DB Backup Instructions

Note: These instructions are for Microsoft SQL Server Management Studio under Microsoft SQL Server 2008 R2

Contents

[1 Before Backing Up: 2](#_Toc346457974)

[2 Backing Up Your Database 4](#_Toc346457975)

[2.1 Creating a .bak 4](#_Toc346457976)

[2.2 Creating a .sql 6](#_Toc346457977)

[3 Restoring your Database 10](#_Toc346457978)

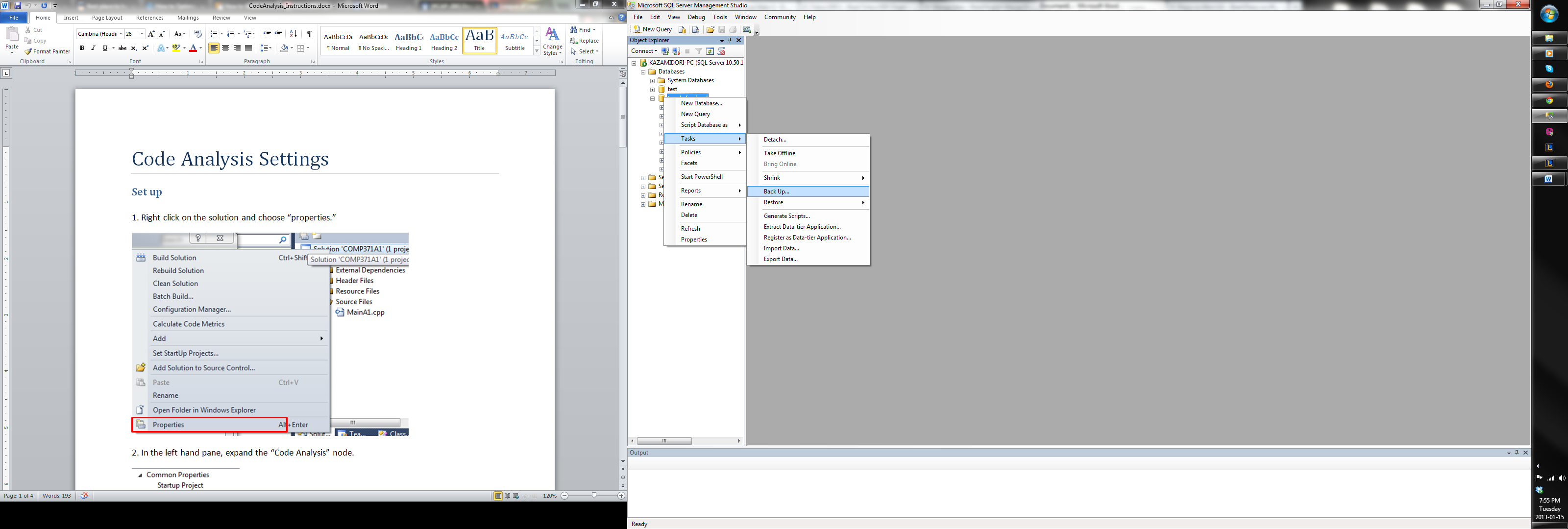
[3.1 From a .bak 10](#_Toc346457979)

[3.2 From a .sql 14](#_Toc346457980)

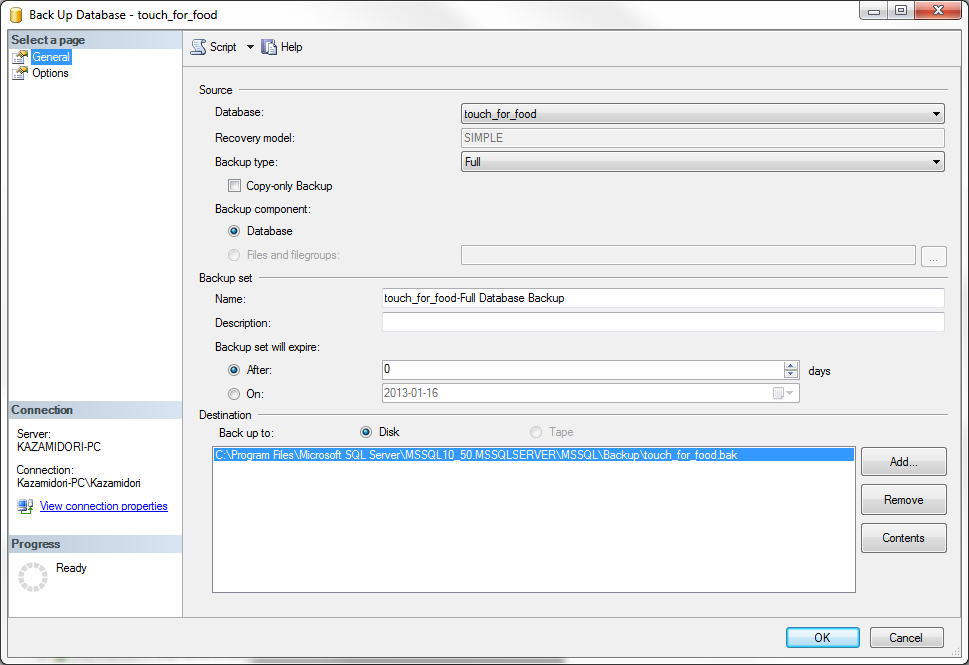
# Before Backing Up:

Be sure to check that there is no pre-existing .bak in your system. If there is an existing .bak, running the backup can result in multiple instances of the database to be stored in a single .bak. This can and has caused us problems and is also unnecessary because we have been using SVN to keep a history of our databases. There are a few ways to go about removing the .bak; the simplest and surefire way is to empty your Backup folder. In a later section, I will demonstrate how to back up without having to physically go to empty the Backup directory.

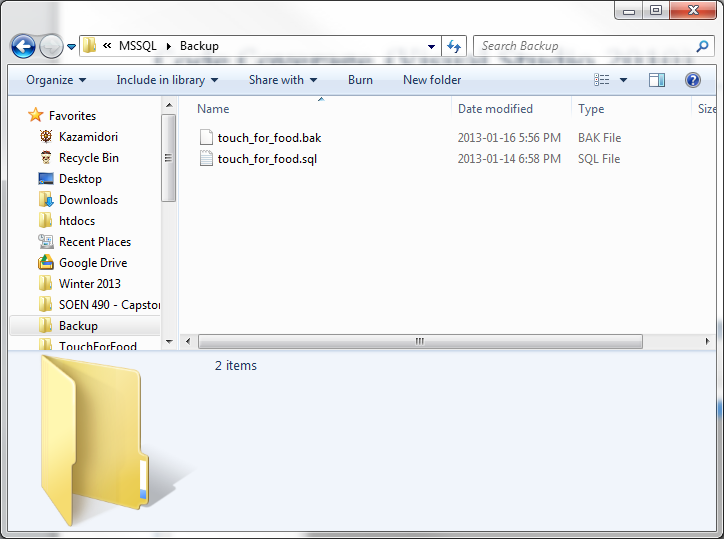
Mine is located at C:\Program Files\Microsoft SQL Server\MSSQL10\_50.MSSQLSERVER\MSSQL\Backup, but because our setups are different, here is where you can find the location of your Backup directory.

1. In the “Object Explorer”, right click on the database you want to backup and go to Tasks 🡪 Back Up…

2. Under Destination, you will see the path to your Backup folder.



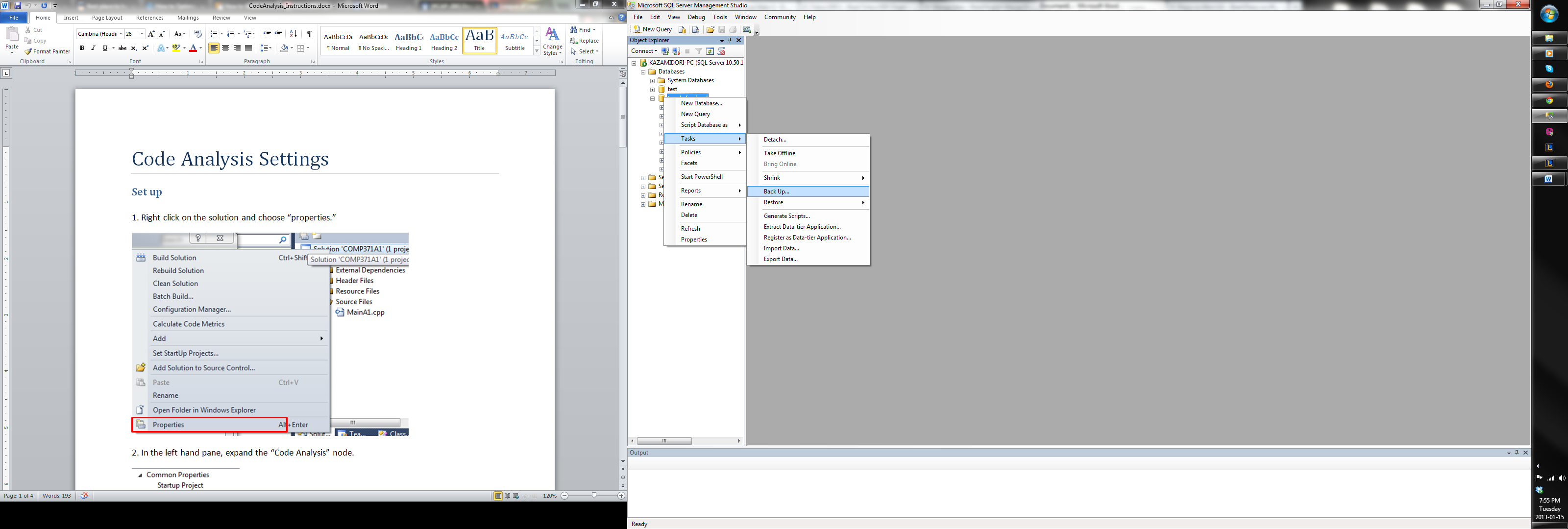
3. This is what’s currently in my Backup folder. If you are absolutely paranoid about backing up your database, empty this directory

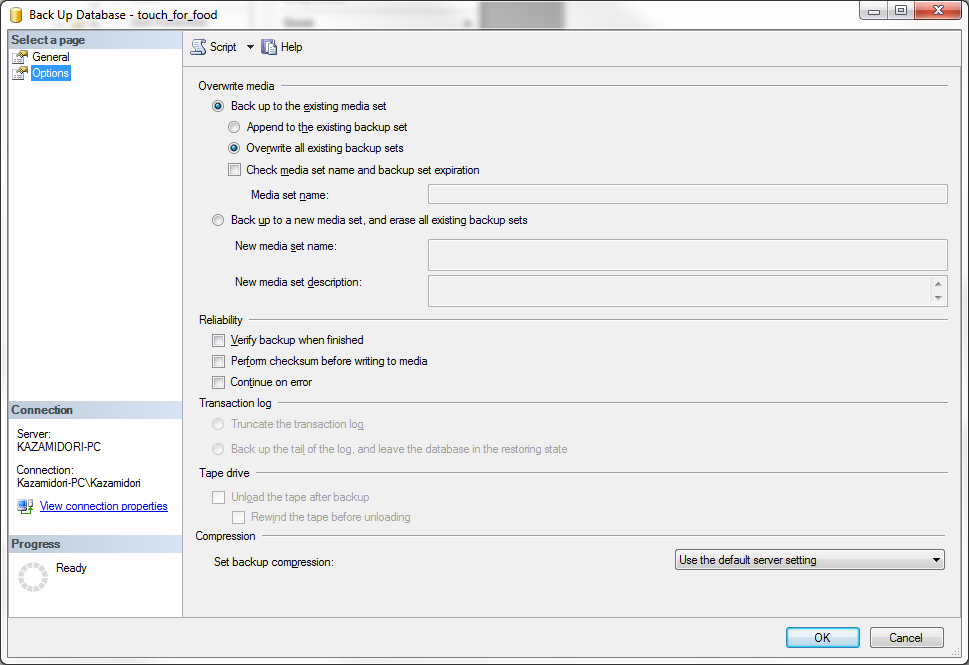


# Backing Up Your Database

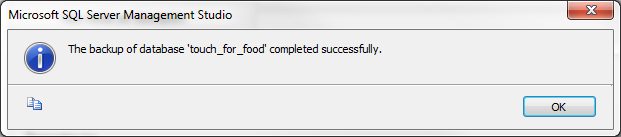
## Creating a .bak

1. In the “Object Explorer”, right click on the database you want to backup and go to Tasks 🡪 Back Up…



2. Under “Select a page”, make sure to check off “Overwrite all existing backup sets”

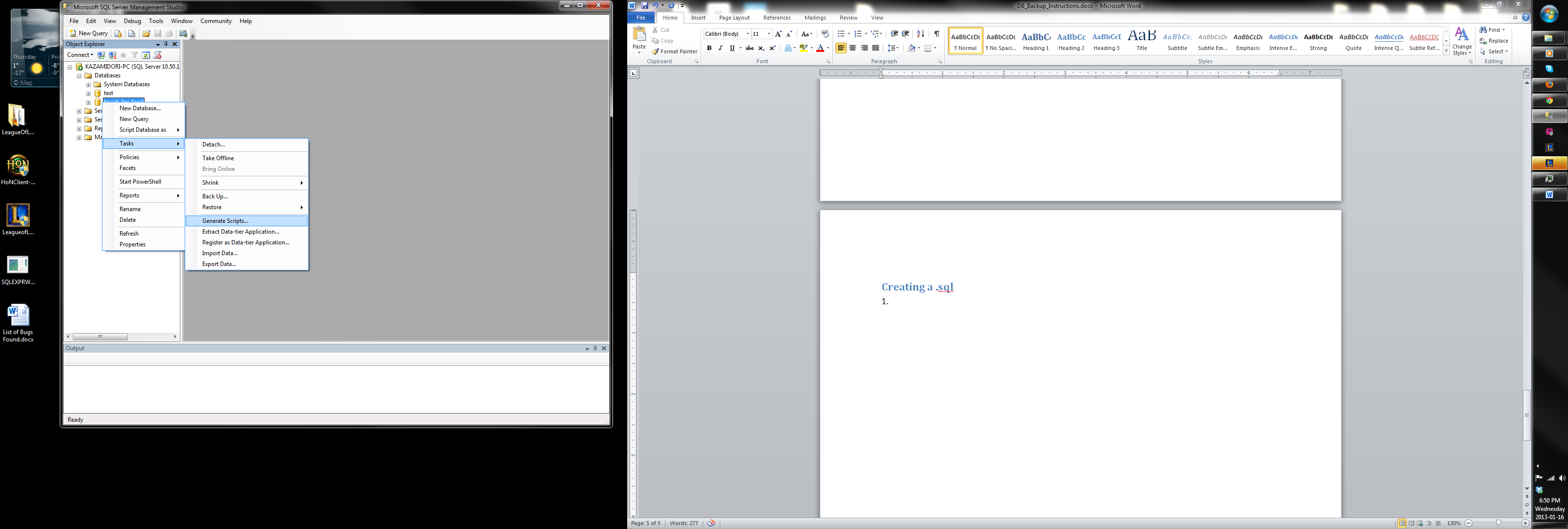
3. Press OK and the following dialog will pop up:



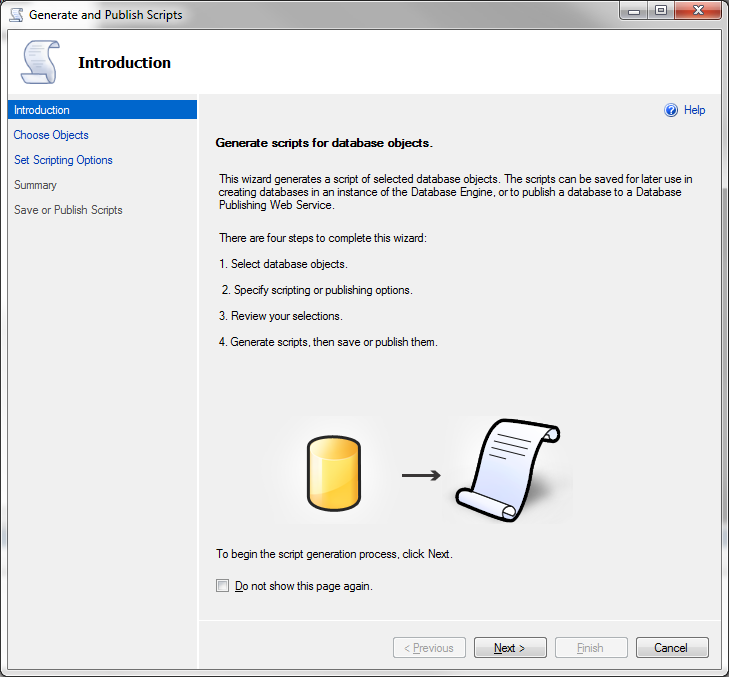
You can check the directory mentioned in the previous section to see that the .bak exists. Be sure to commit the .bak to our SVN repository if the changes made are crucial to TFF’s functionality.

## Creating a .sql

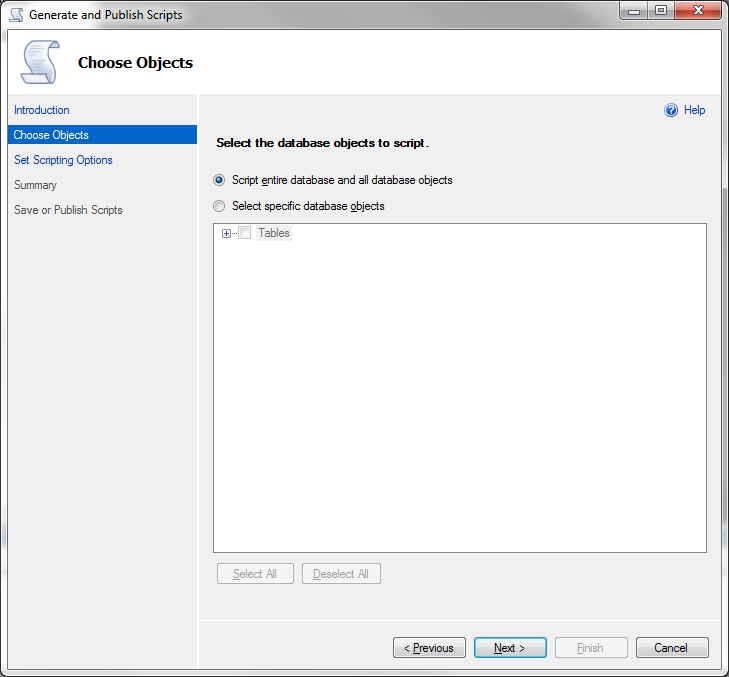
1. In the “Object Explorer”, right click on the database you want to backup and go to Tasks 🡪 Generate Scripts…



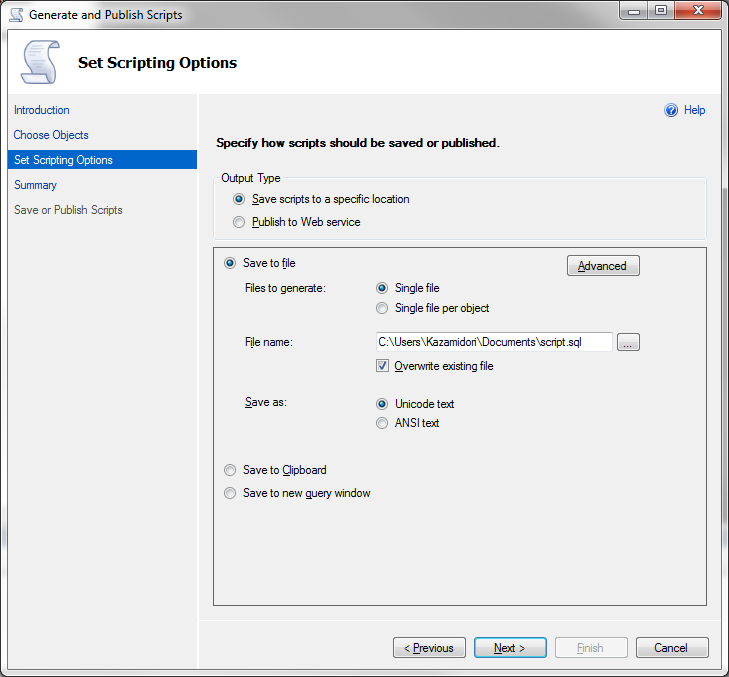
2. Click Next



3. Click Next



4. Click “Advanced”. Note, the file name field specifies where the .sql will be created. You can modify the path to access your .sql more easily.

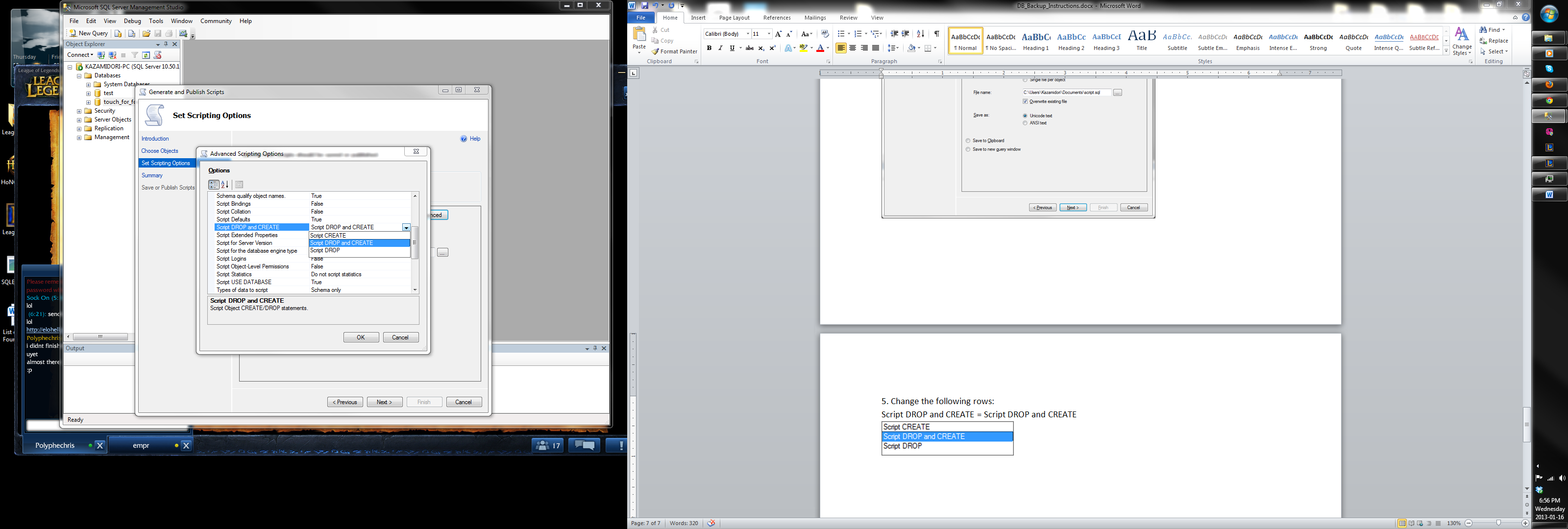


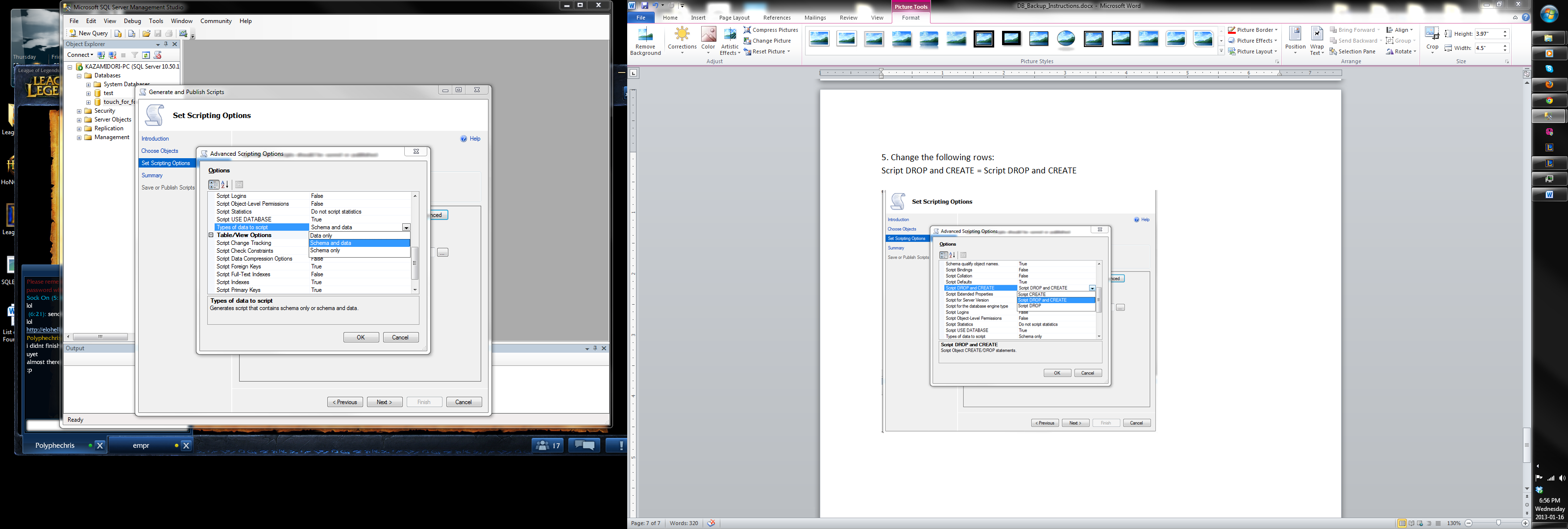
5. Modify the following rows:

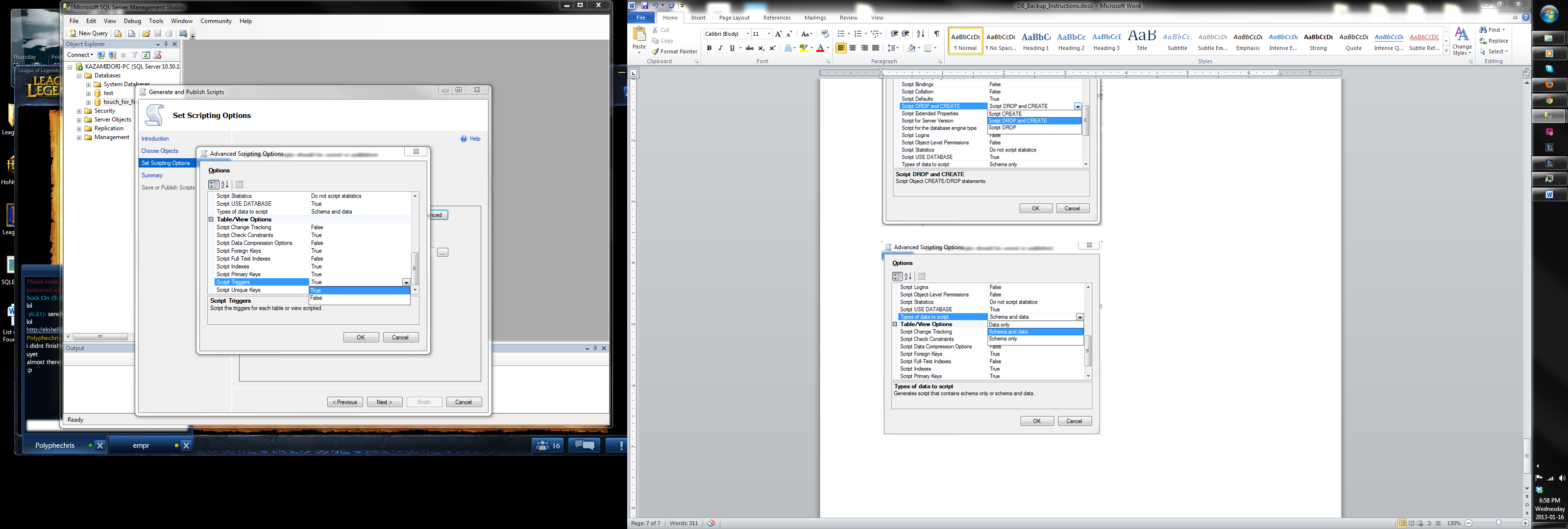
Script DROP and CREATE = Script DROP and CREATE

Types of data to script = Schema and data

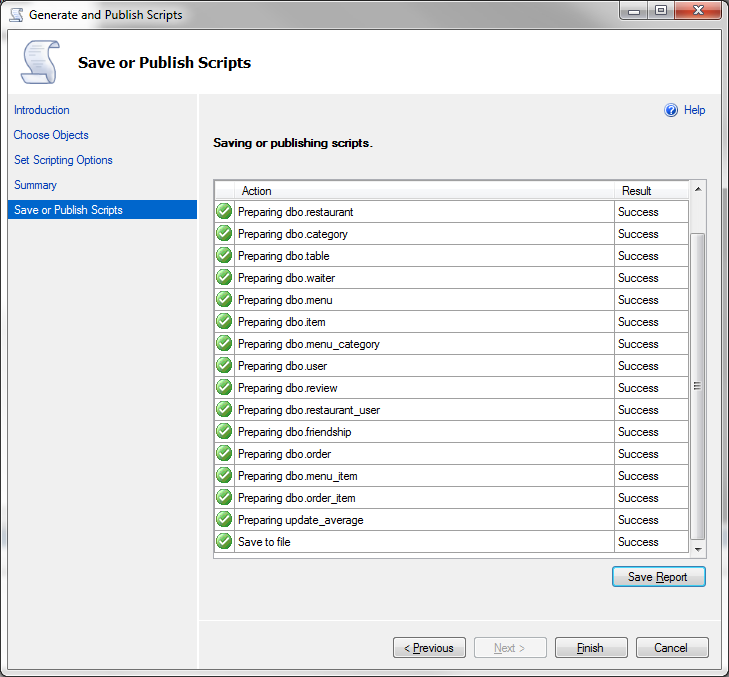
Script Triggers = True







6. Press OK and Next and Next and then Finish

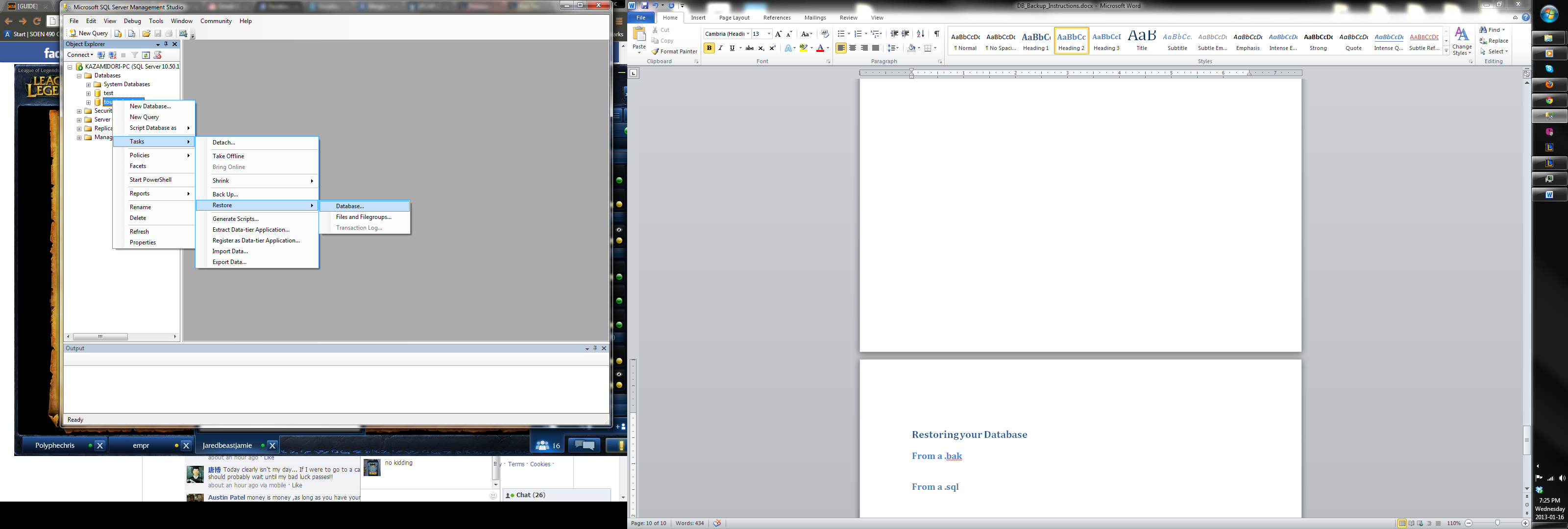


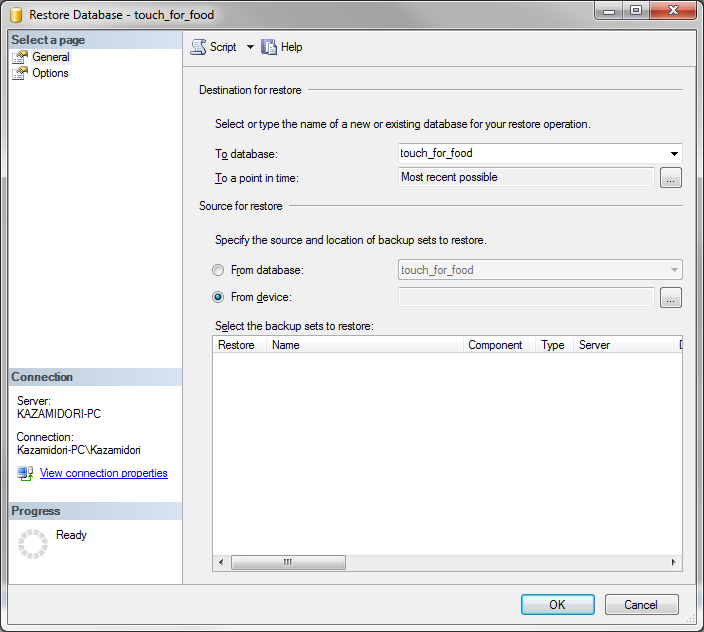
The .sql should now have been created. To find out where it is stored, refer to step 4 and look at the “File Name:” row.

# Restoring your Database

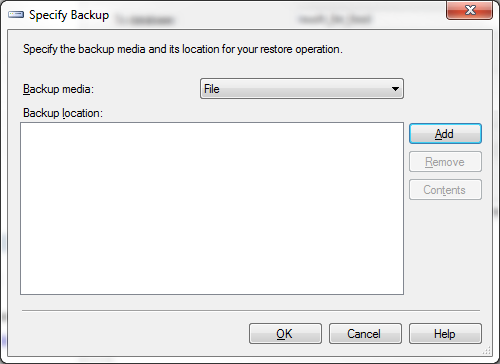
## From a .bak

1. In the “Object Explorer”, right click on the database you want to restore to and go to Tasks 🡪 Restore 🡪 Database…

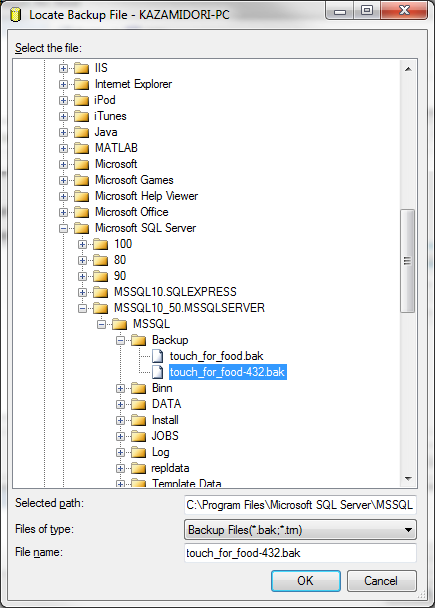


2. Select “From Device” and click the “…”

3. Click on “Add”

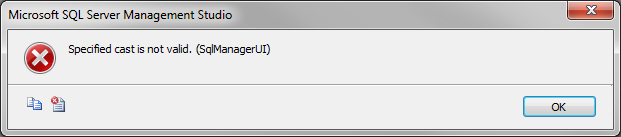


4. Select the .bak you want to restore from and press OK. Note: For ease of use, place the .bak you want to restore from in the directory Backup mentioned earlier since this is the default location of where the studio opens at.



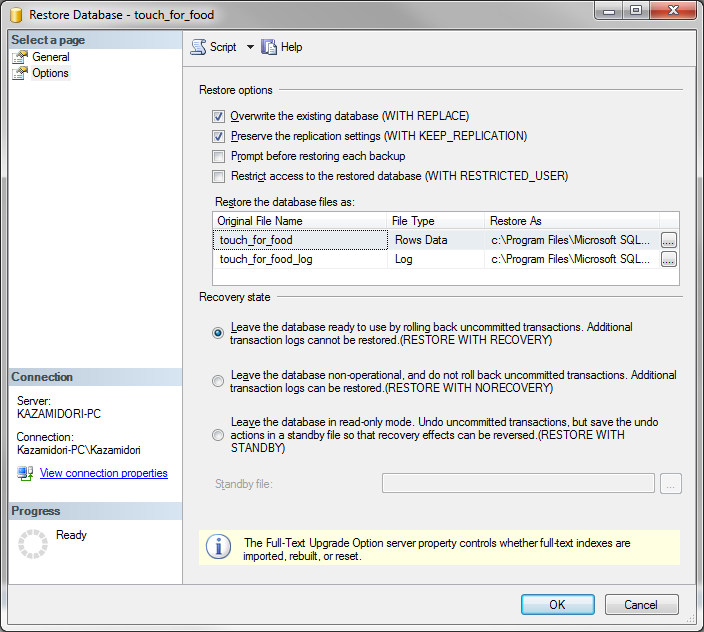
5. Press OK again. You should see this.



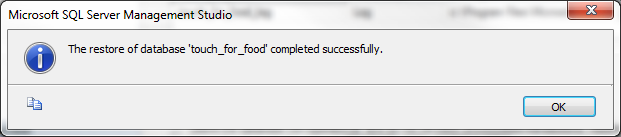
If you see don’t see that, you probably got this error. 

That means that the .bak that was committed most likely came from a different version of SQL Server (probably 2012). If that’s the case, the only way for you to restore is by running the .sql which will be covered later on.

6. Under “Select a page”, select options and check off the first two boxes under “Restore Options”.



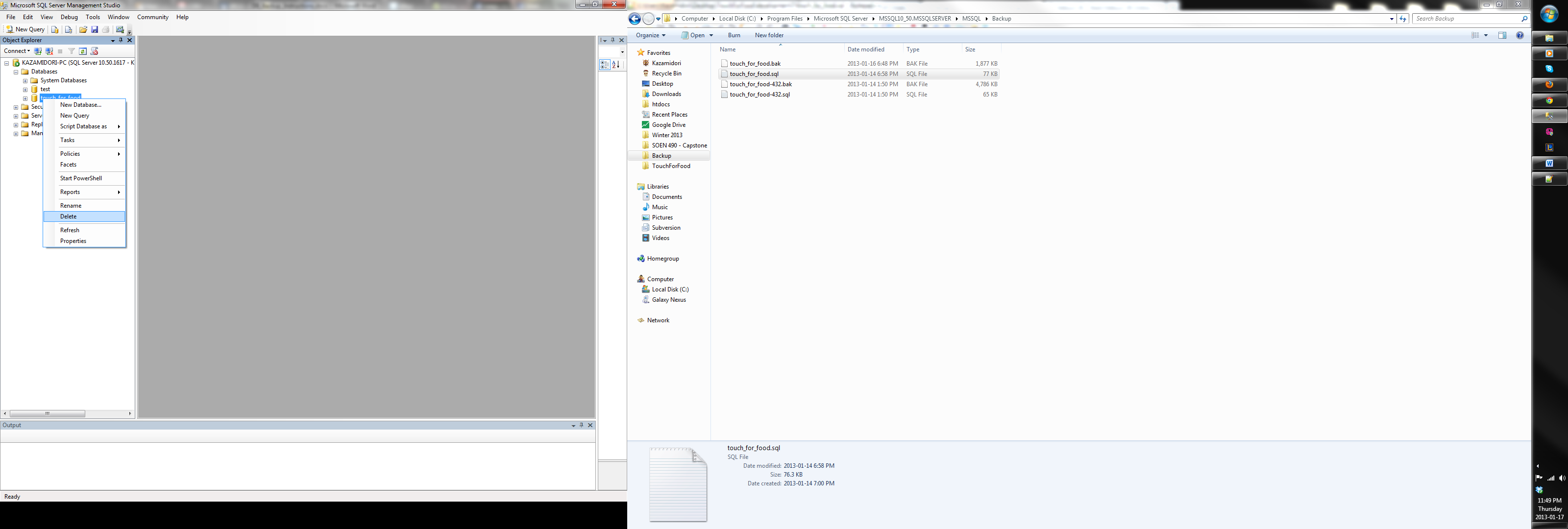
7. Press OK and should get this success message:



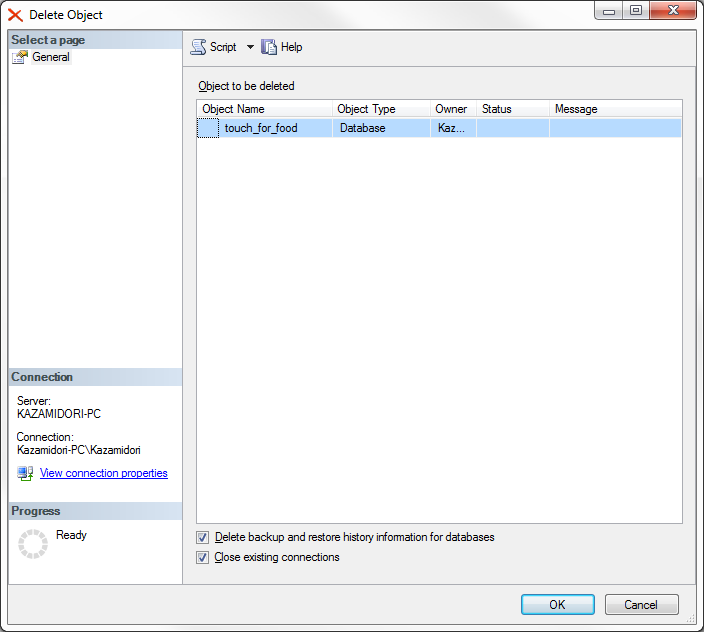
## From a .sql

It has come to my attention that a script generated by SQL 2012 has different parameters compared to SQL 2008. I will mention the steps required to run both scripts properly.

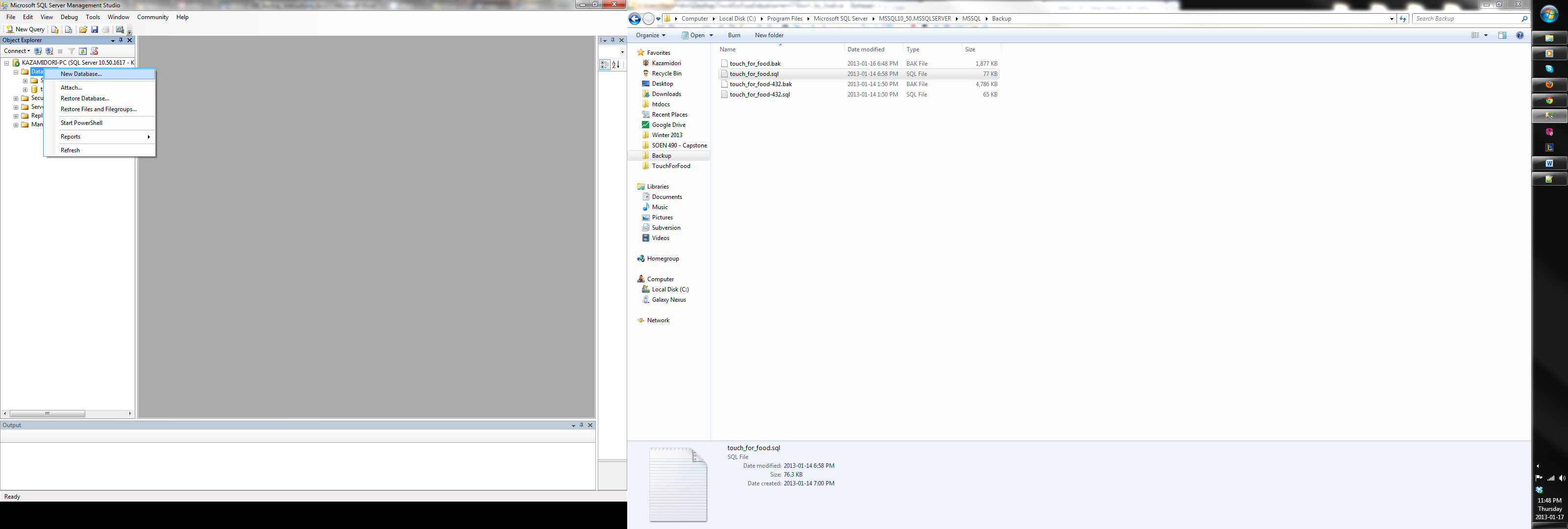
1. Delete the existing touch\_for\_food database.



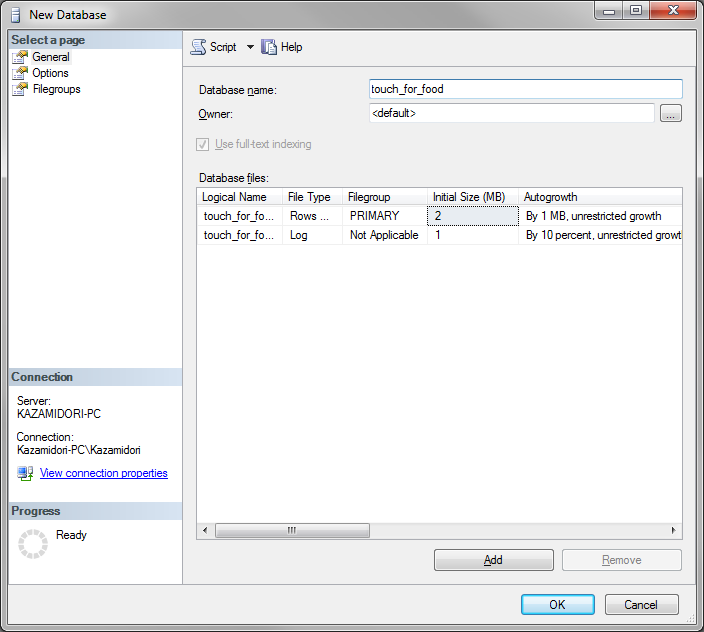
2. Close existing connections will eliminate any potential complications



3. Create a New Database

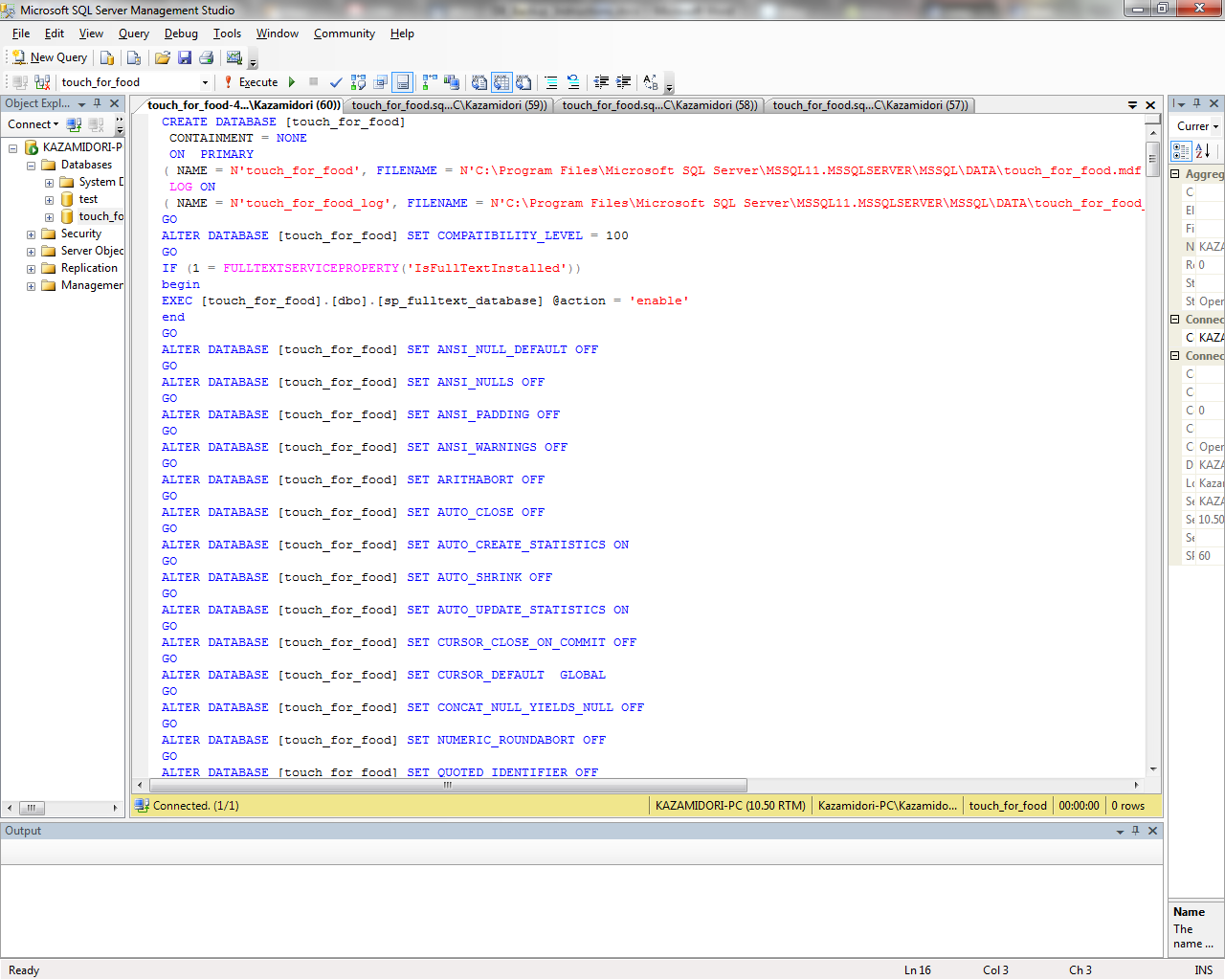


4. Name it touch\_for\_food

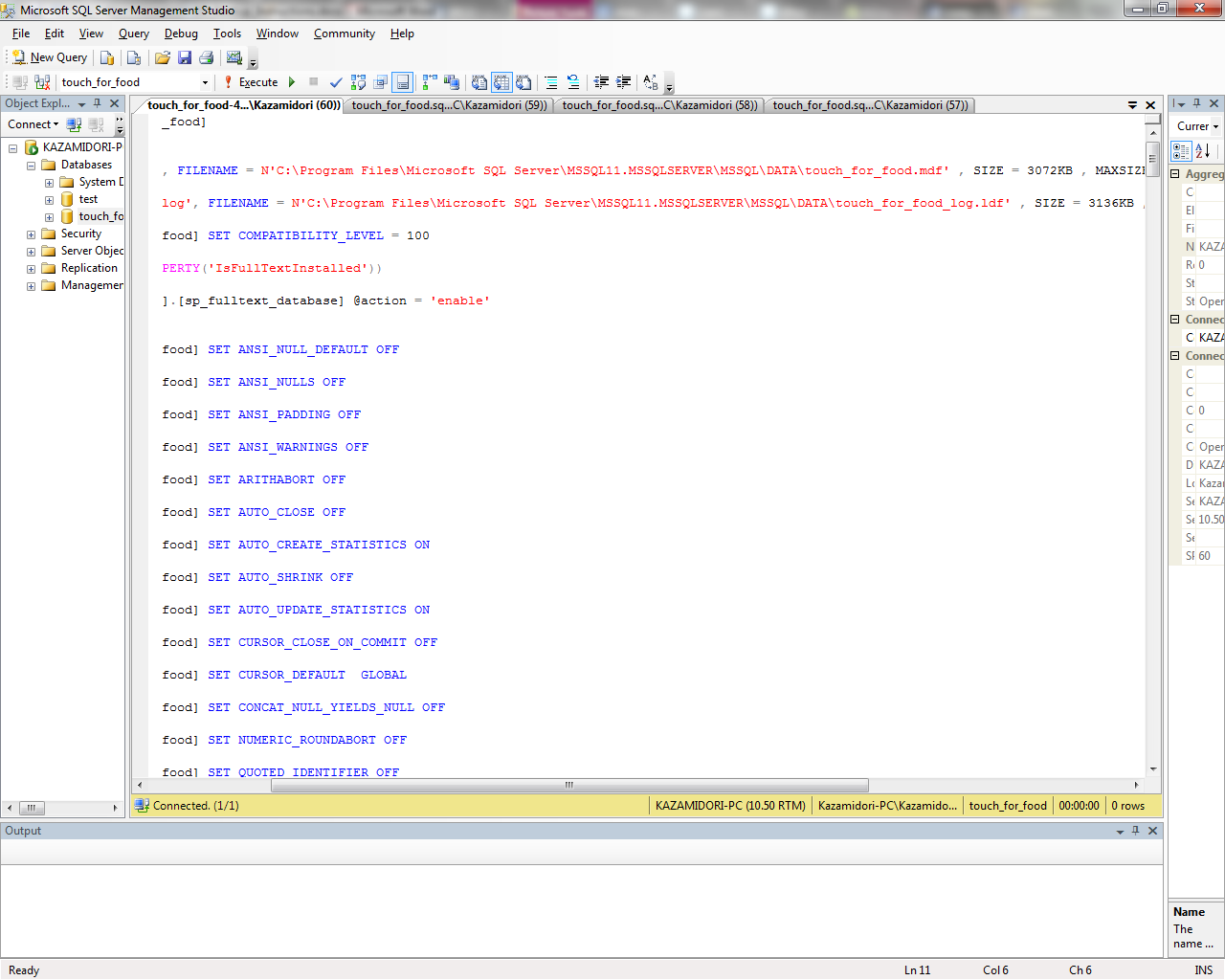


5. Last Step:

Note: I went over a few versions of the .sql file and it’s a bit confusing. Between all the commits and regeneration of the .sql file from SQL 2008 and SQL 2012, the file has never consistently remained the same… But, this is generally the solution to run the .sql if it doesn’t work for you.



If someone with SQL 2012 commits a new .bak and .sql, the only way for users running SQL 2008 to restore their database is with the .sql. However, modifications need to be made because the script points to a directory that does not exist in SQL 2008.



In SQL 2008, my directory is

C:\Program Files\Microsoft SQL Server\**MSSQL10\_50**.MSSQLSERVER\MSSQL\DATA

In SQL 2012, their directory is

C:\Program Files\Microsoft SQL Server\**MSSQL11**.MSSQLSERVER\MSSQL\DATA

Depending on your configuration, it might be located elsewhere. The root of the directory is the same place where you would place your .bak in your Backup folder (the end of the directory path is DATA instead of Backup).

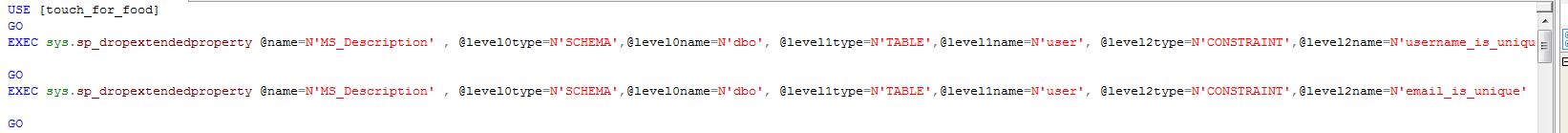
Drag the .sql file into a query window and press execute. Refresh the object explorer and there should be new tables in your touch\_for\_food database.

# .sql File Still Not Working…

This section covers any new problems discovered in the .sql file that the instructions do not cover.

## Issues from SQL 2012 .sql

1. In revision 439, the beginning of the .sql file had a bunch of ALTER TABLE arguments. Generally, the .sql file should start off with the following statement:



If the .sql isn’t working for you, be sure to compare with the previous revisions and see if the beginning of the .sql has changed drastically.

## Issues from SQL 2008 .sql