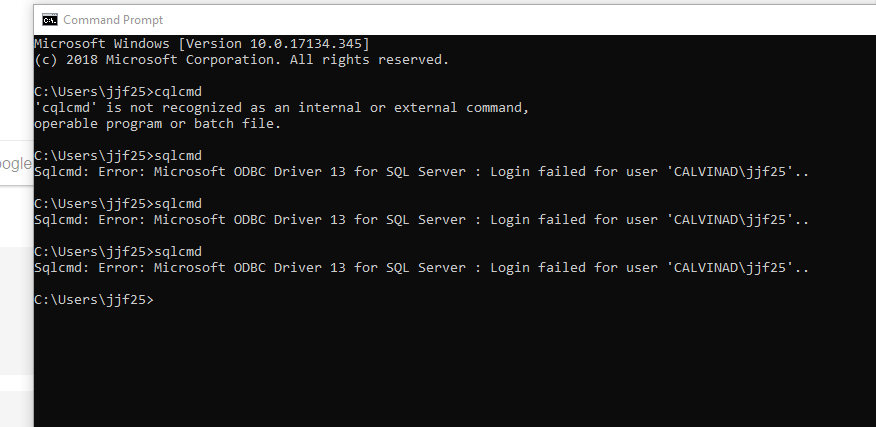
1. Why in heaven’s name would anyone use sqlcmd when SSM is soooo much more convenient? Actually, sqlcmd (and other variants) are used extensively in scripting. So, it will be well worth your time to become familiar with its nuances someday.
2. On the next command line, enter the general SQL command to list all of the contents for the HumanResources.Department table. Make sure you hit enter after GO. You should get the following results:



1. Eeech. Ugly. I know.
2. We'll look at one more feature that comes with SQL Server 2012 SQL Server Manager Studio (not in versions prior to 2008). That's the use of Powershell. Start Powershell as Administrator
3. Now we'll run the command to list all the departments. At the Powershell command prompt, enter the following:

**Invoke-Sqlcmd -ServerInstance localhost -Database AdventureWorks2012 -query "SELECT \* From HumanResources.Department"**

1. Now, think about a couple of subtle differences in using Powershell compared to the cmd.exe when using the sqlcmd command. Comment on those differences.

* Unable to use. Because of this issue. 

1. We’ll stop for now. I’m sure this exercise has been tedious enough. Please make sure your responses are entered into the MS Word document and submitted into Moodle assignment named “SUBMIT RESPONSES TO LAB INSTRUCTIONS HERE”
2. Thanks!