

Contents

Tutorials: SQL Server Management Studio

Connect & Query SQL Server using SSMS

Scripting Objects in SSMS





Using Templates in SSMS

SSMS Configuration

Tips and Tricks for using SSMS

Tutorials for SQL Server Management Studio (SSMS)

1/14/2019 • 2 minutes to read • [Edit Online](#)

APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

[Please share your feedback about the SQL Docs Table of Contents!](#)

The SQL Server Management Studio (SSMS) tutorial introduces you to the integrated environment for managing your SQL Server infrastructure. SQL Server Management Studio presents a graphical interface for configuring, monitoring, and administering instances of SQL Server. It also allows you to deploy, monitor, and upgrade the data-tier components used by your applications, such as databases. SQL Server Management Studio also provides Transact-SQL, MDX, DMX, and XML language editors for editing and debugging scripts.

What You Will Learn

These tutorials will help you understand the presentation of information in SSMS and how to take advantage of its features.

The best way to get acquainted with SSMS is through hands-on practice. These tutorials will familiarize you with the various features available within SSMS. These tutorials will teach you how to manage the components of SSMS and how to find the features that you use regularly.

Here is what the tutorials cover:

- [Tutorial: Connect & Query SQL Server using SSMS](#)

In this Tutorial, you learn how to connect to your SQL Server instance. You will also learn some basic Transact-SQL (T-SQL) commands to create and then query a new database.

- [Tutorial: Scripting Objects in SSMS](#)

In this Tutorial, you learn how to script out various objects in SSMS, including databases and queries.

- [Tutorial: Using Templates in SSMS](#)

In this Tutorial, you learn how to work with the pre-built Templates within SSMS. The templates are a little-known feature that store a number of Transact-SQL code snippets for various database administration tasks.

- [Tutorial: SSMS Configuration](#)

In this Tutorial, you learn the basics of configuring your SSMS environment, such as the changing the environmental layout. This Tutorial also explains what the different SSMS components are.

- [Tutorial: Additional Tips and Tricks for using SSMS](#)

In this Tutorial, you will learn additional tips and tricks for using SSMS. The Tutorial includes the following:

- Commenting and uncommenting text
- Indenting text
- Filtering Objects in Object Explorer
- Accessing your SQL Server error log
- Finding the name of your instance

Requirements

This tutorial is intended for experienced database administrators and database developers who are not familiar with Visual Studio, but who are familiar with database concepts and Transact-SQL.

You must have the following installed to use this tutorial:

- Install the latest version of [SQL Server Management Studio \(SSMS\)](#).

The first section walks you through creating a database but other sample databases can be found here: [AdventureWorks Sample Databases](#). Instructions for restoring databases in SSMS can be found here: [Restoring a Database](#).

See Also

[Database Engine Tutorials](#)

Tutorial: Connect to and query a SQL Server instance by using SQL Server Management Studio

1/14/2019 • 5 minutes to read • [Edit Online](#)

Please share your feedback about the SQL Docs Table of Contents!

This tutorial teaches you how to use SQL Server Management Studio (SSMS) to connect to your SQL Server instance and run some basic Transact-SQL (T-SQL) commands. The article demonstrates how to do the following:

- Connect to a SQL Server instance
- Create a database ("TutorialDB")
- Create a table ("Customers") in your new database
- Insert rows into your new table
- Query the new table and view the results
- Use the query window table to verify your connection properties
- Change the server that your query window is connected to

Prerequisites

To complete this tutorial, you need SQL Server Management Studio and access to a SQL Server instance.

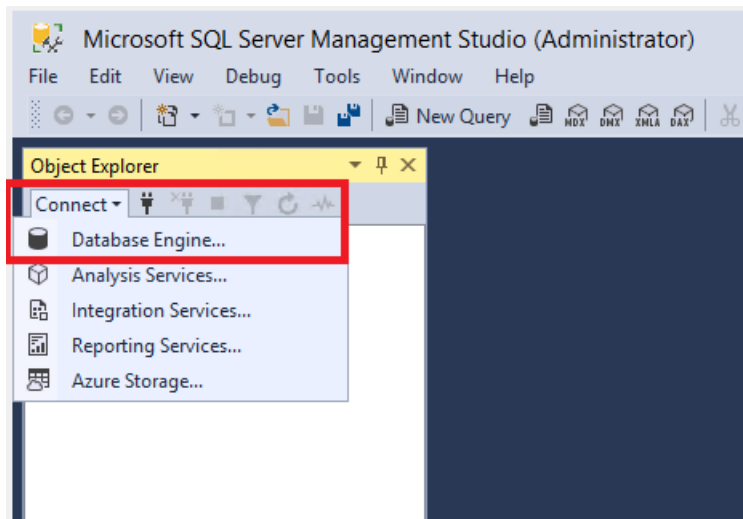
- Install [SQL Server Management Studio](#).

If you don't have access to a SQL Server instance, select your platform from the following links. If you choose SQL Authentication, use your SQL Server login credentials.

- **Windows:** [Download SQL Server 2017 Developer Edition](#).
- **macOS:** [Download SQL Server 2017 on Docker](#).

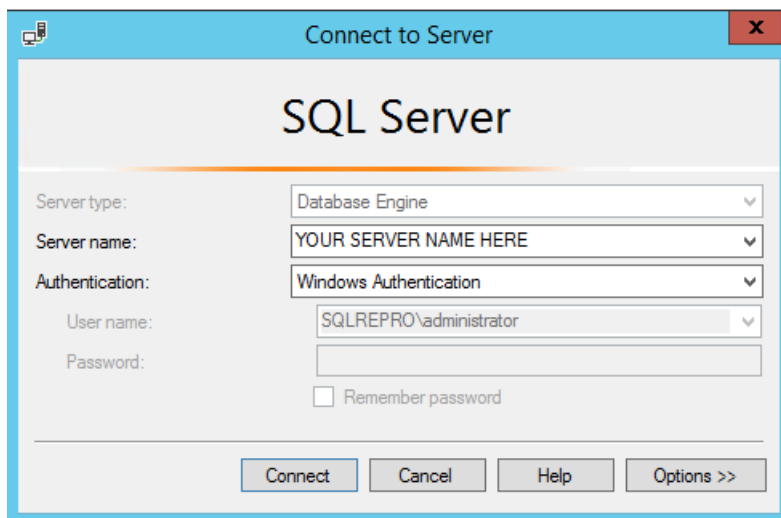
Connect to a SQL Server instance

1. Start SQL Server Management Studio. The first time you run SSMS, the **Connect to Server** window opens. If it doesn't open, you can open it manually by selecting **Object Explorer** > **Connect** > **Database Engine**.



2. In the **Connect to Server** window, do the following:

- For **Server type**, select **Database Engine** (usually the default option).
- For **Server name**, enter the name of your SQL Server instance. (This article uses the instance name SQL2016ST on the hostname NODE5 [NODE5\SQL2016ST].) If you're unsure how to determine your SQL Server instance name, see [Additional tips and tricks for using SSMS](#).



- For **Authentication**, select **Windows Authentication**. This article uses Windows Authentication, but SQL Server login is also supported. If you select **SQL Login**, you will be prompted for a username and password. For more information about authentication types, see [Connect to server \(database engine\)](#).

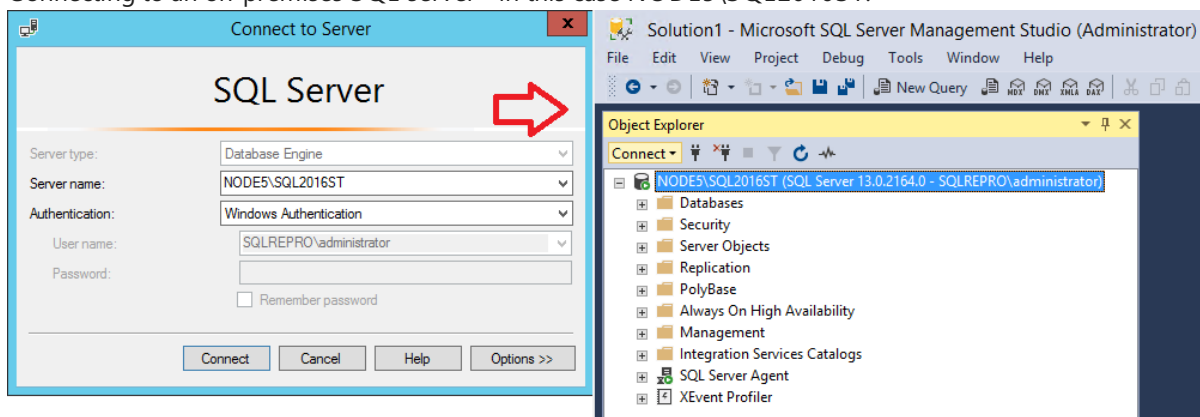
You can also modify additional connection options by selecting **Options**. Examples of connection options are the database you're connecting to, the connection timeout value, and the network protocol. This article uses the default values for all the options.

3. After you've completed all the fields, select **Connect**.

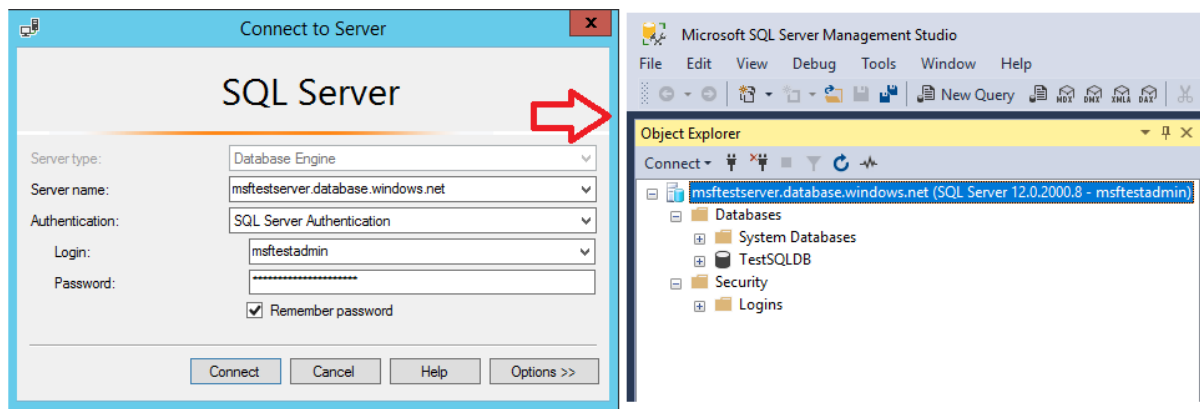
Examples of successful connections

To verify that your SQL Server connection succeeded, expand and explore the objects within **Object Explorer**. These objects will be different depending on the type of server you're connected to.

- Connecting to an on-premises SQL server - in this case NODE5\SQL2016ST:



- Connecting to SQL Azure DB - in this case msfttestserver.database.windows.net:



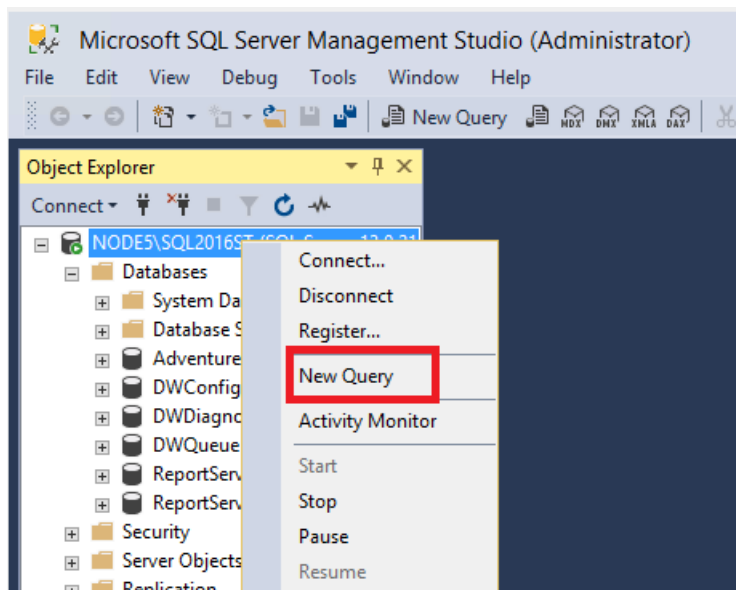
NOTE

In this tutorial, you previously used *Windows Authentication* to connect to your on-premises SQL server, but this method is not supported for SQL Azure DB. As such, this image shows using SQL Authentication to connect to the SQL Azure DB. For more information, see [SQL on-premises authentication](#) and [SQL Azure authentication](#).

Create a database

Create a database named TutorialDB by doing the following:

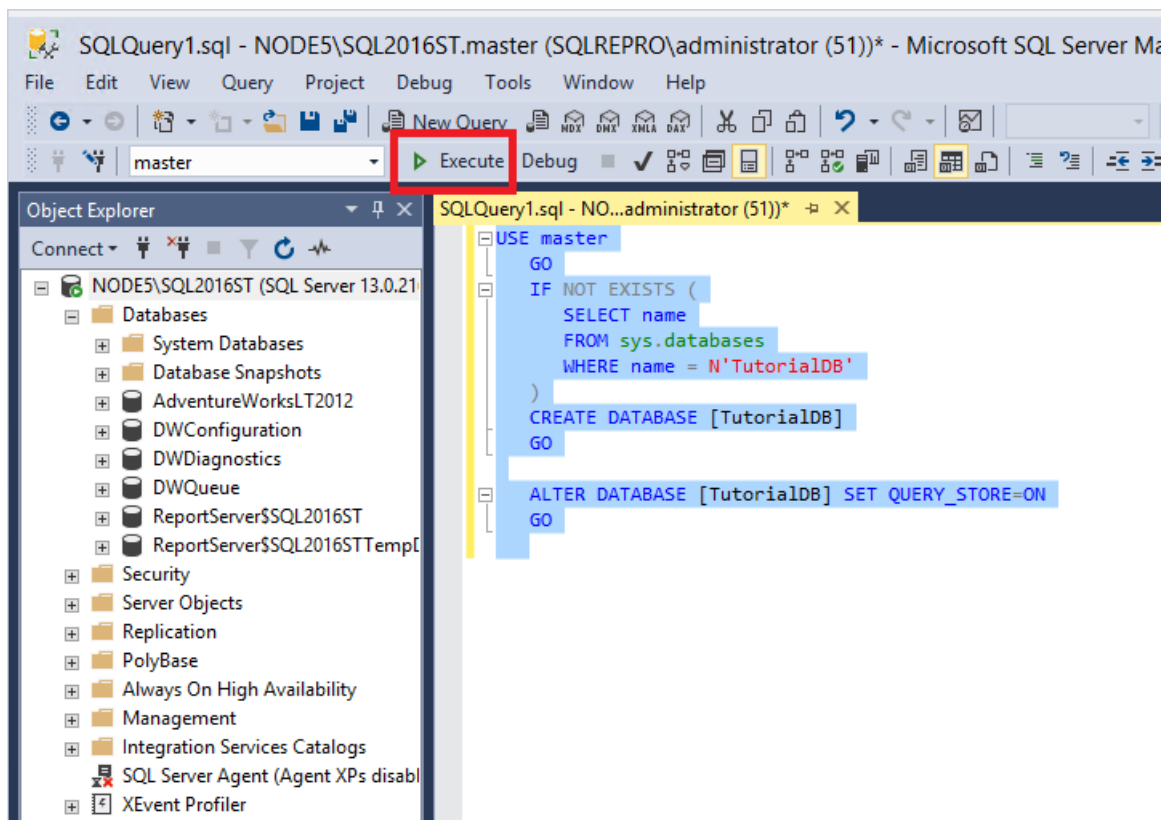
1. Right-click your server instance in Object Explorer, and then select **New Query**:



2. Into the query window, paste the following T-SQL code snippet:

```
USE master
GO
IF NOT EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'TutorialDB'
)
CREATE DATABASE [TutorialDB]
GO
```

3. To execute the query, select **Execute** (or select F5 on your keyboard).

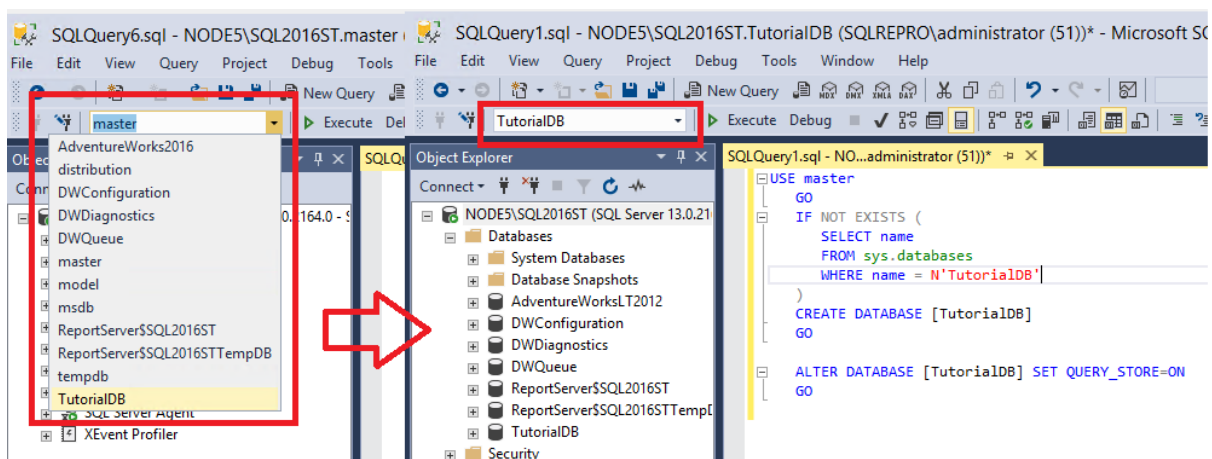


After the query is complete, the new TutorialDB database appears in the list of databases in Object Explorer. If it isn't displayed, right-click the **Databases** node, and then select **Refresh**.

Create a table in the new database

In this section, you create a table in the newly created TutorialDB database. Because the query editor is still in the context of the *master* database, switch the connection context to the *TutorialDB* database by doing the following:

1. In the database drop-down list, select the database that you want, as shown here:



2. Paste the following T-SQL code snippet into the query window, select it, and then select **Execute** (or select F5 on your keyboard).

You can either replace the existing text in the query window or append it to the end. To execute everything in the query window, select **Execute**. To execute a portion of the text, highlight that portion, and then select **Execute**.

```
-- Create a new table called 'Customers' in schema 'dbo'
-- Drop the table if it already exists
IF OBJECT_ID('dbo.Customers', 'U') IS NOT NULL
DROP TABLE dbo.Customers
GO
-- Create the table in the specified schema
CREATE TABLE dbo.Customers
(
    CustomerId      INT      NOT NULL    PRIMARY KEY, -- primary key column
    Name            [NVARCHAR](50) NOT NULL,
    Location        [NVARCHAR](50) NOT NULL,
    Email           [NVARCHAR](50) NOT NULL
);
GO
```

After the query is complete, the new Customers table is displayed in the list of tables in Object Explorer. If the table is not displayed, right-click the **TutorialDB > Tables** node in Object Explorer, and then select **Refresh**.

Insert rows into the new table

Insert some rows into the Customers table that you created previously. To do so, paste the following T-SQL code snippet into the query window, and then select **Execute**:

```
-- Insert rows into table 'Customers'
INSERT INTO dbo.Customers
([CustomerId],[Name],[Location],[Email])
VALUES
( 1, N'Orlando', N'Australia', N''),
( 2, N'Keith', N'India', N'keith0@adventure-works.com'),
( 3, N'Donna', N'Germany', N'donna0@adventure-works.com'),
( 4, N'Janet', N'United States', N'janet1@adventure-works.com')
GO
```

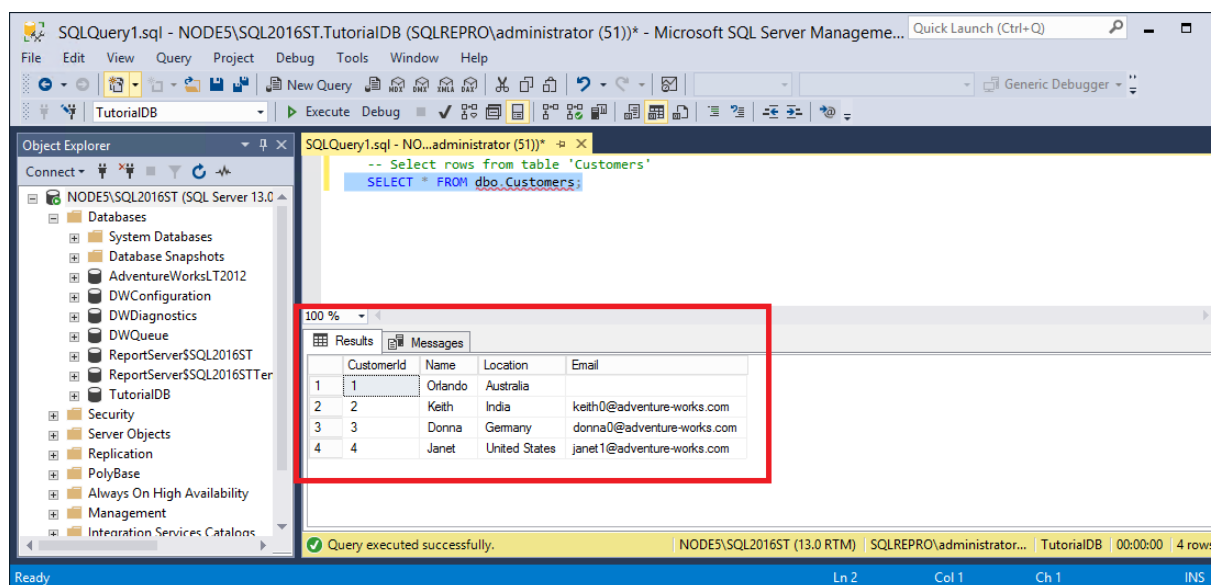
Query the table and view the results

The results of a query are visible below the query text window. To query the Customers table and view the rows that were previously inserted, do the following:

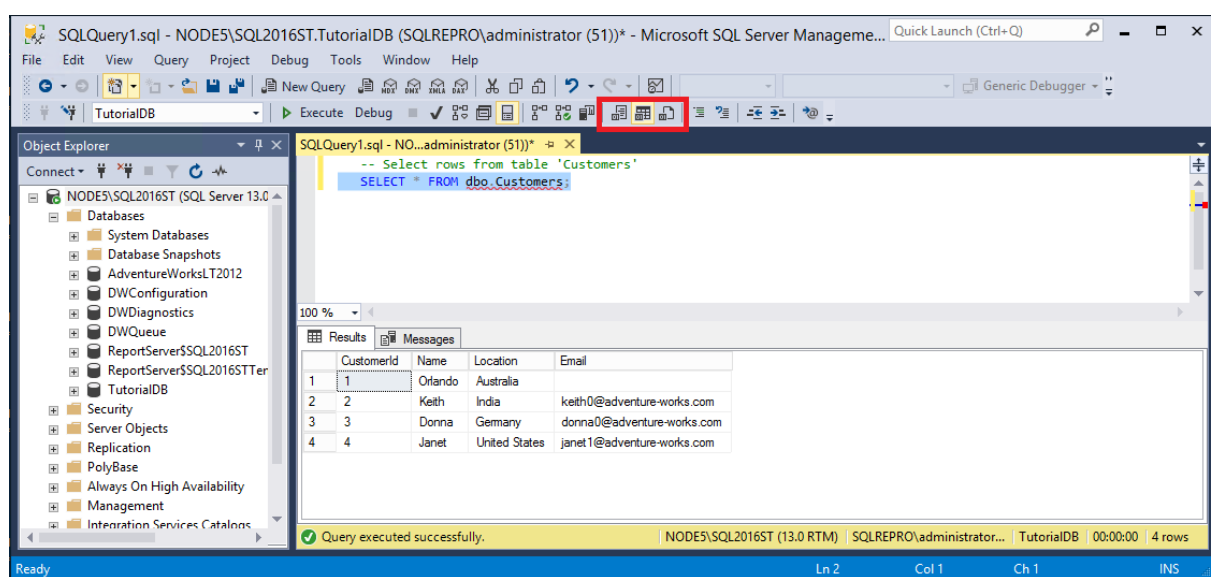
1. Paste the following T-SQL code snippet into the query window, and then select **Execute**:

```
-- Select rows from table 'Customers'
SELECT * FROM dbo.Customers;
```

The results of the query are displayed under the area where text was entered:



2. Modify the way results are presented by selecting one of the following options:

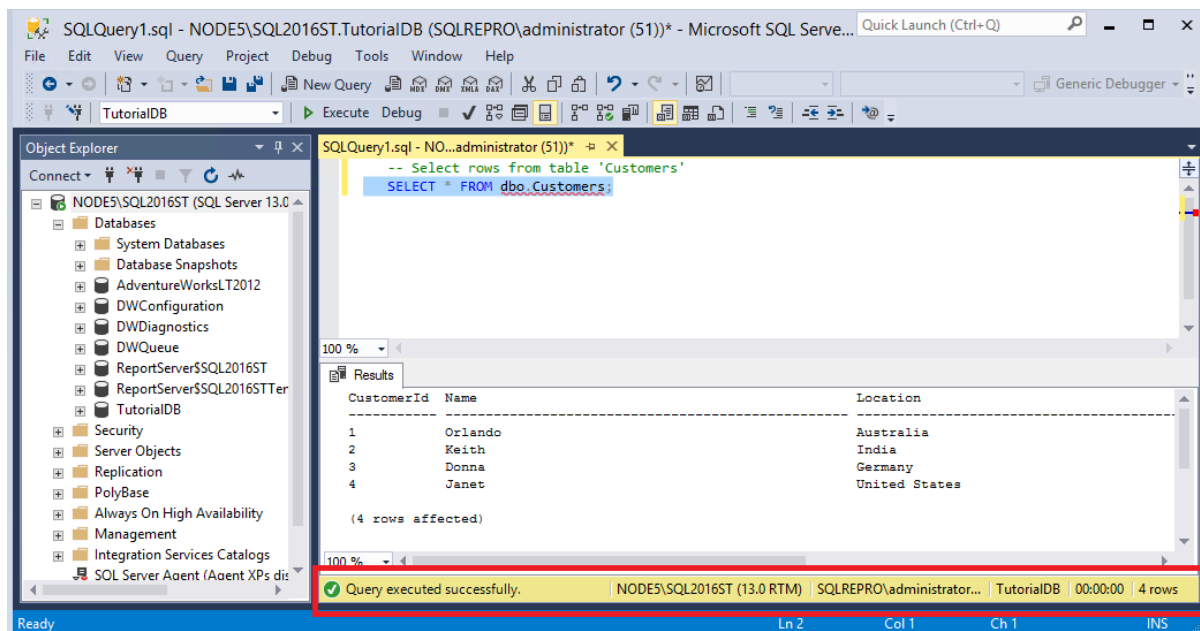


- The middle button displays the results in **Grid View**, which is the default option.
- The first button displays the results in **Text View**, as shown in the image in the next section.
- The third button lets you save the results to a file whose extension is .rpt by default.

Verify your connection properties by using the query window table

You can find information about the connection properties under the results of your query. After you run the previously mentioned query in the preceding step, review the connection properties at the bottom of the query window.

- You can determine which server and database you're connected to, and the username that you're logged in with.
- You can also view the query duration and the number of rows that are returned by the previously executed query.

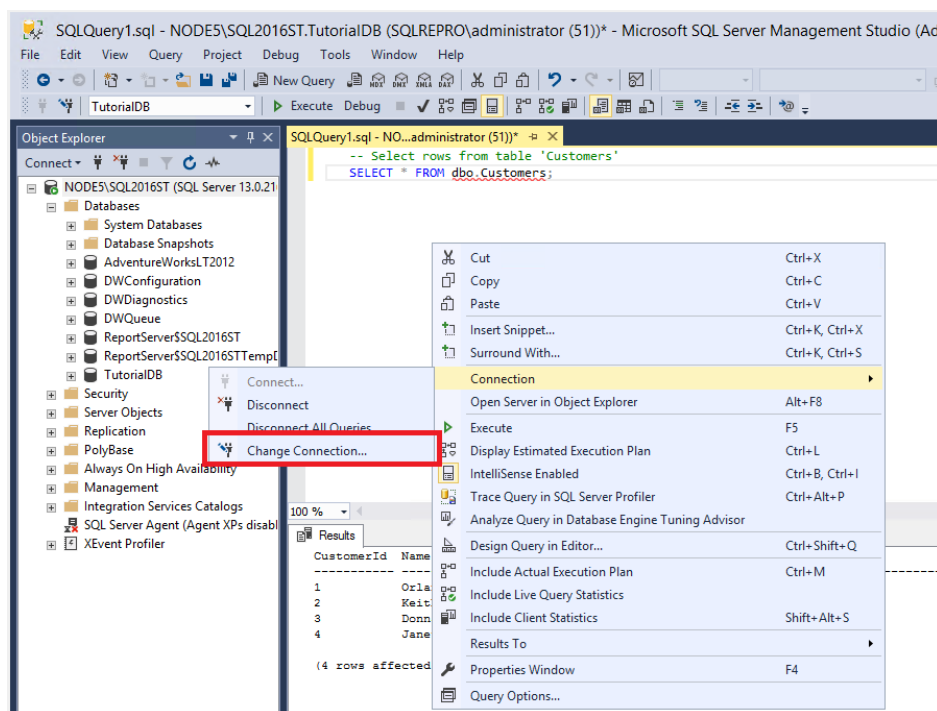


In the image, note that the results are displayed in **Text View**.

Change the server that the query window is connected to

You can change the server that your current query window is connected to by doing the following:

1. Right-click in the query window, and then select **Connection** > **Change connection**. The **Connect to Server** window opens again.
2. Change the server that your query is connected to.



NOTE

This action changes only the server that the query window is connected to, not the server that Object Explorer is connected to.

Next steps

The next article teaches you how to script various objects within SQL Server Managment Studio.

Go to the next article to learn more:

[Next steps](#)

Tutorial: Script objects in SQL Server Management Studio

10/1/2018 • 5 minutes to read • [Edit Online](#)

This tutorial teaches you to generate Transact-SQL (T-SQL) scripts for various objects found within SQL Server Management Studio (SSMS). In this tutorial, you find examples of how to script the following objects:

- Queries, when you perform actions within the GUI
- Databases in two different ways (Script As and Generate Script)
- Tables
- Stored procedures
- Extended events

To script any object in **Object Explorer**, right-click it and select the **Script Object As** option. This tutorial shows you the process.

Prerequisites

To complete this tutorial, you need SQL Server Management Studio, access to a server that's running SQL Server, and an AdventureWorks database.

- Install [SQL Server Management Studio](#).
- Install [SQL Server 2017 Developer Edition](#).
- Download [AdventureWorks2016 sample databases](#).

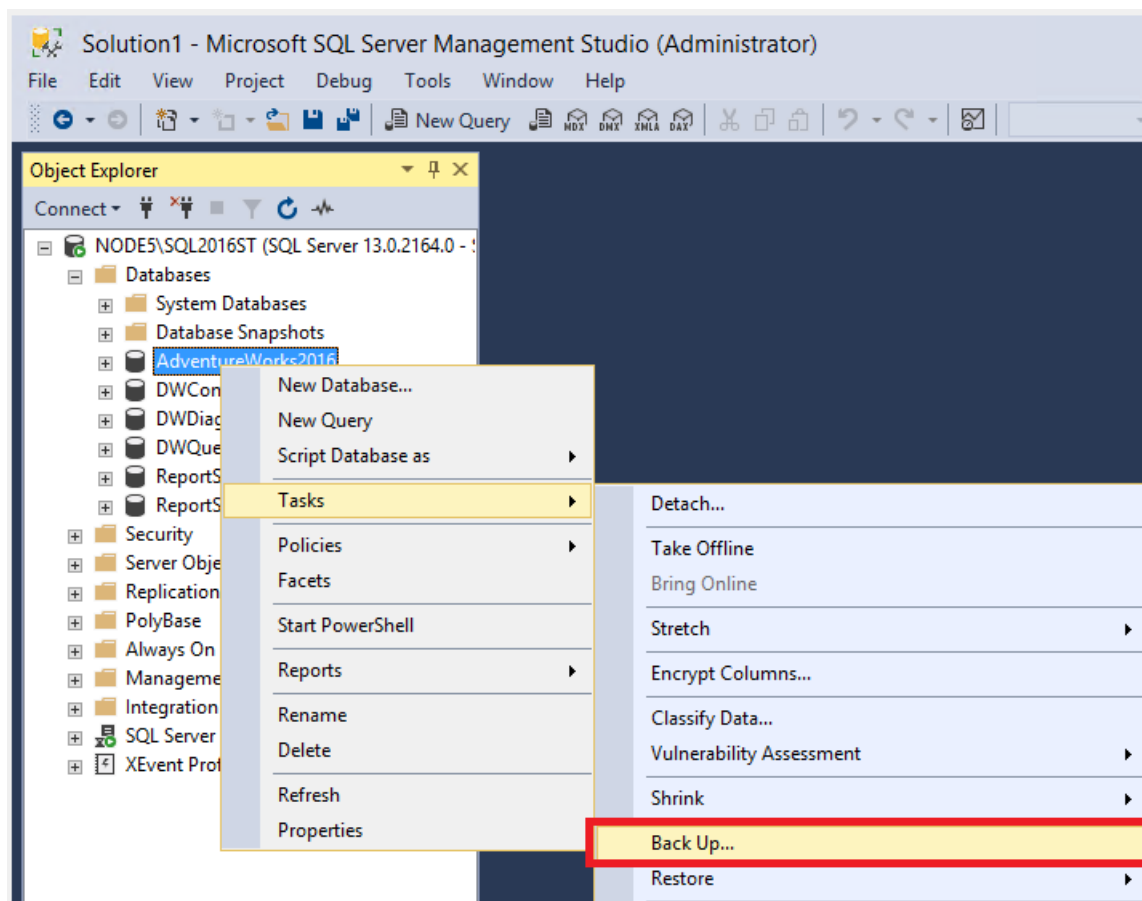
Instructions for restoring databases in SSMS are here: [Restore a database](#).

Script queries from the GUI

You can generate the associated T-SQL code for a task whenever you use the GUI in SSMS to complete it. The following examples show how to do so when you back up a database and when you shrink the transaction log. These same steps can be applied to any action that's completed via the GUI.

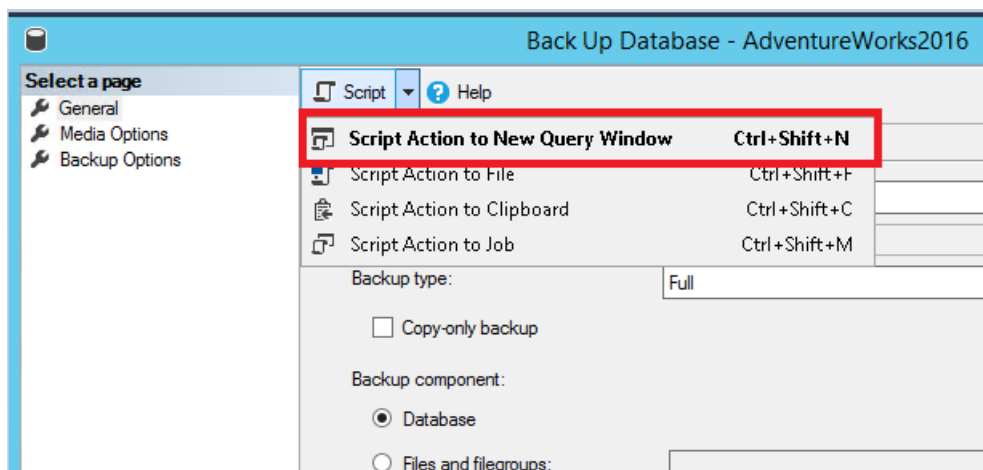
Script T-SQL when you back up a database

1. Connect to a server that's running SQL Server.
2. Expand the **Databases** node.
3. Right-click the database **Adventureworks2016** > **Tasks** > **Back Up**:

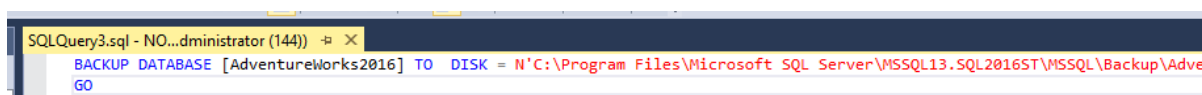


4. Configure the backup the way you want. For this tutorial, everything is left at default. However, any changes made in the window also reflect in the script.

5. Select **Script > Script Action to New Query Window**:



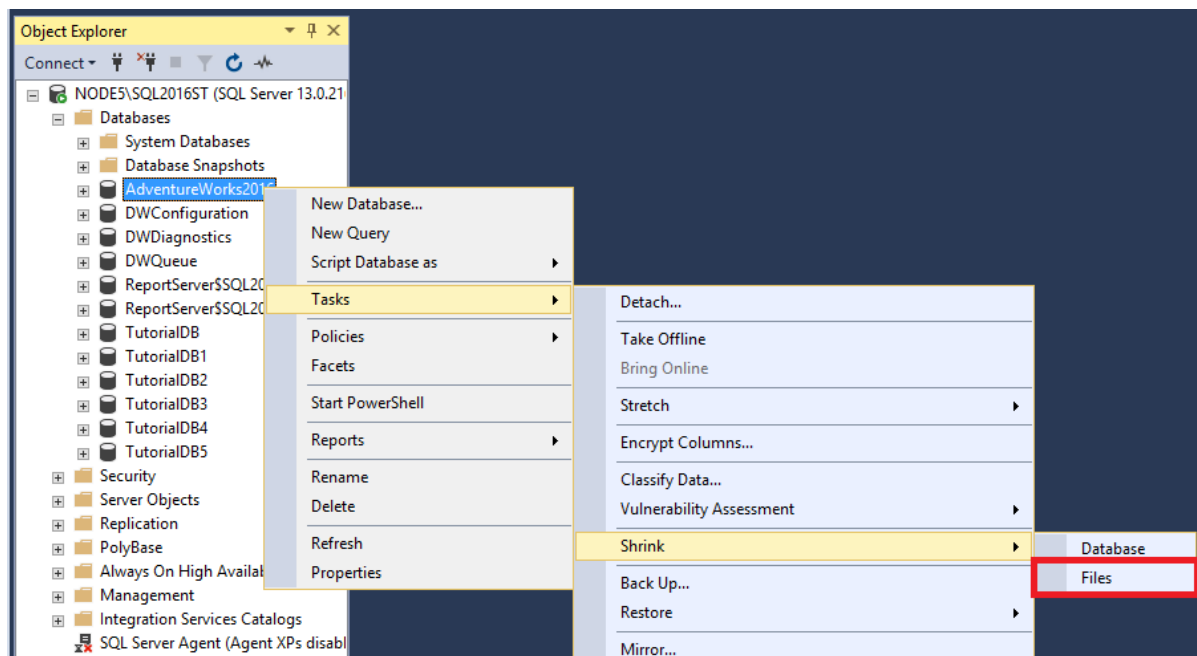
6. Review the T-SQL populated in the query window.



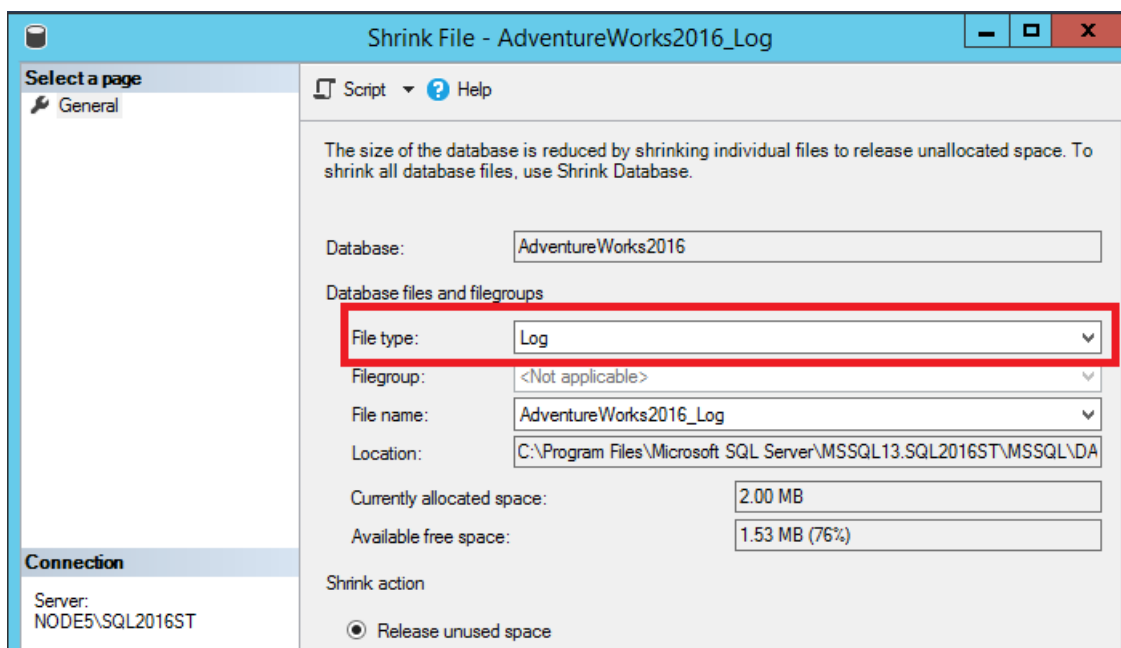
7. Select **Execute** to execute the query to back up the database via T-SQL.

Script T-SQL when you shrink the transaction log

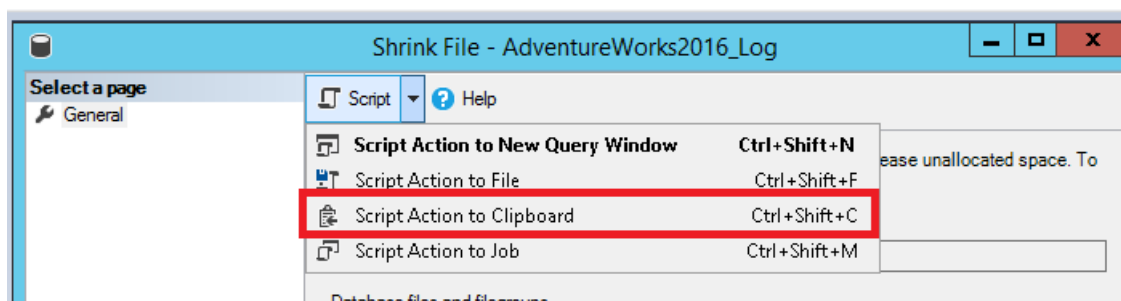
1. Right-click the database **AdventureWorks2016 > Tasks > Shrink > Files**:



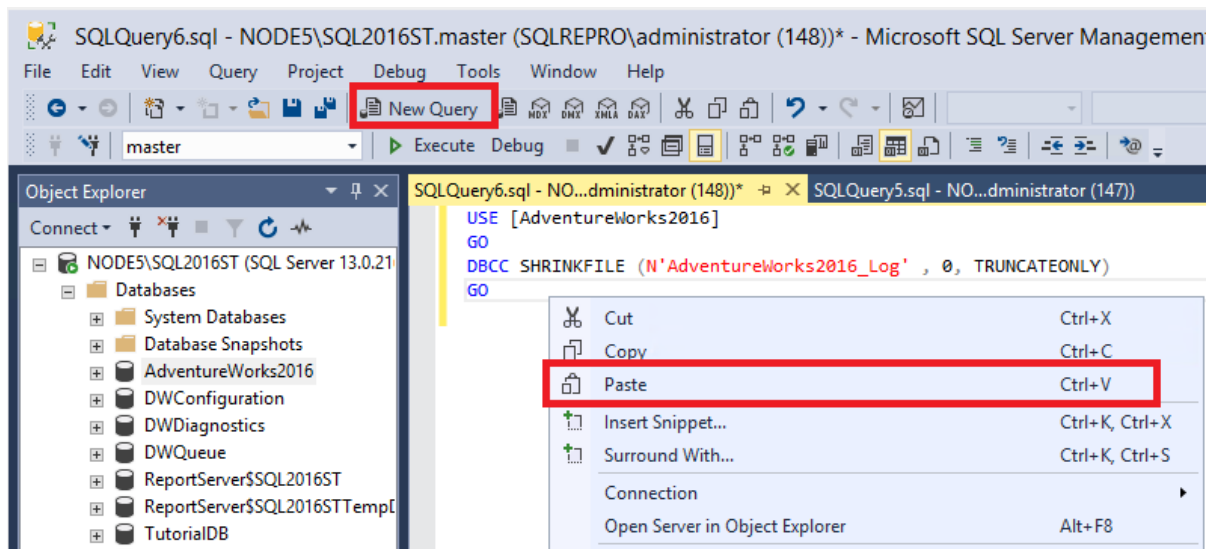
2. Select **Log** from the **File type** drop-down list box:



3. Select **Script** and **Script Action to Clipboard**:



4. Open a **New Query** window and paste. (Right-click in the window. Then select **Paste**.)



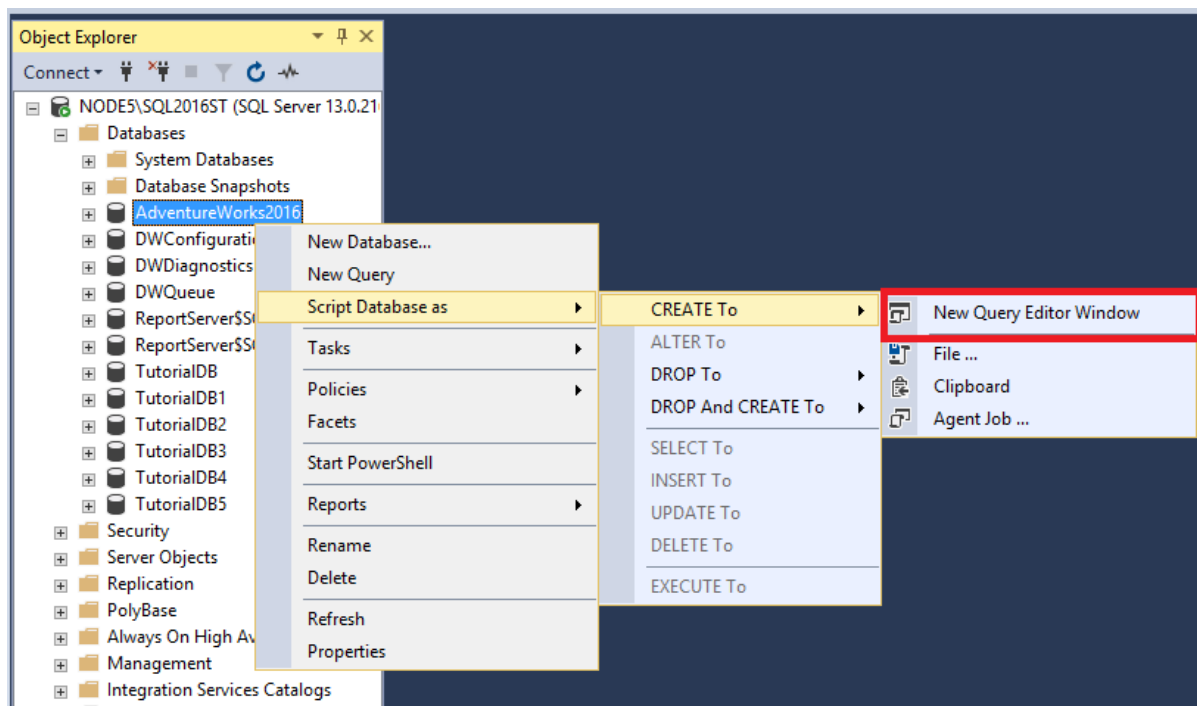
5. Select **Execute** to execute the query and shrink the transaction log.

Script databases

The following section teaches you to script out the database by using the **Script As** and **Generate Scripts** options. The **Script As** option re-creates the database and its configuration options. You can script both the schema and the data by using the **Generate Scripts** option. In this section, you create two new databases. You use the **Script As** option to create *AdventureWorks2016a*. You use the **Generate Scripts** option to create *AdventureWorks2016b*.

Script a database by using the Script option

1. Connect to a server that's running SQL Server.
2. Expand the **Databases** node.
3. Right-click the database **AdventureWorks2016** > **Script Database As** > **Create To** > **New Query Editor Window**:



4. Review the database creation query in the window:

```

SQLQuery7.sql - NO...dministrator (143))
USE [master]
GO

/***** Object: Database [AdventureWorks2016]    Script Date: 3/16/2018 10:04:02 PM *****/
CREATE DATABASE [AdventureWorks2016]
CONTAINMENT = NONE
ON PRIMARY
( NAME = N'AdventureWorks2016_Data', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL13.SQL2016ST\MSSQL\DATA\AdventureWorks2016_
LOG ON
( NAME = N'AdventureWorks2016_Log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL13.SQL2016ST\MSSQL\DATA\AdventureWorks2016_L
GO

ALTER DATABASE [AdventureWorks2016] SET COMPATIBILITY_LEVEL = 130
GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [AdventureWorks2016].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO

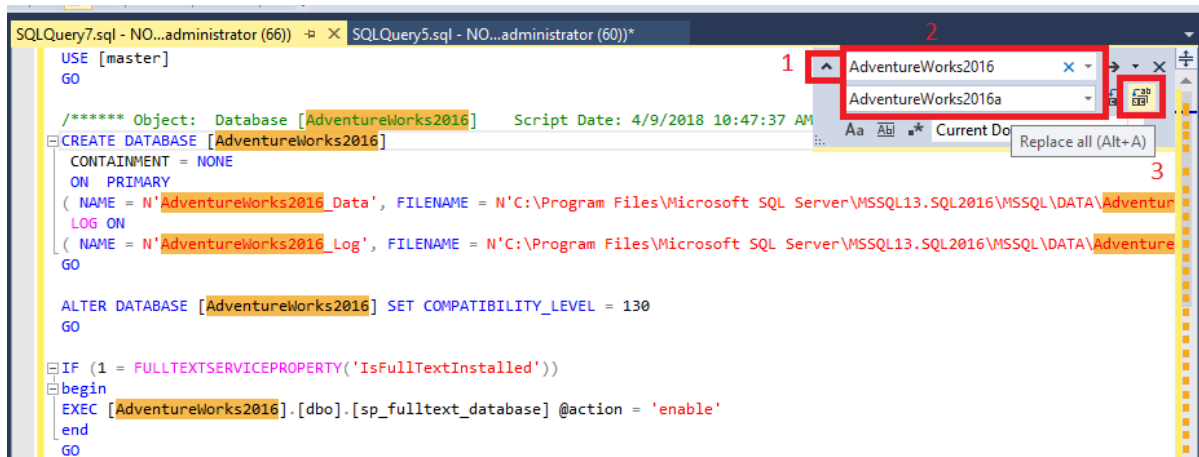
ALTER DATABASE [AdventureWorks2016] SET ANSI_NULL_DEFAULT OFF
GO

ALTER DATABASE [AdventureWorks2016] SET ANSI_NULLS ON
GO

```

This option scripts out only the database configuration options.

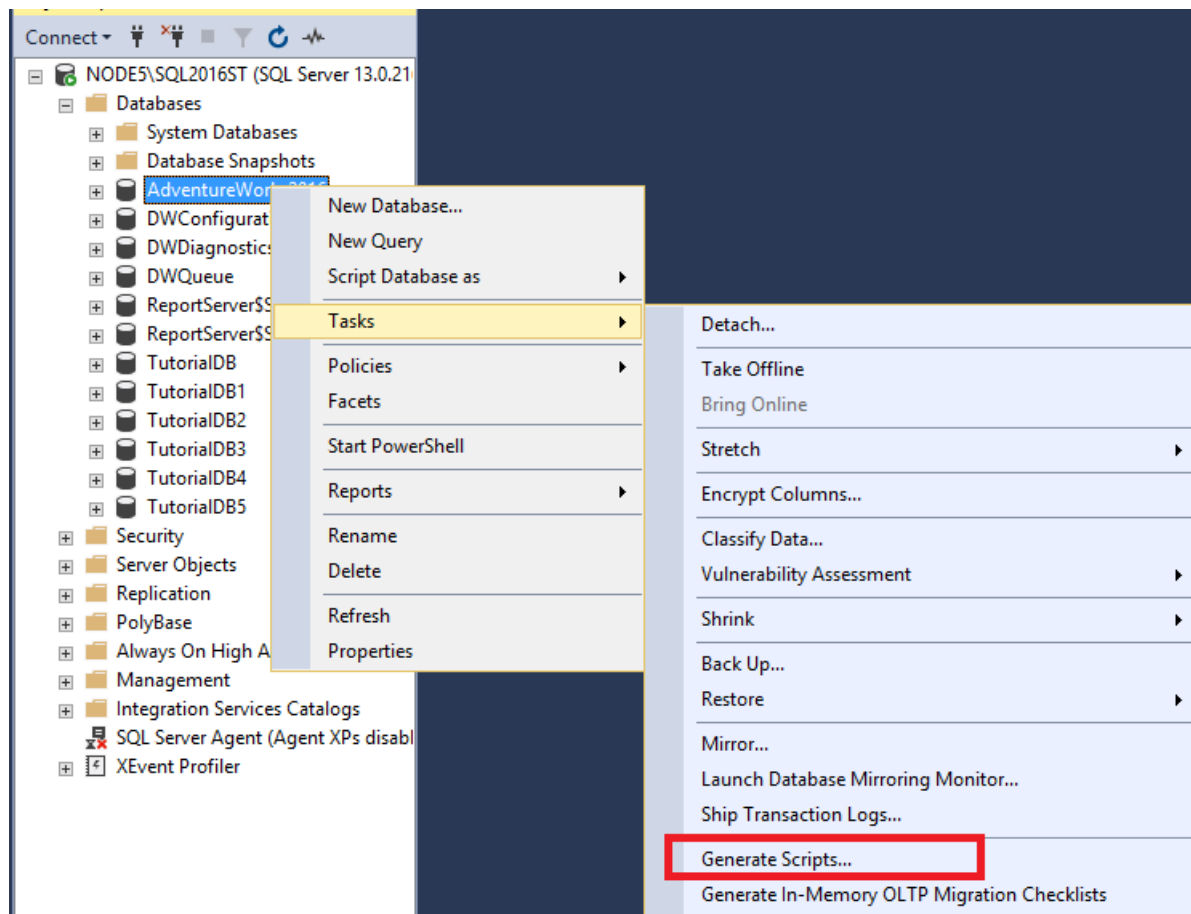
- On your keyboard, select Ctrl+F to open the **Find** dialog box. Select the down arrow to open the **Replace** option. On the top **Find** line, type AdventureWorks2016, and on the bottom **Replace** line, type AdventureWorks2016a.
- Select **Replace All** to replace all instances of *AdventureWorks2016* with *AdventureWorks2016a*.



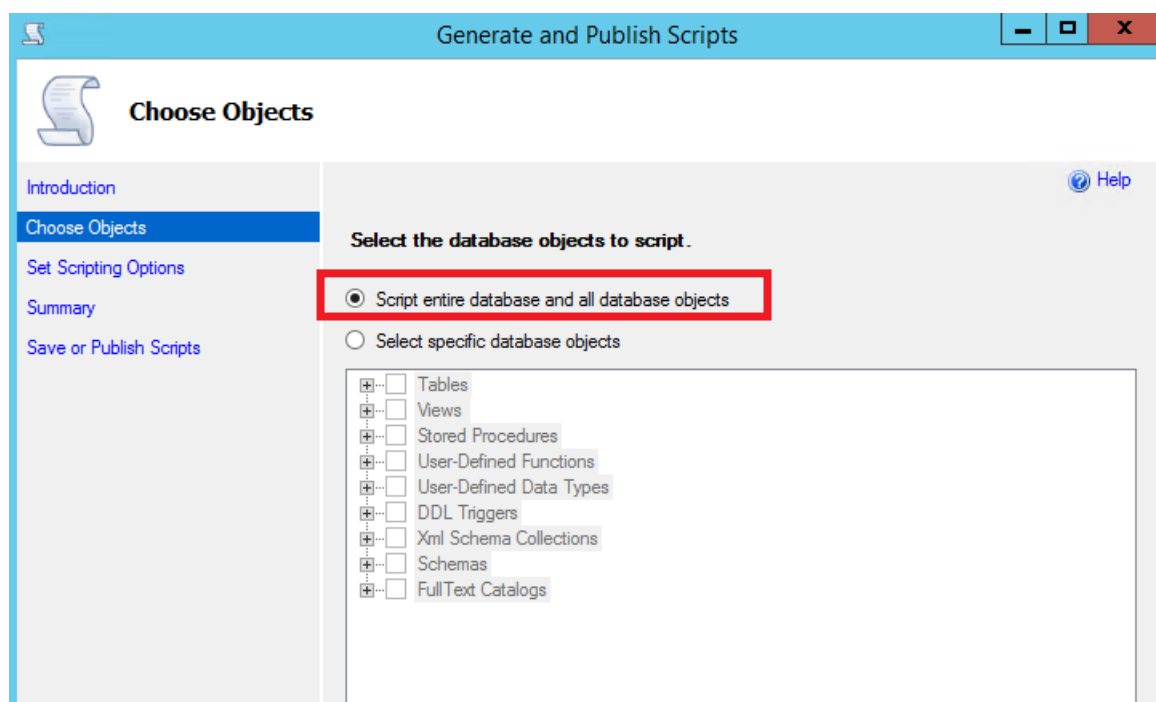
- Select **Execute** to execute the query and create your new AdventureWorks2016a database.

Script a database by using the Generate Scripts option

- Connect to a server that's running SQL Server.
- Expand the **Databases** node.
- Right-click **AdventureWorks2016** > **Tasks** > **Generate Scripts**:

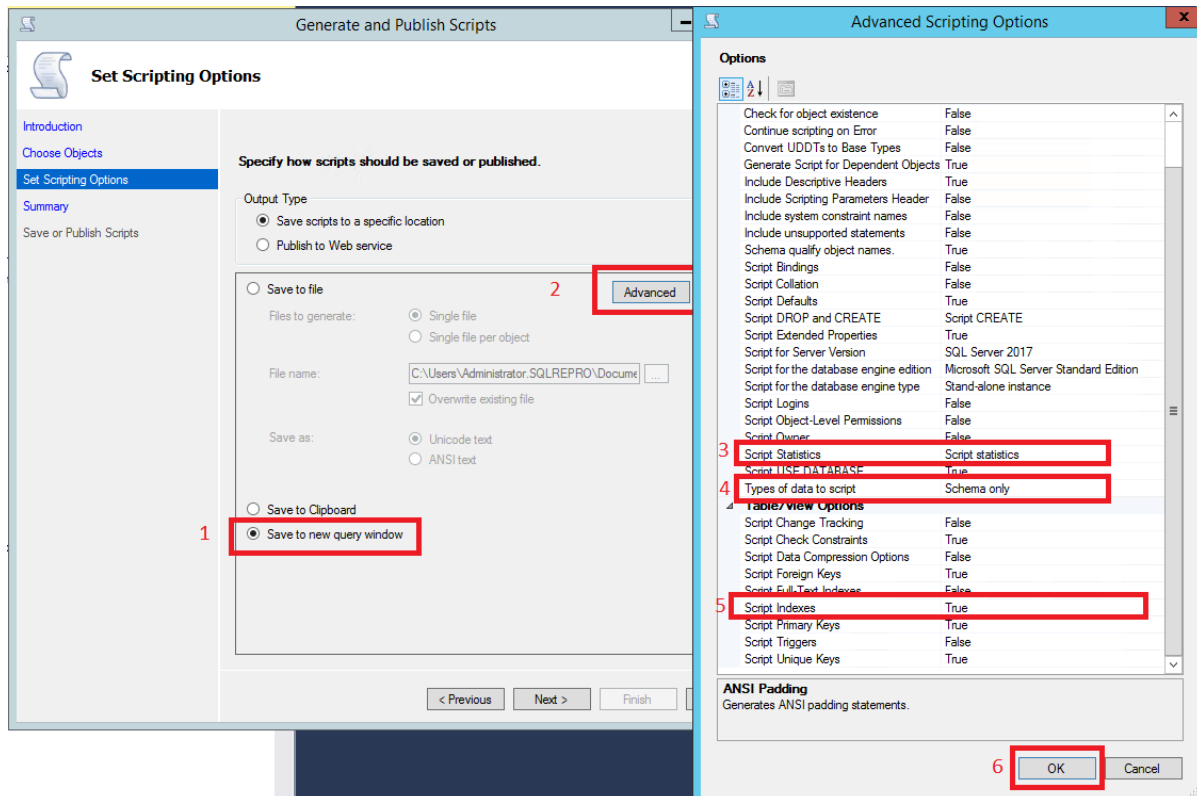


4. The **Introduction** page opens. Select **Next** to open the **Chose Objects** page. You can select the entire database or specific objects in the database. Select **Script entire database and all database objects**.



5. Select **Next** to open the **Set Scripting Options** page. Here you can configure where to save the script and some additional advanced options.
 - a. Select **Save to new query window**.
 - b. Select **Advanced** and make sure these options are set:
 - **Script Statistics** set to *Script Statistics*.
 - **Types of data to script** set to *Schema only*.

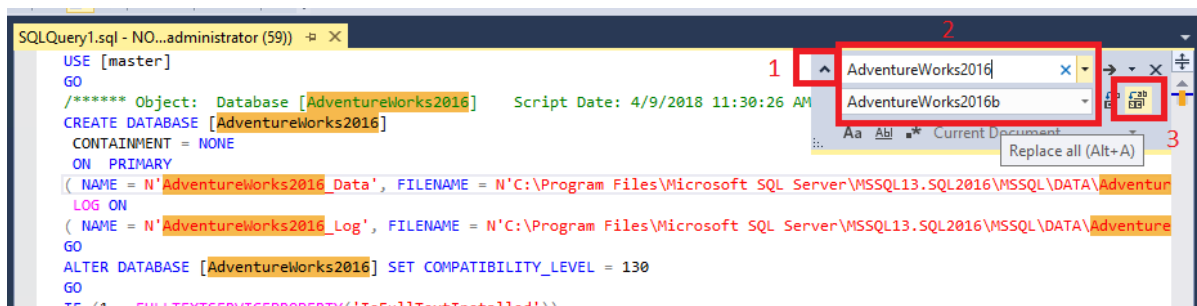
- **Script Indexes** set to *True*.



NOTE

You can script the data for the database when you select *Schema and data* for the **Types of data to script** option. However, this isn't ideal with large databases. It can take more memory than SSMS can allocate. This limitation is okay for small databases. If you want to move data for a larger database, use the [Import and Export Wizard](#).

6. Select **OK**, and then select **Next**.
7. Select **Next** on the **Summary**. Then select **Next** again to generate the script in a **New Query** window.
8. On your keyboard, open the **Find** dialog box (Ctrl+F). Select the down arrow to open the **Replace** option. On the top **Find** line, enter *AdventureWorks2016*. On the bottom **Replace** line, enter *AdventureWorks2016b*.
9. Select **Replace All** to replace all instances of *AdventureWorks2016* with *AdventureWorks2016b*.

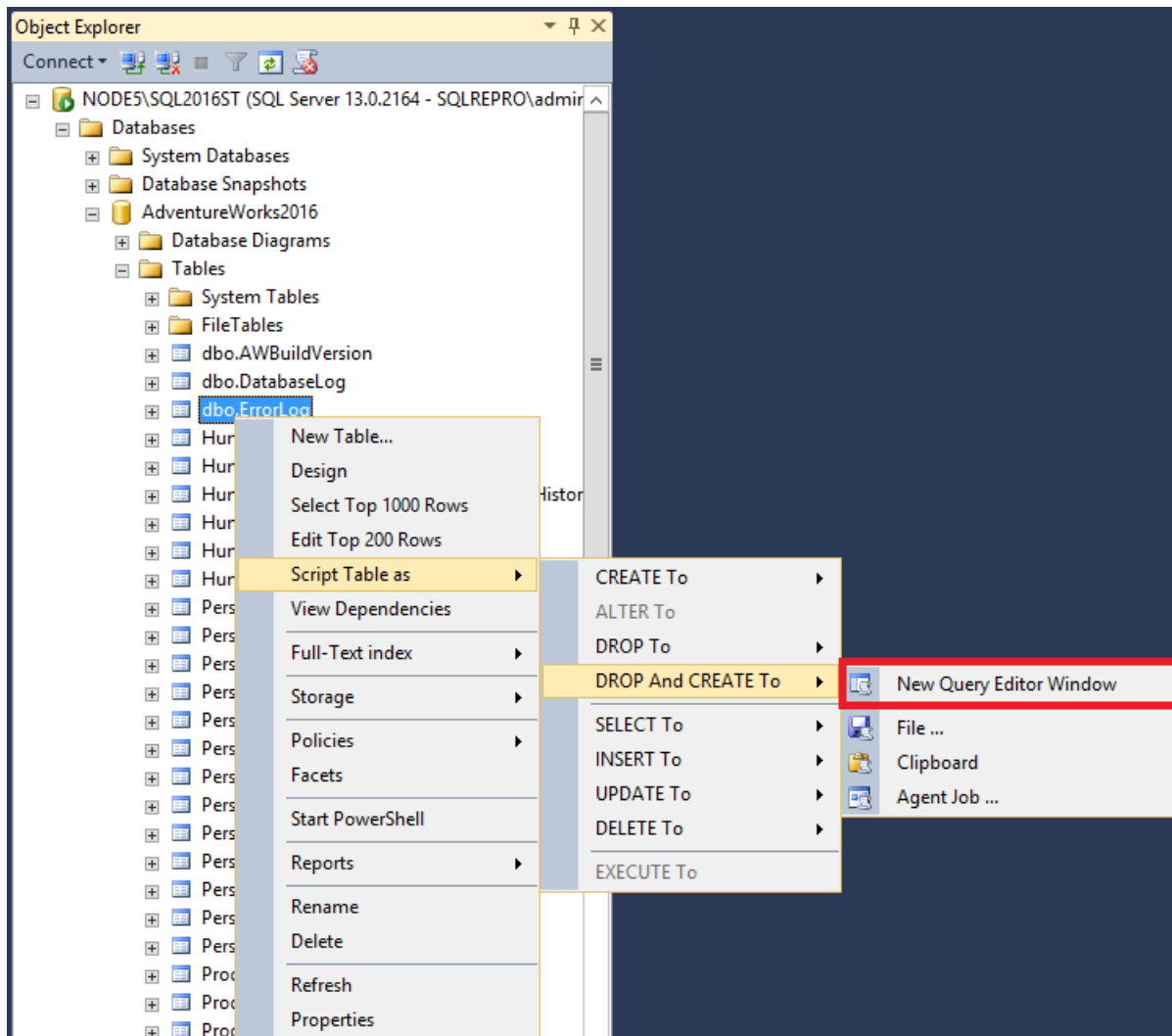


10. Select **Execute** to execute the query and create your new AdventureWorks2016b database.

Script tables

This section covers how to script out tables from your database. Use this option to either create the table or drop and create the table. You can also use this option to script the T-SQL associated with modifying the table. An example is to insert into it or update to it. In this section, you drop a table and then re-create it.

1. Connect to a server that's running SQL Server.
2. Expand your **Databases** node.
3. Expand your **AdventureWorks2016** database node.
4. Expand your **Tables** node.
5. Right-click **dbo.ErrorLog** > **Script Table as** > **DROP And CREATE To** > **New Query Editor Window**:



6. Select **Execute** to execute the query. This action drops the *Errorlog* table and re-creates it.

NOTE

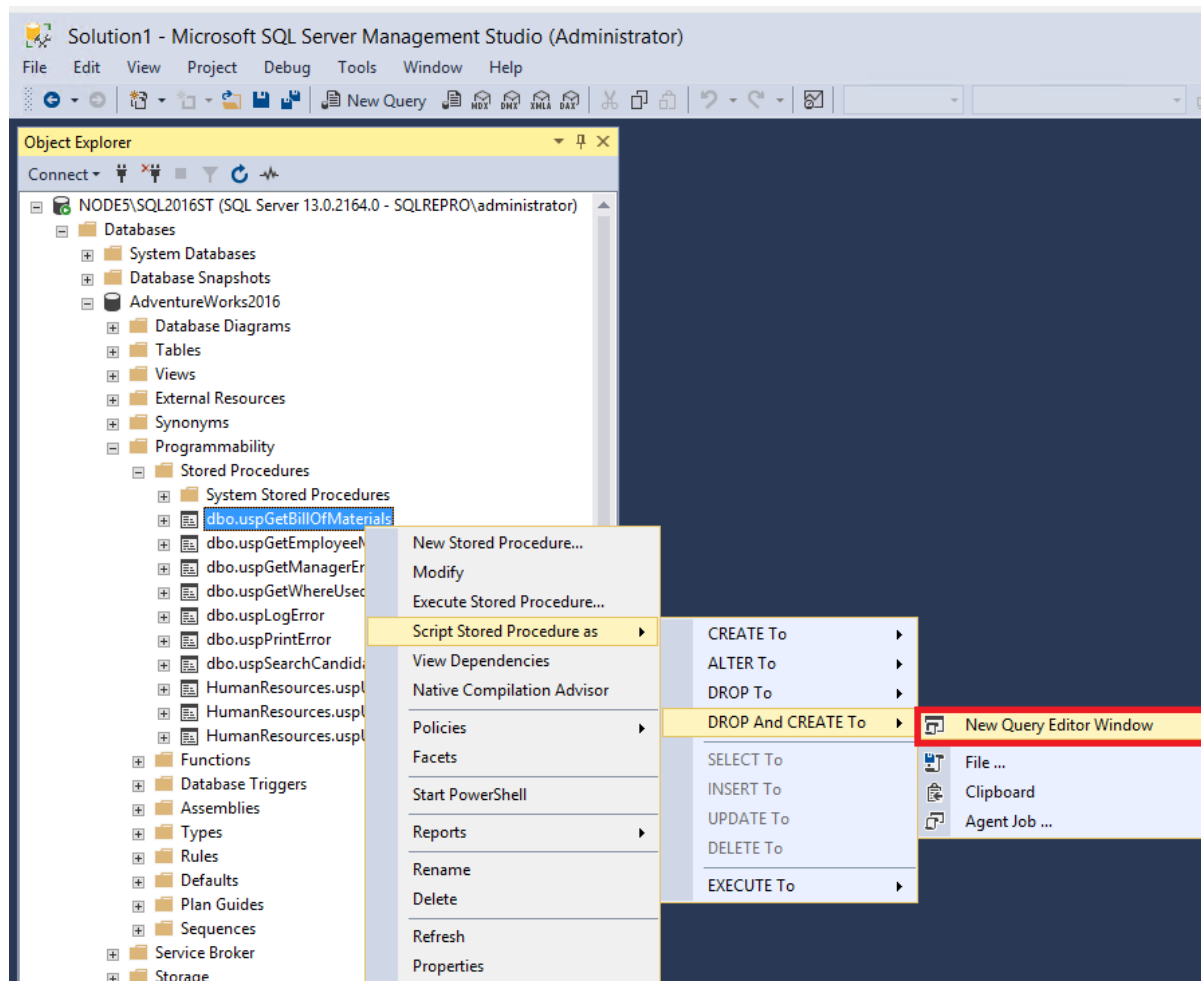
The *Errorlog* table is empty by default in the AdventureWorks2016 database. So you're not losing any data by dropping the table. However, following these steps on a table with data causes data loss.

Script stored procedures

In this section, you'll learn how to drop and create a stored procedure.

1. Connect to a server that's running SQL Server.
2. Expand your **Databases** node.
3. Expand your **Programmability** node.
4. Expand your **Stored Procedure** node.

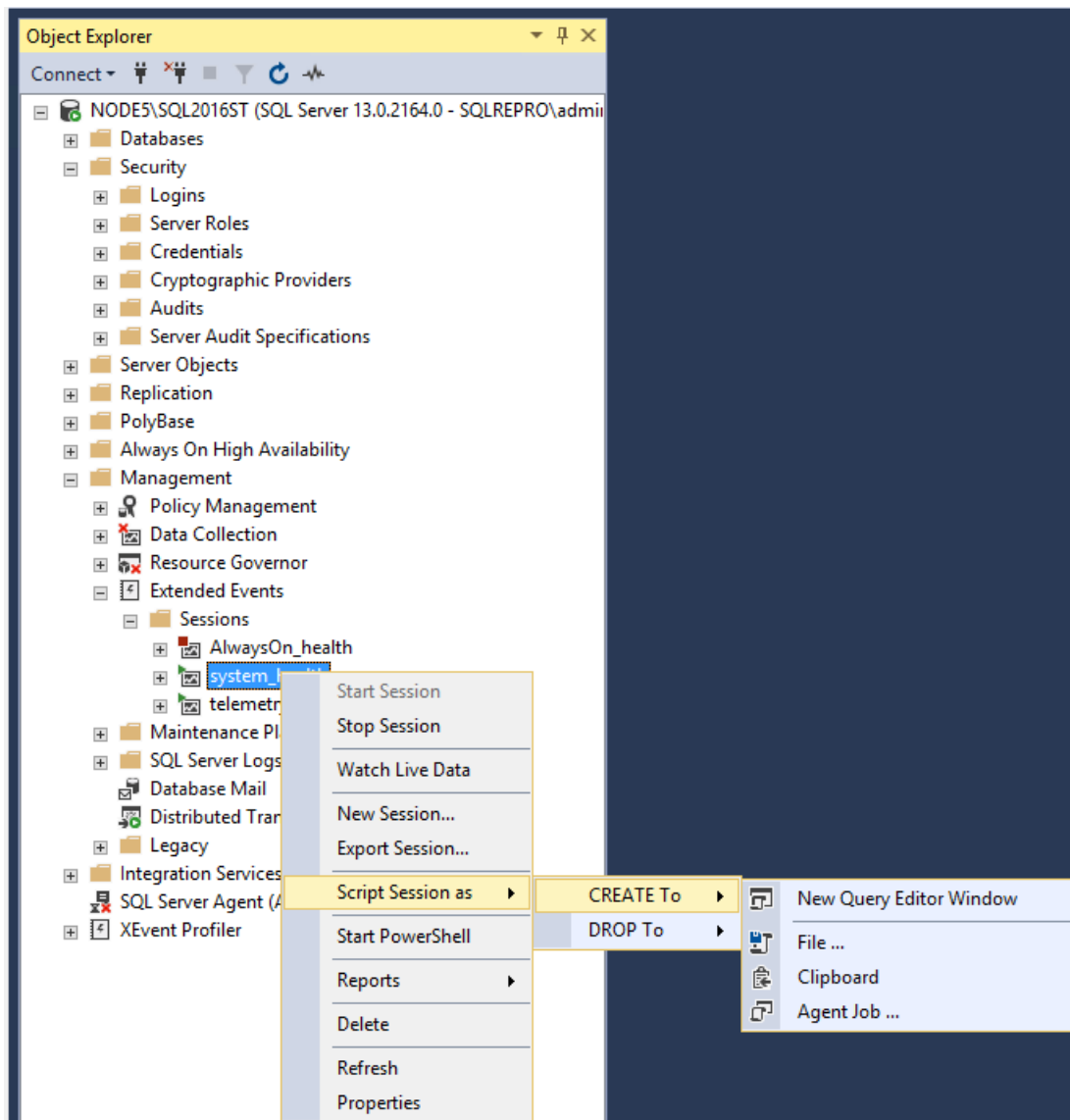
5. Right-click the stored procedure **dbo.uspGetBillOfMaterials** > **Script Stored Procedure As** > **DROP and CREATE To** > **New Query Editor Window**:



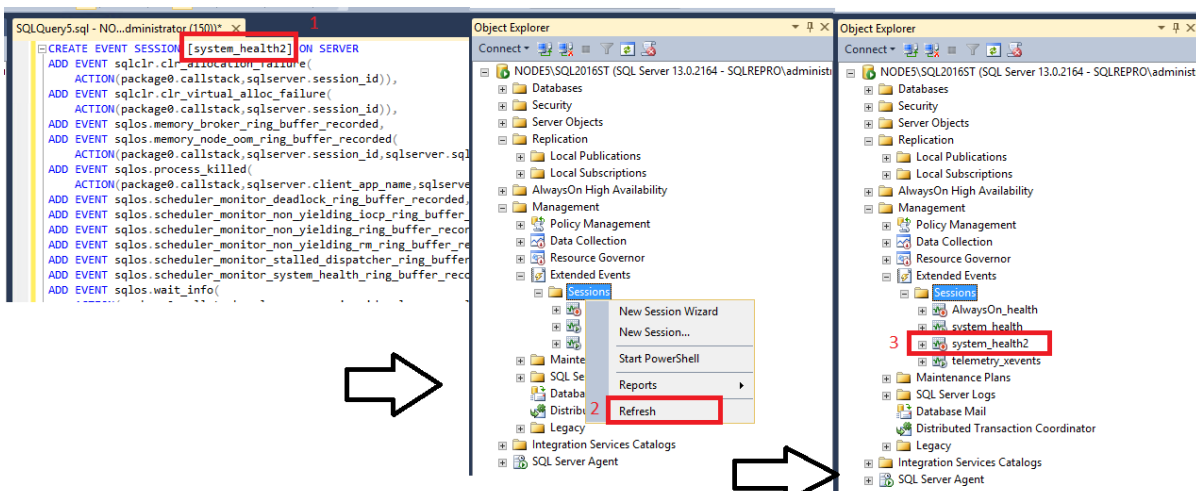
Script extended events

This section covers how to script out [extended events](#).

1. Connect to a server that's running SQL Server.
2. Expand your **Management** node.
3. Expand your **Extended Events** node.
4. Expand your **Sessions** node.
5. Right-click the extended session you're interested in > **Script Session As** > **New Query Editor Window**:



6. In the **New Query Editor Window**, modify the new name of the session from *system_health* to *system_health2*. Select **Execute** to execute the query.
7. Right-click **Sessions** in **Object Explorer**. Select **Refresh** to see your new extended event session. The green icon next to the session indicates the session is running. The red icon indicates the session is stopped.



NOTE

You can start the session by right-clicking it and selecting **Start**. However, this is a copy of the already running **system_health** session, so you can skip this step. You can delete the copy of the extended event session: right-click it and select **Delete**.

Next steps

The next article introduces you to the prebuilt T-SQL templates found within SSMS.

Go to the next article to learn more:

[Next steps](#)

Tutorial: Using templates in SQL Server Management Studio

10/1/2018 • 2 minutes to read • [Edit Online](#)

This tutorial introduces you to the prebuilt Transact-SQL (T-SQL) templates that are available in SQL Server Management Studio (SSMS). In this article, you learn how to:

- Use the template browser to generate T-SQL scripts
- Edit an existing template
- Locate templates on disk
- Create a new template

Prerequisites

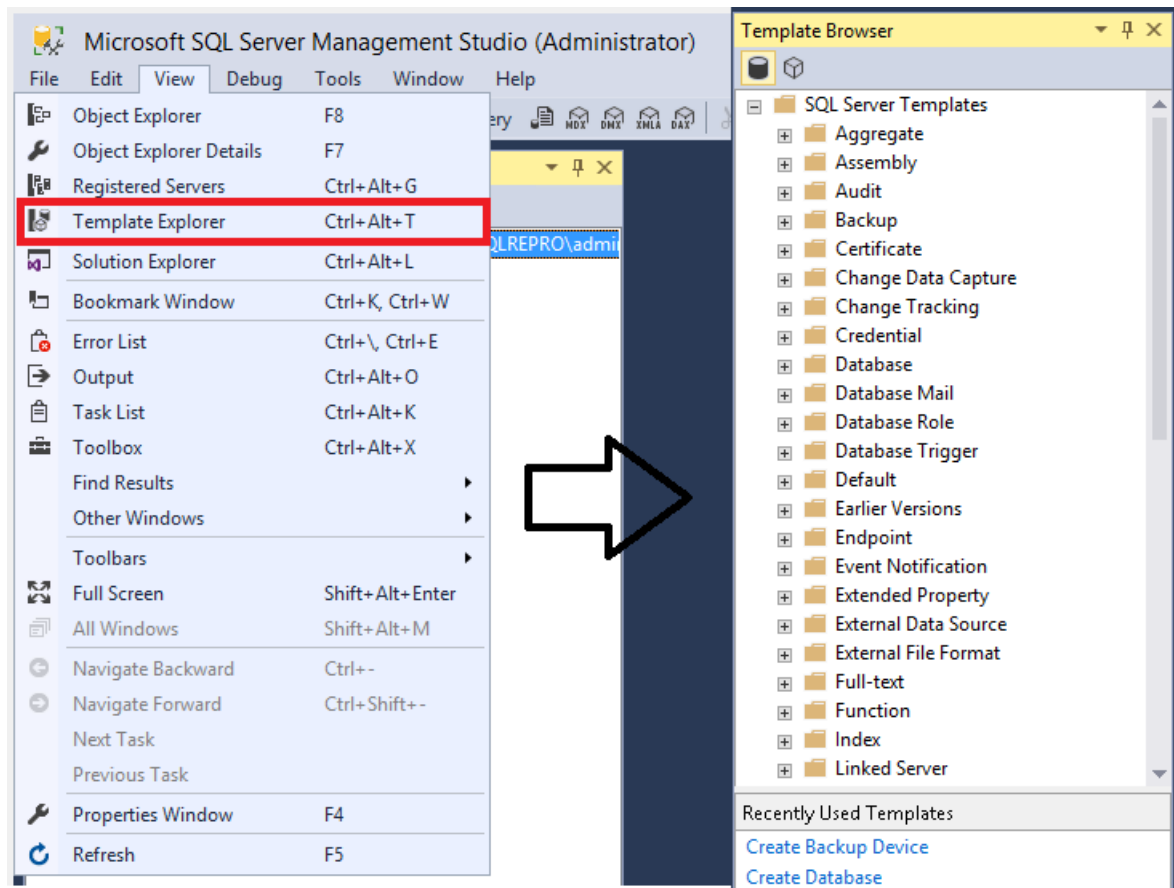
To complete this tutorial, you need SQL Server Management Studio and access to a SQL server.

- Install [SQL Server Management Studio](#).
- Install [SQL Server 2017 Developer Edition](#).

Use Template Browser

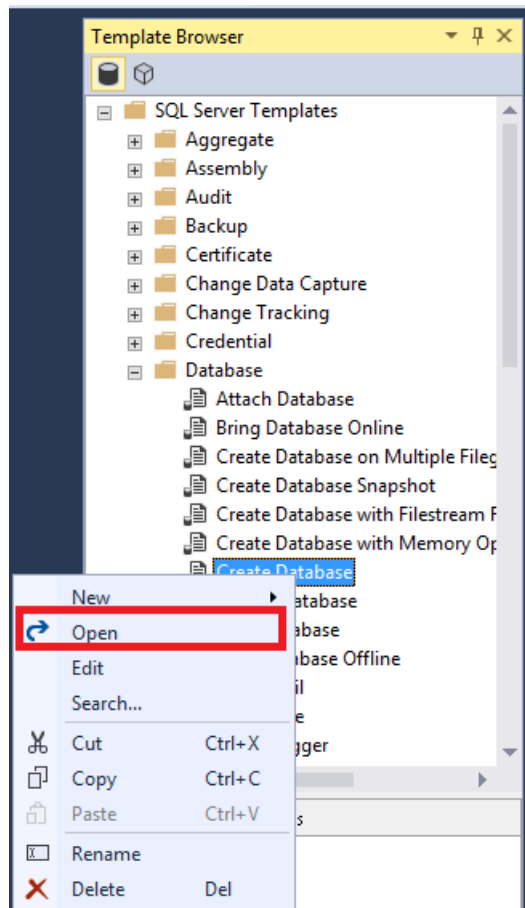
In this section, you learn how to locate and use Template Browser.

1. Open SQL Server Management Studio.
2. In the **View** menu, select **Template Browser** (Ctrl+Alt+T):



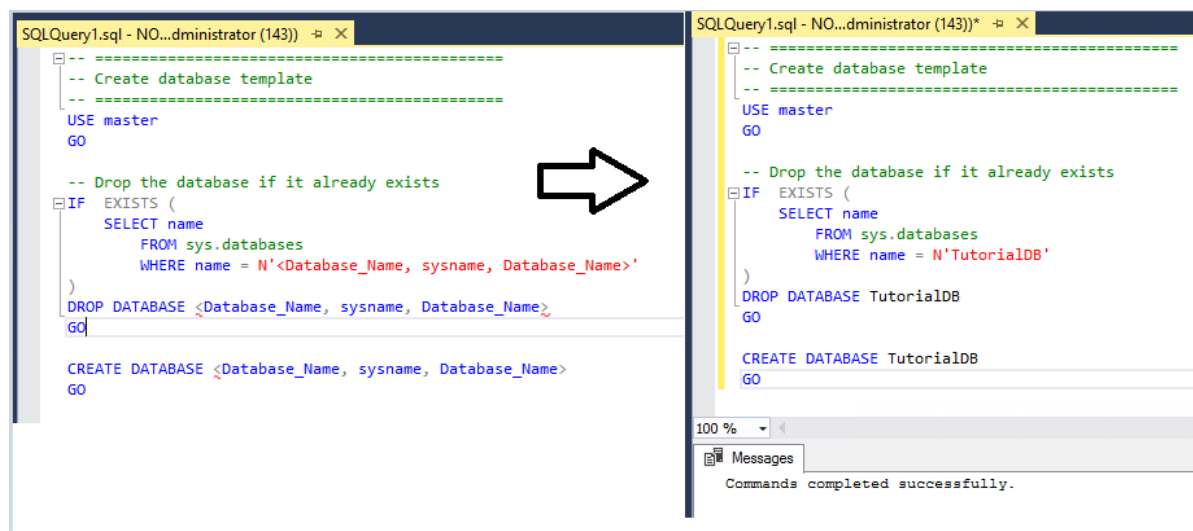
You can see recently used templates at the bottom of the template browser.

3. Expand the node you're interested in. Right-click the template, and then select **Open**:



You can also double-click the template name to open it.

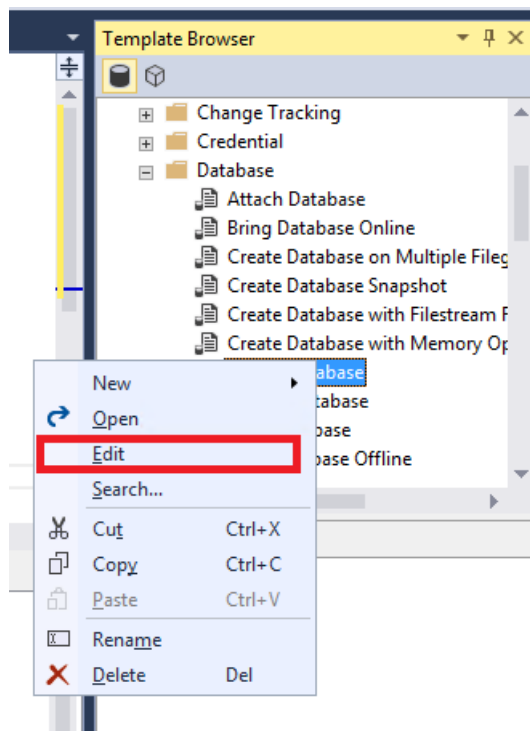
4. A new query window opens. The T-SQL script is already populated.
5. Modify the template to suit your needs, and then select **Execute** to run the query:



Edit an existing template

You can also edit existing templates in Template Browser.

1. In Template Browser, go to the template you want to work with.
2. Right-click the template, and then select **Edit**:

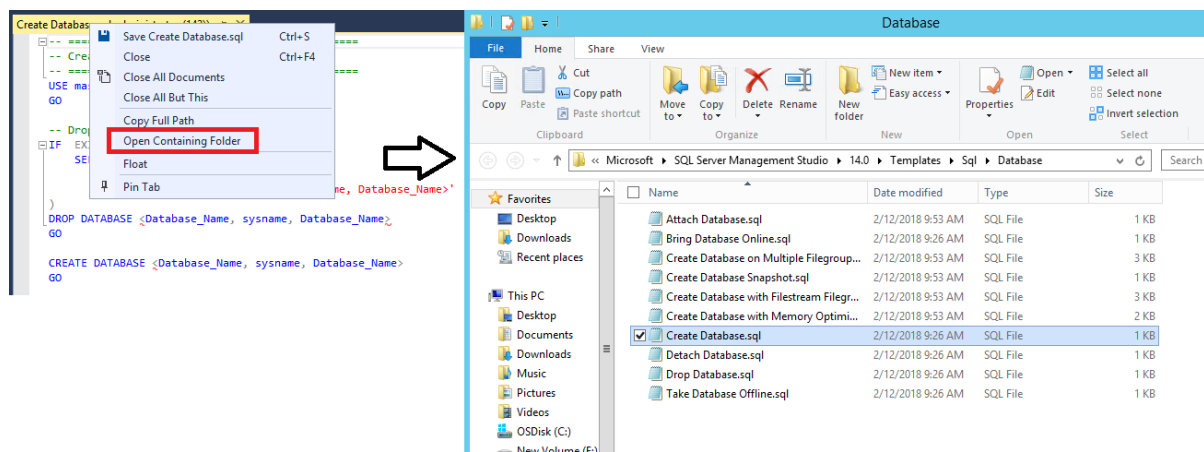


3. In the query window that opens, make the changes that you want to make.
4. To save the template, select **File > Save** (Ctrl+S).
5. Close the query window.
6. Reopen the template. Your edits should appear.

Locate templates on disk

When a template is open, you can locate the templates that are on disk.

1. In Template Browser, select a template, and then select **Edit**.
2. Right-click **Query Title**, and then select **Open Containing Folder**. The explorer should open where the templates are stored on disk:



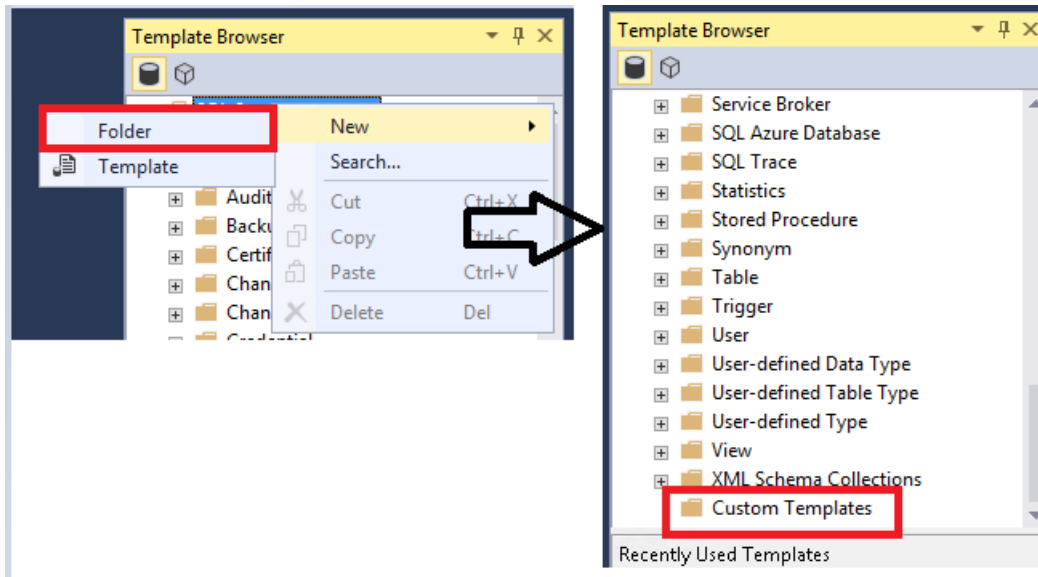
Create a new template

You can also create a new template in Template Browser. The following steps show you how to create a new folder, and then create a new template in that folder. You can also use these steps to create a custom template in an existing folder.

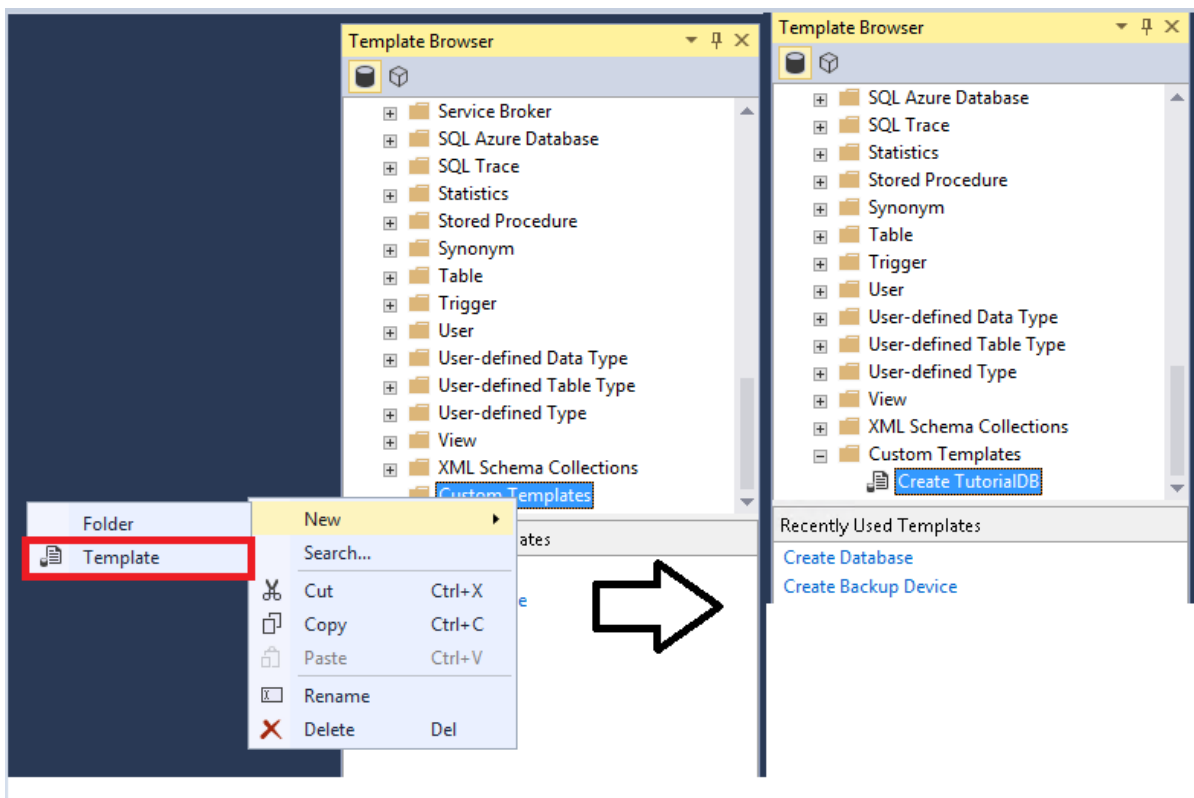
1. Open Template Browser.

2. Right-click **SQL Server Templates**, and then select **New > Folder**.

3. Name this folder **Custom Templates**:



4. Right-click the newly created Custom Templates folder, and then select **New > Template**. Enter a name for your template:



5. Right-click the template you created, and then select **Edit**. The New Query Window opens.

6. Enter the T-SQL text that you want to save.

7. In the **File** menu, select **Save**.

8. Close the existing query window, and then open your new custom template.

Next steps

The next article provides additional tips and tricks for using SQL Server Management Studio.

Tutorial: SQL Server Management Studio components and configuration

10/1/2018 • 3 minutes to read • [Edit Online](#)

This tutorial describes the various window components in SQL Server Management Studio (SSMS), and some basic configuration options for your workspace. In this article, you learn how to:

- Identify the components that make up the SSMS environment
- Change the environment layout, and reset it to the default
- Maximize the query editor
- Change the font
- Configure startup options
- Reset the configuration to the default

Prerequisites

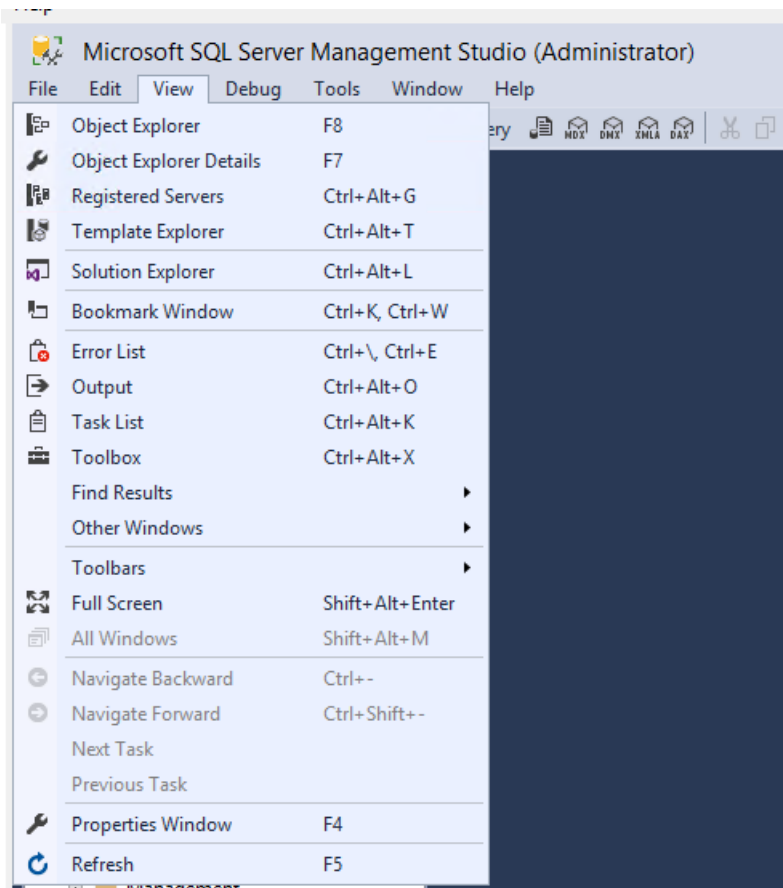
To complete this tutorial, you need SQL Server Management Studio.

- Install [SQL Server Management Studio](#).

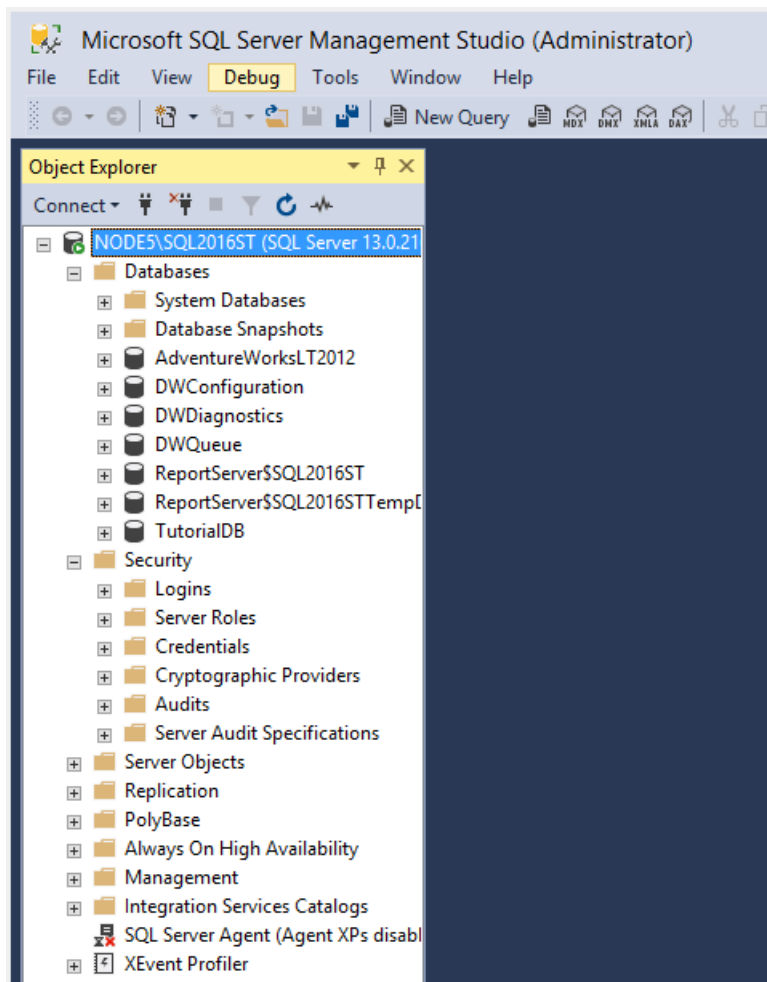
SQL Server Management Studio components

This section describes the various window components that are available in the workspace, and how to use them.

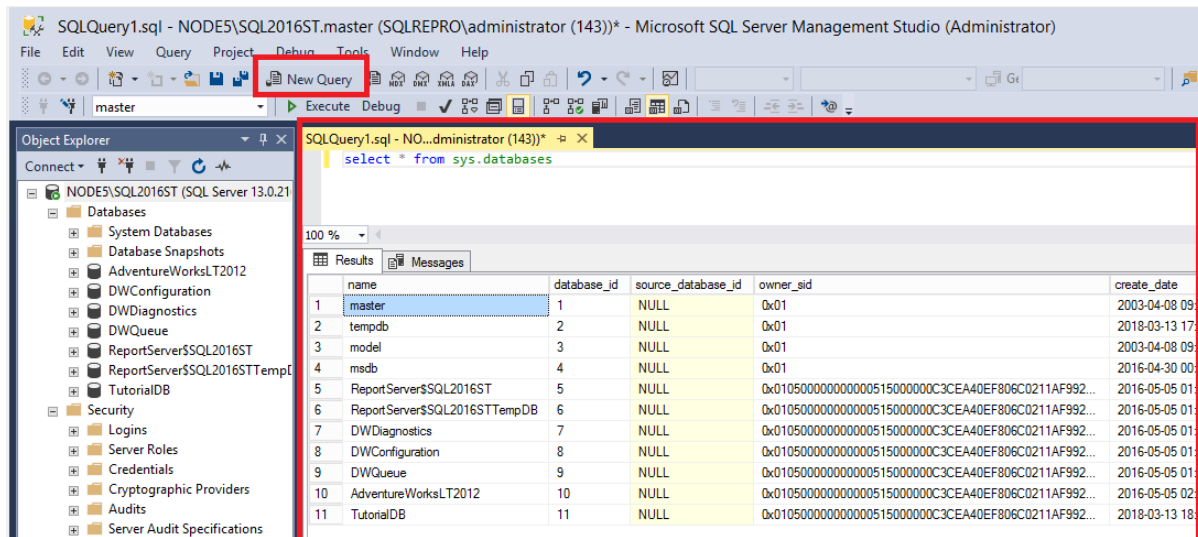
- To close a window, select the **X** in the right corner of the title bar.
- To reopen a window, select the window in the **View** menu.



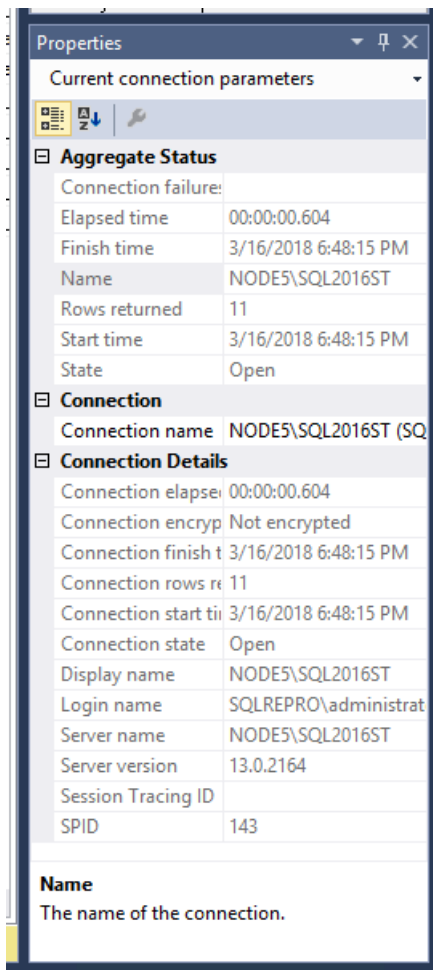
- **Object Explorer** (F8): Object Explorer is a tree view of all the database objects in a server. This view includes the databases of the SQL Server Database Engine, SQL Server Analysis Services, SQL Server Reporting Services, and SQL Server Integration Services. Object Explorer includes information for all servers that are connected to it.



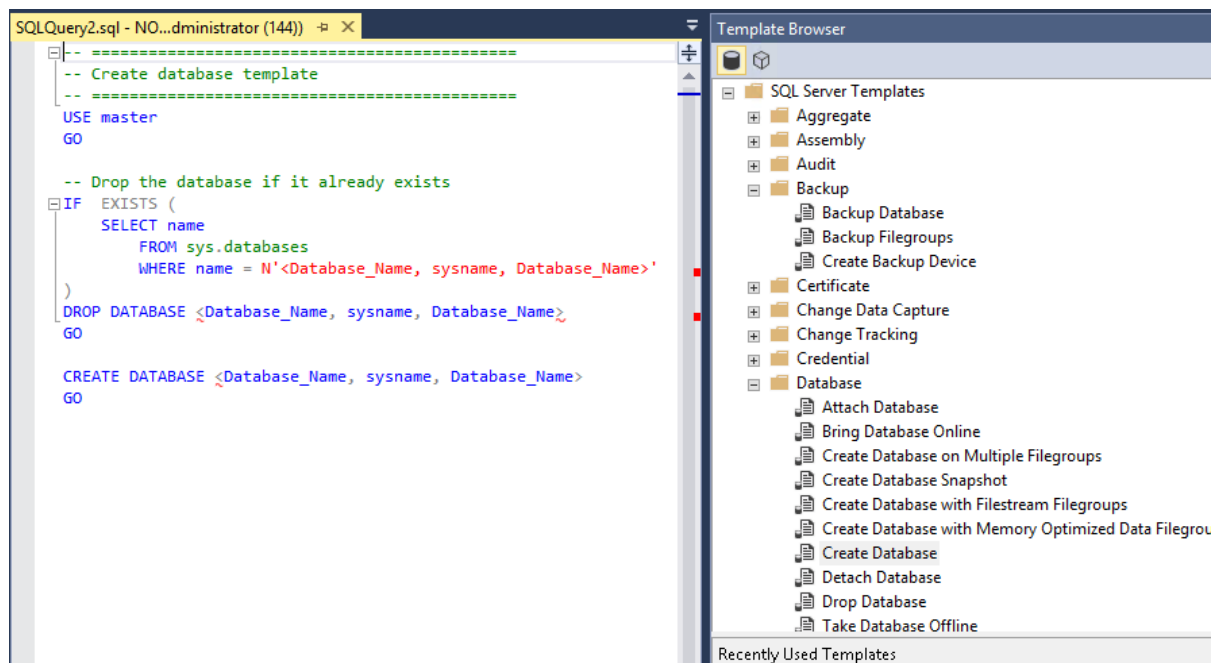
- **Query Window (Ctrl+N):** After you select **New Query**, enter your Transact-SQL (T-SQL) queries in this window. The results of your queries also appear here.



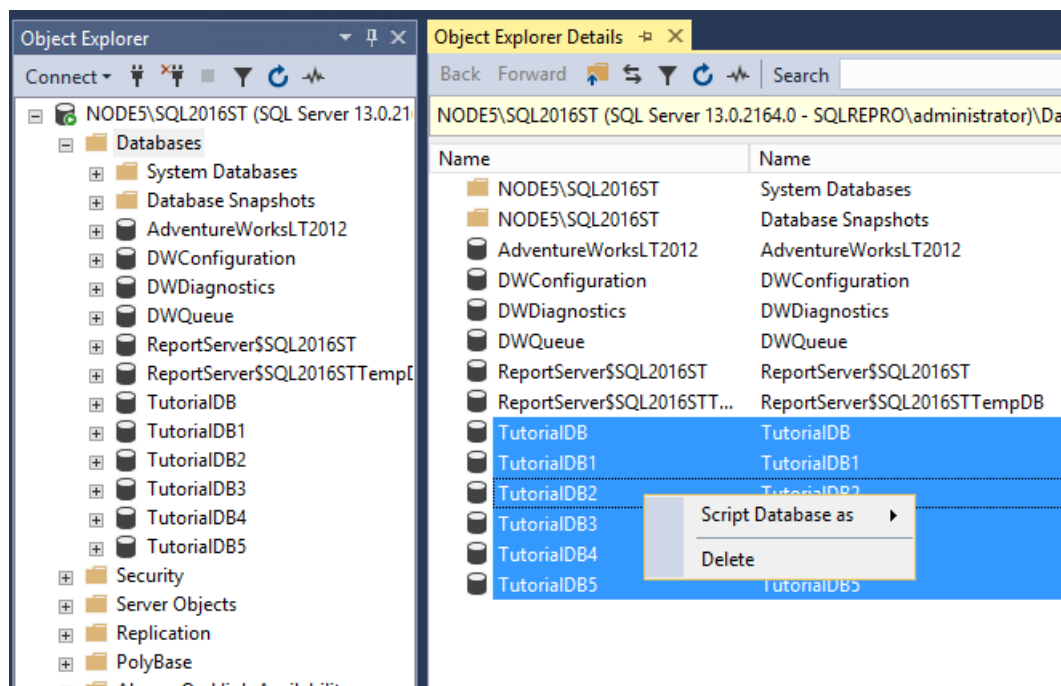
- **Properties (F4):** You can see the Properties view when the Query Window is open. The view displays basic properties of the query. For example, it shows the time that a query started, the number of rows returned, and connection details.



- **Template Browser (Ctrl+Alt+T):** Template Browser has various pre-built T-SQL templates. You can use these templates to perform various functions, such as creating or backing up a database.



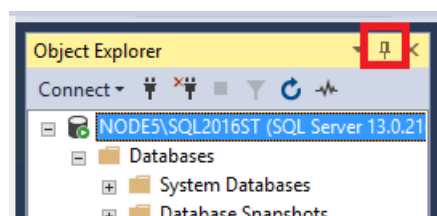
- **Object Explorer Details (F7):** This view is more granular than the view in Object Explorer. You can use Object Explorer Details to manipulate multiple objects at the same time. For example, in this window, you can select multiple databases, and then either delete them or script them out simultaneously.



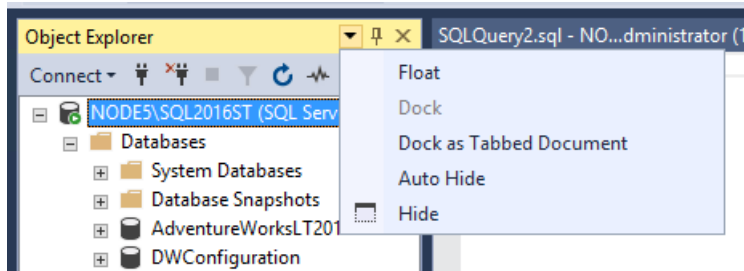
Change the environment layout

This section describes how to change the environment layout, such as how to move various windows.

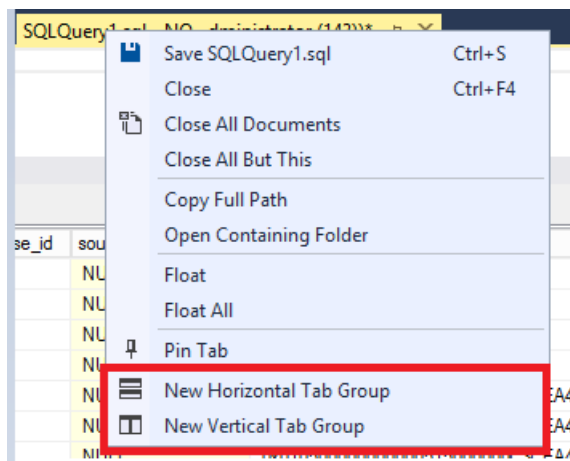
- To move a window, press and hold the title, and then drag the window.
- To pin or unpin a window, select the pushpin icon in the title bar:



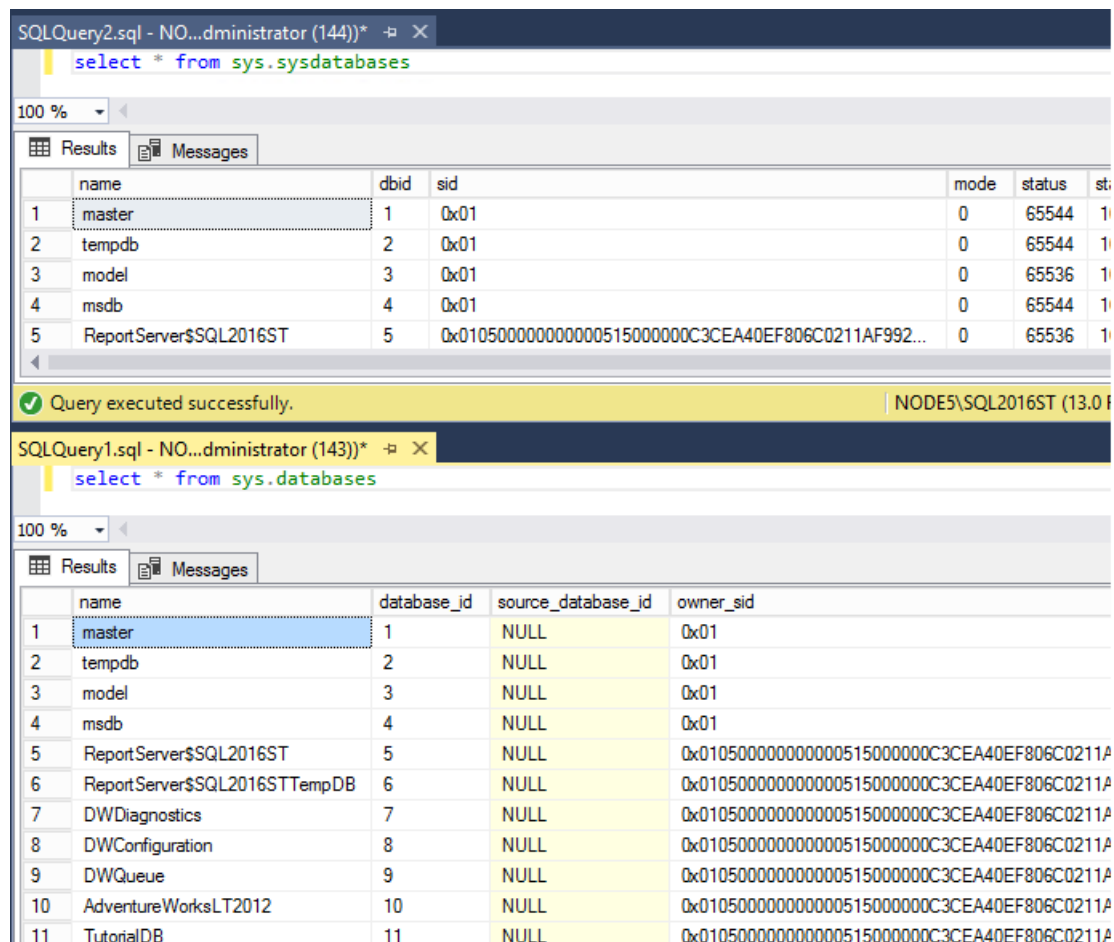
- Each window component has a drop-down menu that you can use to manipulate the window in various ways:



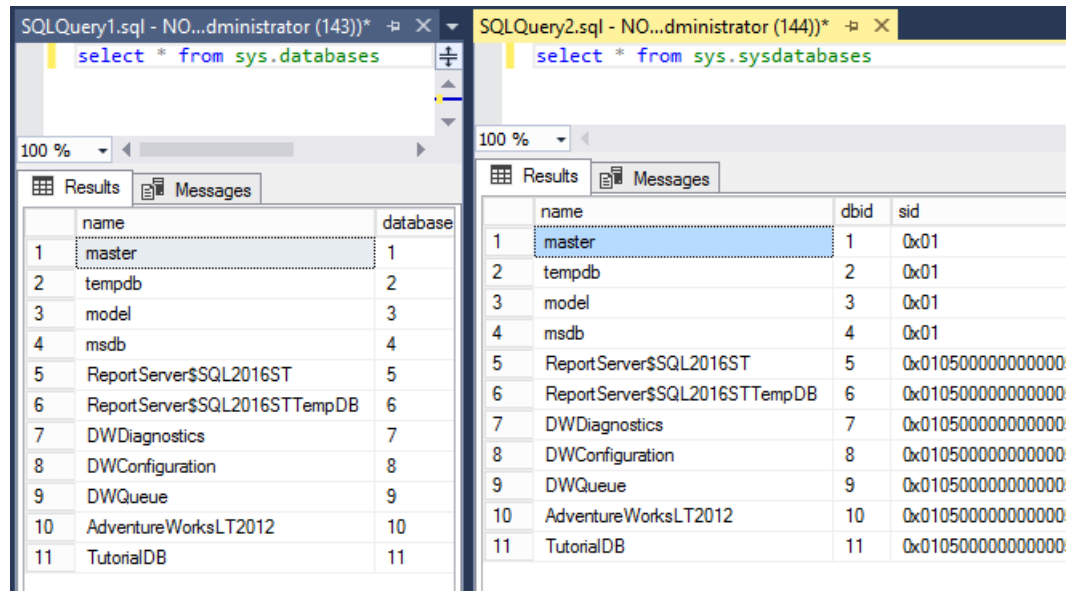
- When two or more query windows are open, the windows can be tabbed vertically or horizontally so that both query windows are visible. To view tabbed windows, right-click the title of the query, and then select the tabbed option that you want:



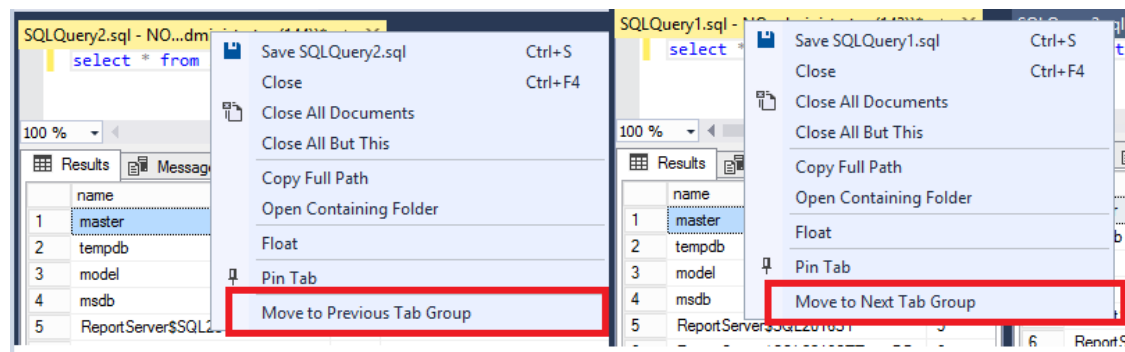
- Here's a Horizontal Tab Group:



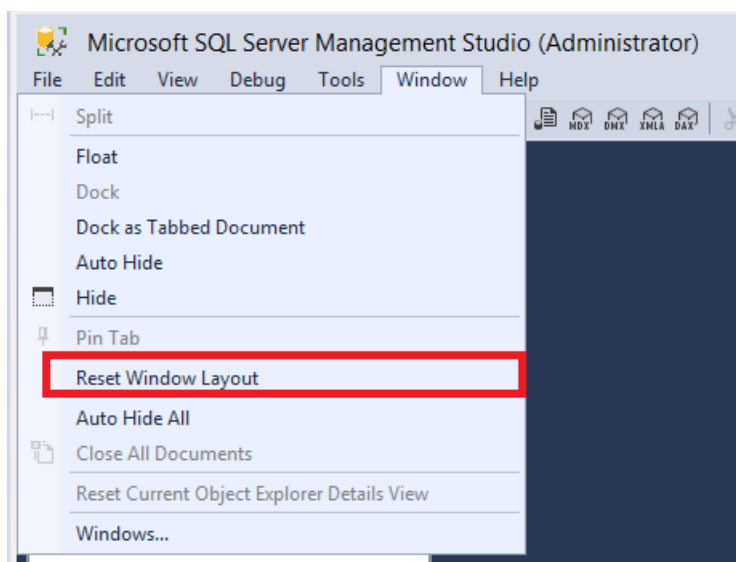
- Here's a Vertical Tab Group:



- To merge the tabs, right-click the query title, and then select **Move to Previous Tab Group** or **Move to Next Tab Group**:



- To restore the default environment layout, in the **Window** menu, select **Reset Window Layout**:



Maximize Query Editor

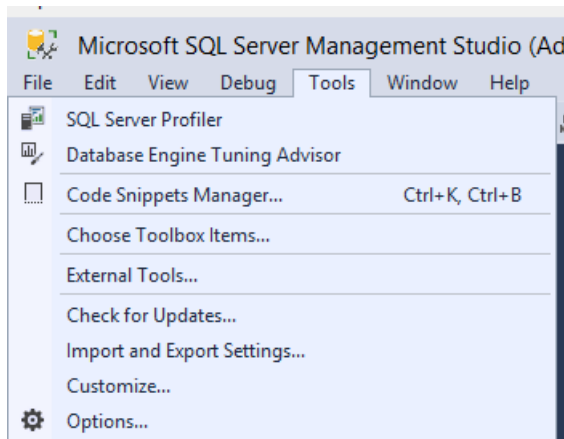
You can maximize Query Editor to full-screen mode:

1. Click anywhere in the Query Editor window.
2. Press Shift+Alt+Enter to toggle between full-screen mode and regular mode.

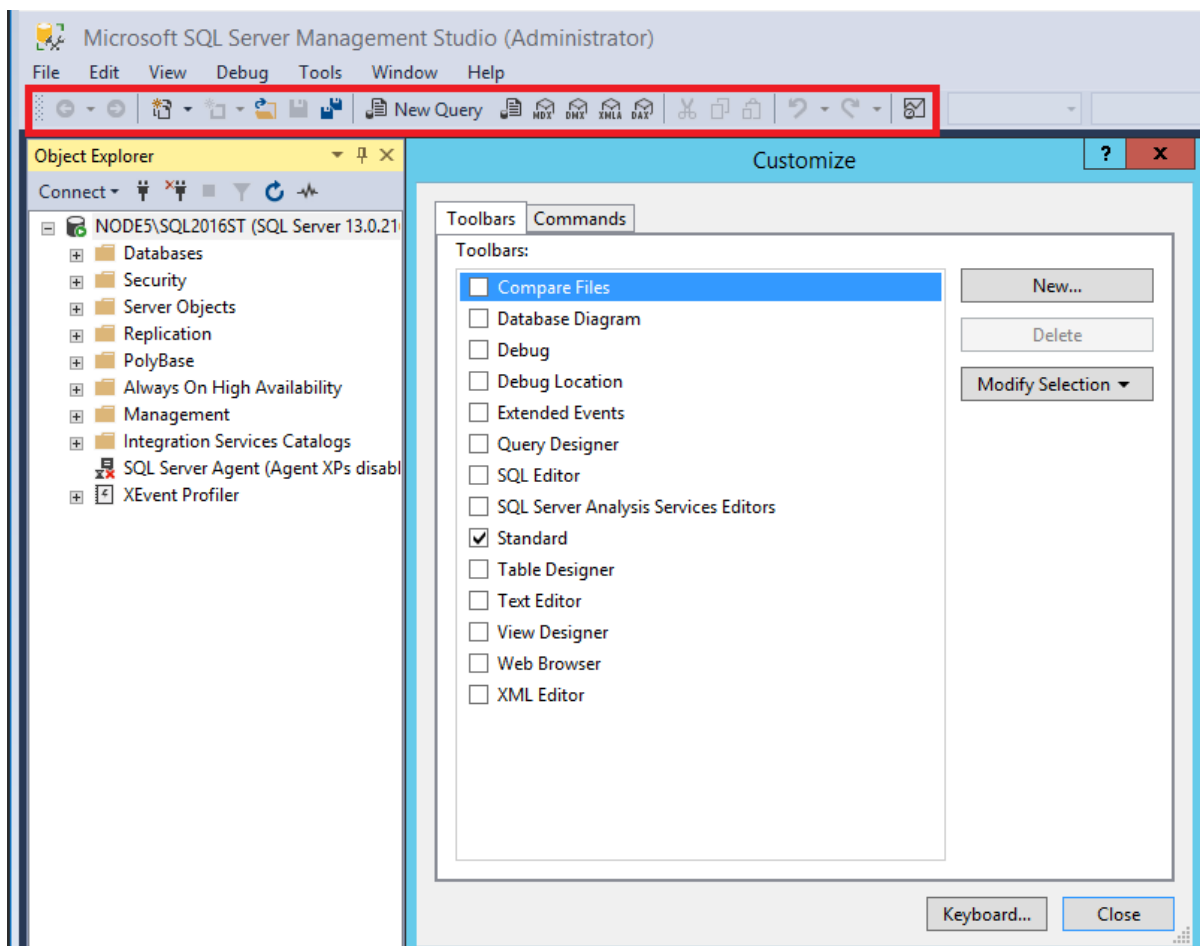
This keyboard shortcut works with any document window.

Change basic settings

This section describes how to modify some basic settings in SSMS from the **Tools** menu.

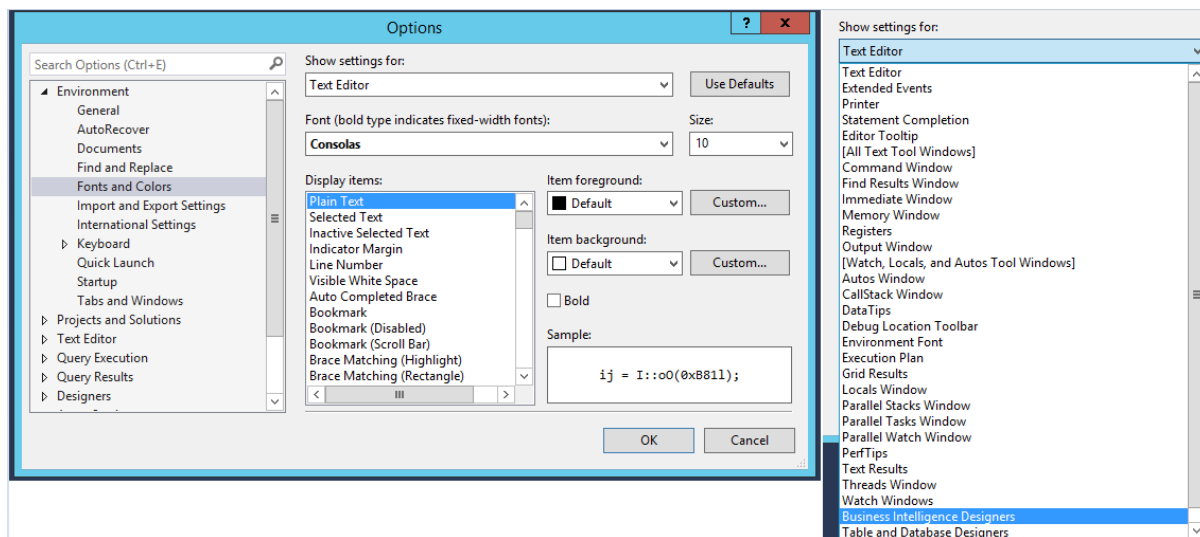


- To modify the highlighted toolbar, select **Tools > Customize**:



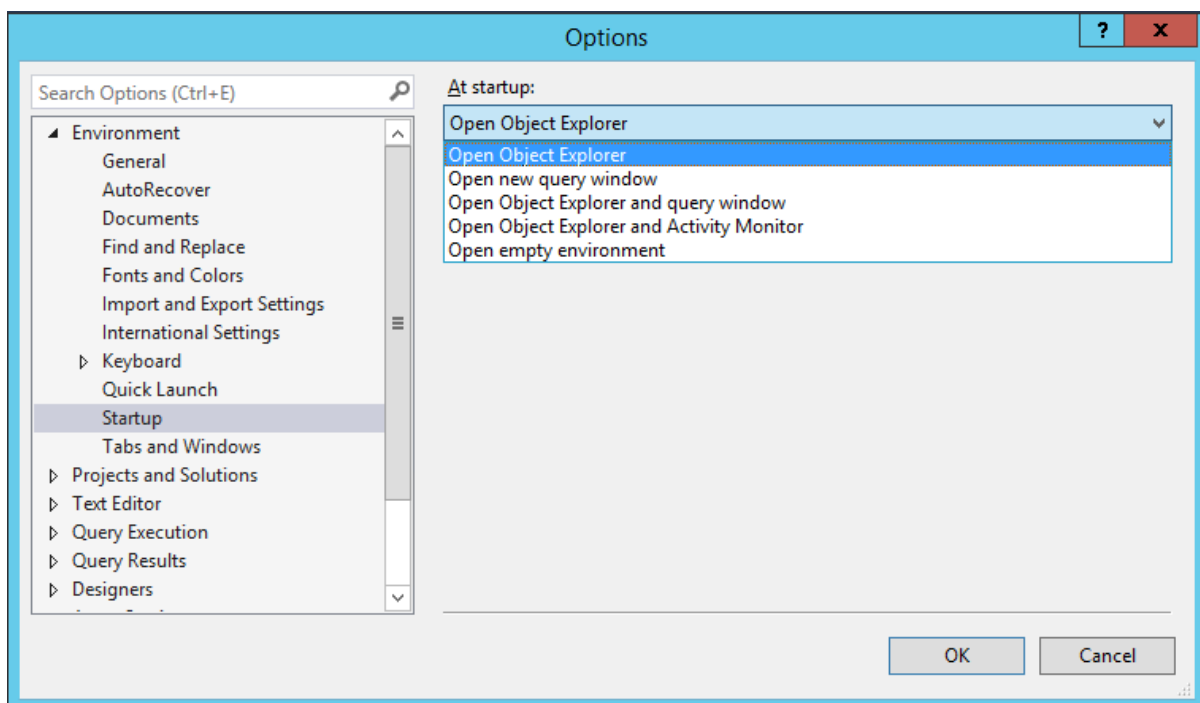
Change the font

- To change the font, select **Tools > Options > Fonts and Colors**:



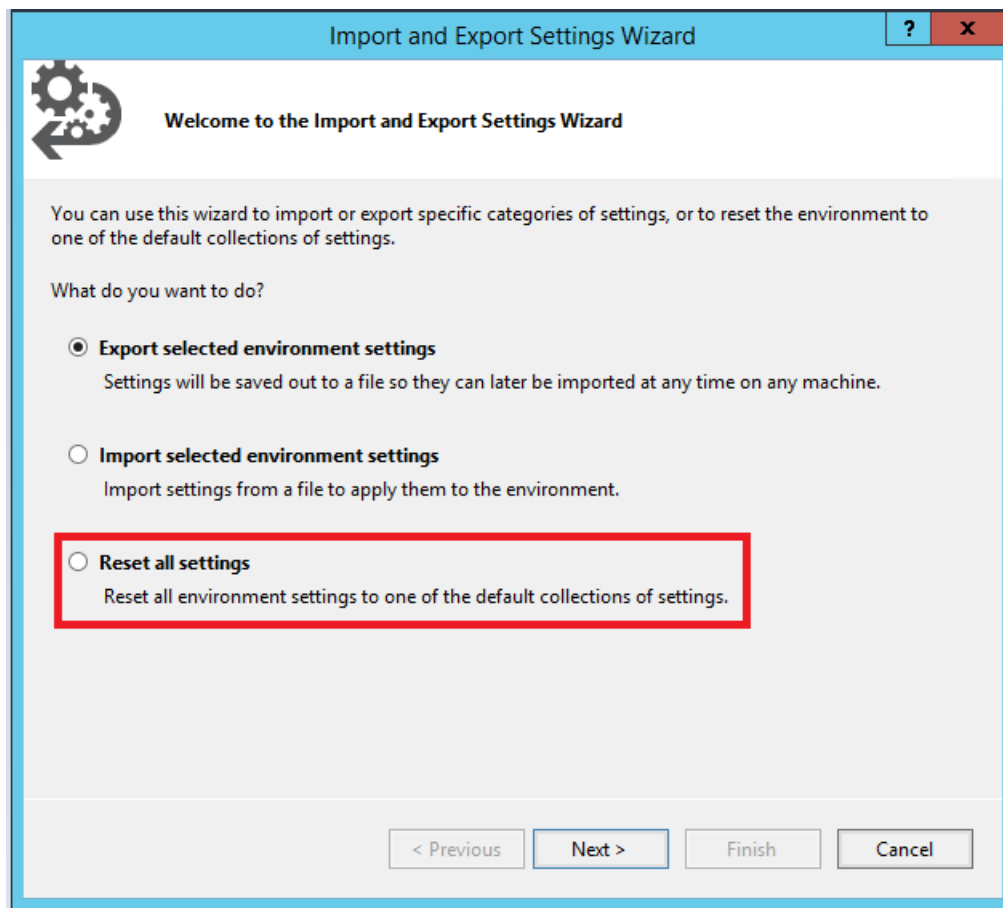
Change startup options

- Startup options determine what your workspace looks like when you first open SSMS. To change startup options, select **Tools > Options > Startup**:



Reset settings to the default

- You can export and import all these settings from the menu. To import or export settings, or to restore default settings, select **Tools > Import and Export Settings**



Next steps

The next article teaches you some additional tips and tricks for using SSMS, such as how to find your SQL Server error log and your SQL instance name.

[Additional tips and tricks for using SSMS](#)

Tutorial: Additional tips and tricks for using SSMS

10/26/2018 • 5 minutes to read • [Edit Online](#)

This tutorial gives you some additional tricks for using SQL Server Management Studio (SSMS). This article shows you how to:

- Comment/uncomment your Transact-SQL (T-SQL) text
- Indent your text
- Filter objects in Object Explorer
- Access your SQL Server error log
- Find the name of your SQL Server instance

Prerequisites

To complete this tutorial, you need SQL Server Management Studio, access to a SQL server, and an AdventureWorks database.

- Install [SQL Server Management Studio](#).
- Install [SQL Server 2017 Developer Edition](#).
- Download an [AdventureWorks sample database](#). To learn how to restore a database in SSMS, see [Restoring a database](#).

Comment/uncomment your T-SQL code

You can comment and uncomment portions of your text by using the **Comment** button on the toolbar. Text that is commented out is not executed.

1. Open SQL Server Management Studio.
2. Connect to your SQL server.
3. Open a New Query window.
4. Paste the following T-SQL code in your text window:

```
USE master
GO

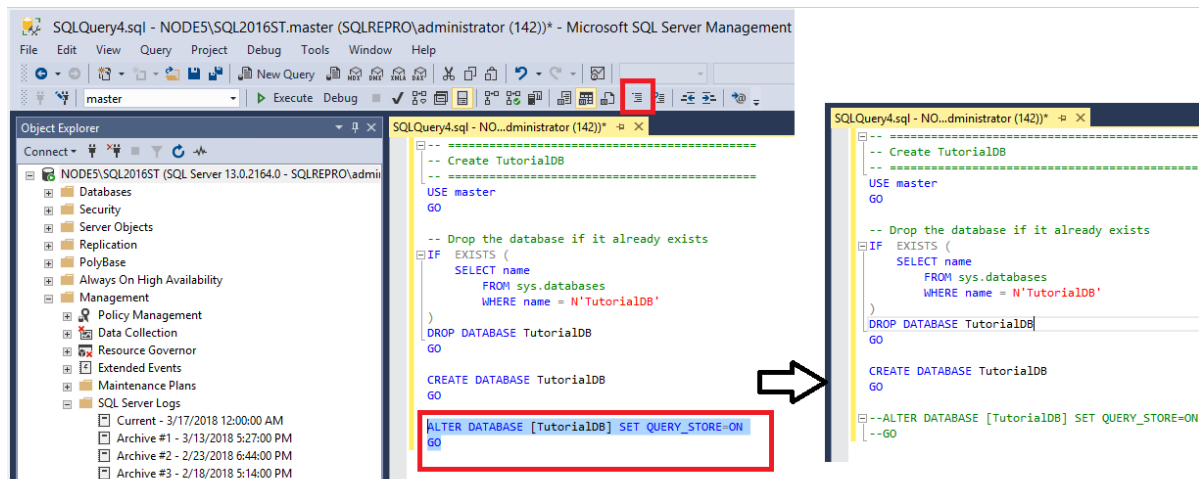
-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'TutorialDB'
)

DROP DATABASE TutorialDB
GO

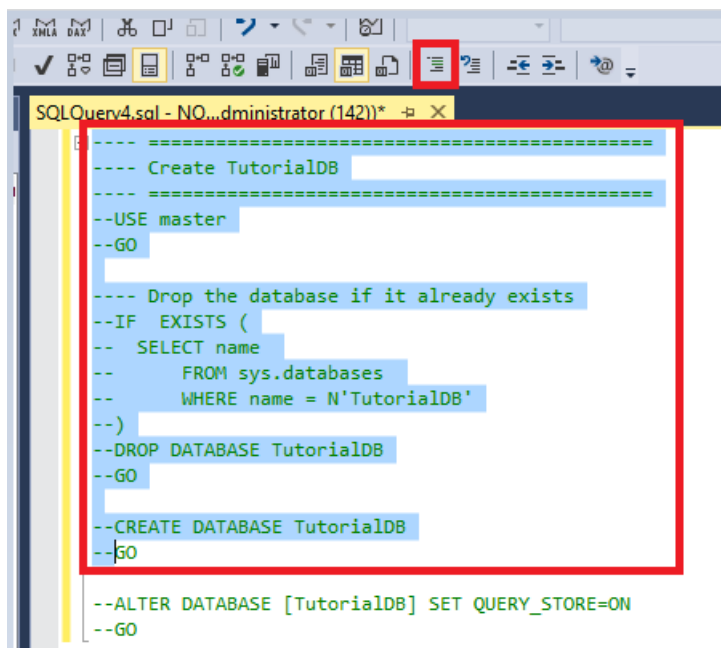
CREATE DATABASE TutorialDB
GO

ALTER DATABASE [TutorialDB] SET QUERY_STORE=ON
GO
```

5. Highlight the **Alter Database** portion of the text, and then select the **Comment** button on the toolbar:



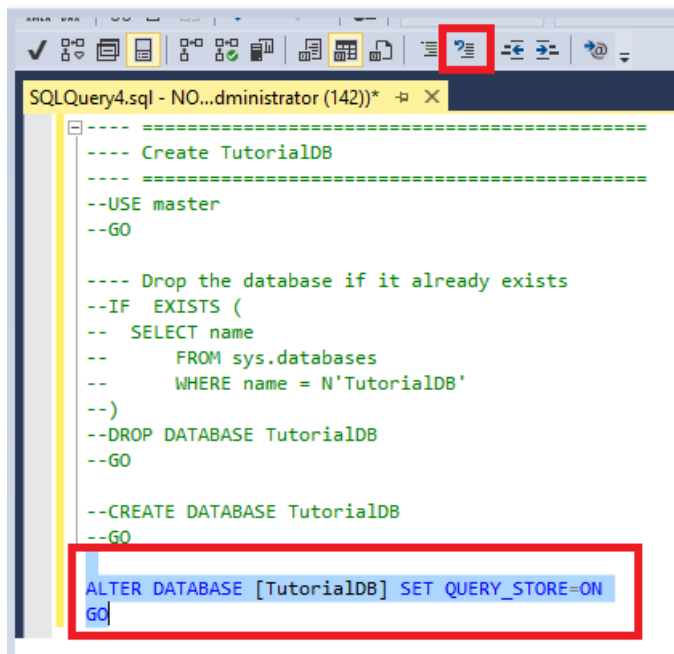
6. Select **Execute** to run the uncommented portion of the text.
7. Highlight everything except for the **Alter Database** command, and then select the **Comment** button:



NOTE

The keyboard shortcut to comment text is **CTRL + K, CTRL + C**.

8. Highlight the **Alter Database** portion of the text, and then select the **Uncomment** button to uncomment it:



NOTE

The keyboard shortcut to uncomment text is **CTRL + K, CTRL + U**.

9. Select **Execute** to run the uncommented portion of the text.

Indent your text

You can use the indentation buttons on the toolbar to increase or decrease the indent of your text.

1. Open a New Query window.
2. Paste the following T-SQL code in your text window:

```
USE master
GO

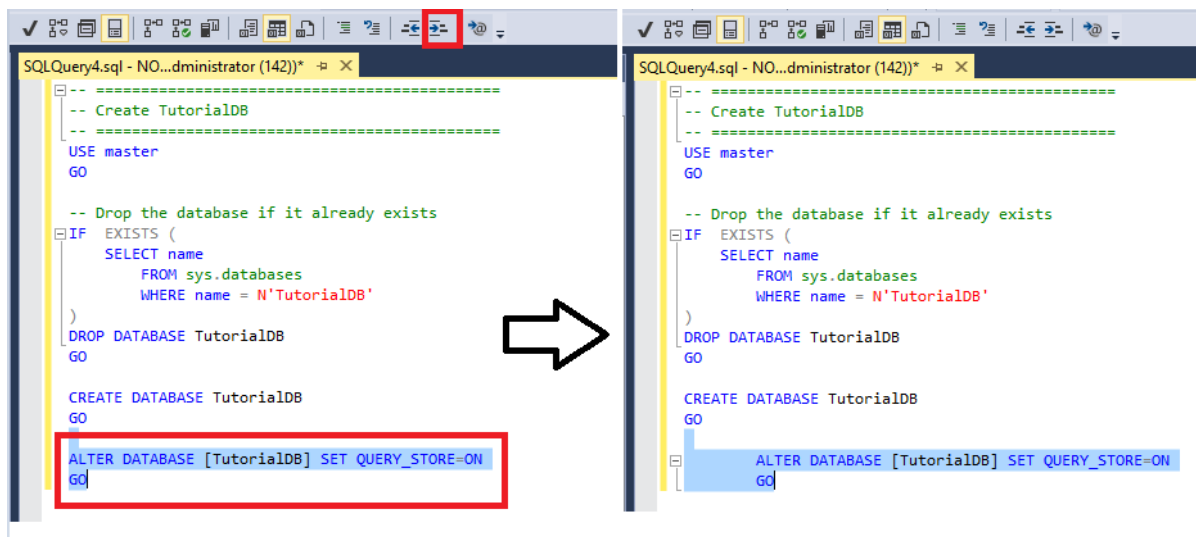
-- Drop the database if it already exists
IF EXISTS (
    SELECT name
    FROM sys.databases
    WHERE name = N'TutorialDB'
)

DROP DATABASE TutorialDB
GO

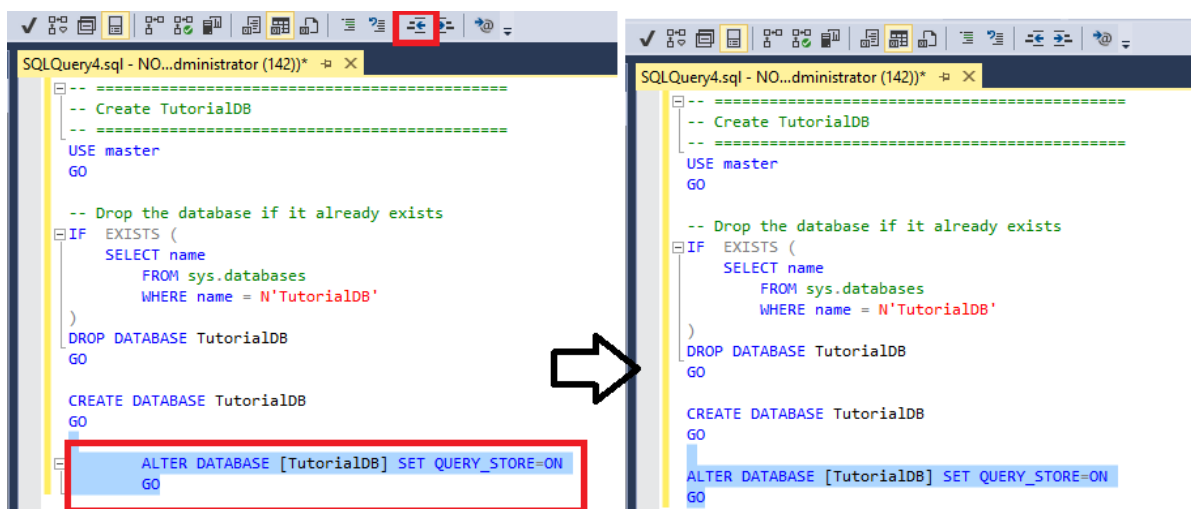
CREATE DATABASE TutorialDB
GO

ALTER DATABASE [TutorialDB] SET QUERY_STORE=ON
GO
```

3. Highlight the **Alter Database** portion of the text, and then select the **Increase Indent** button on the toolbar to move this text forward:



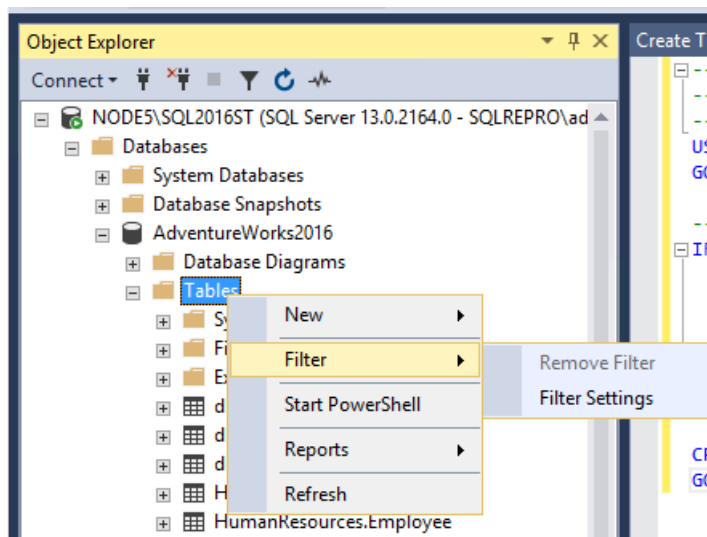
4. Highlight the **Alter Database** portion of the text again, and then select the **Decrease Indent** button to move this text back.



Filter objects in Object Explorer

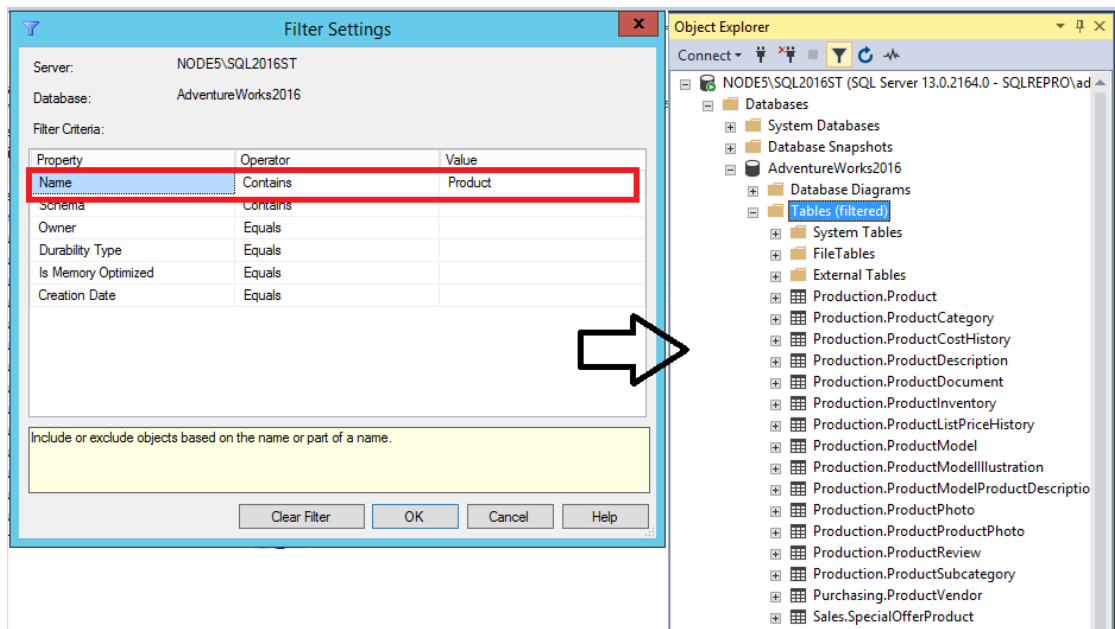
In databases that have many objects, you can use filtering to search for specific tables, views, etc. This section describes how to filter tables, but you can use the following steps in any other node in Object Explorer:

1. Connect to your SQL server.
2. Expand **Databases** > **AdventureWorks** > **Tables**. All the tables in the database appear.
3. Right-click **Tables**, and then select **Filter** > **Filter Settings**:

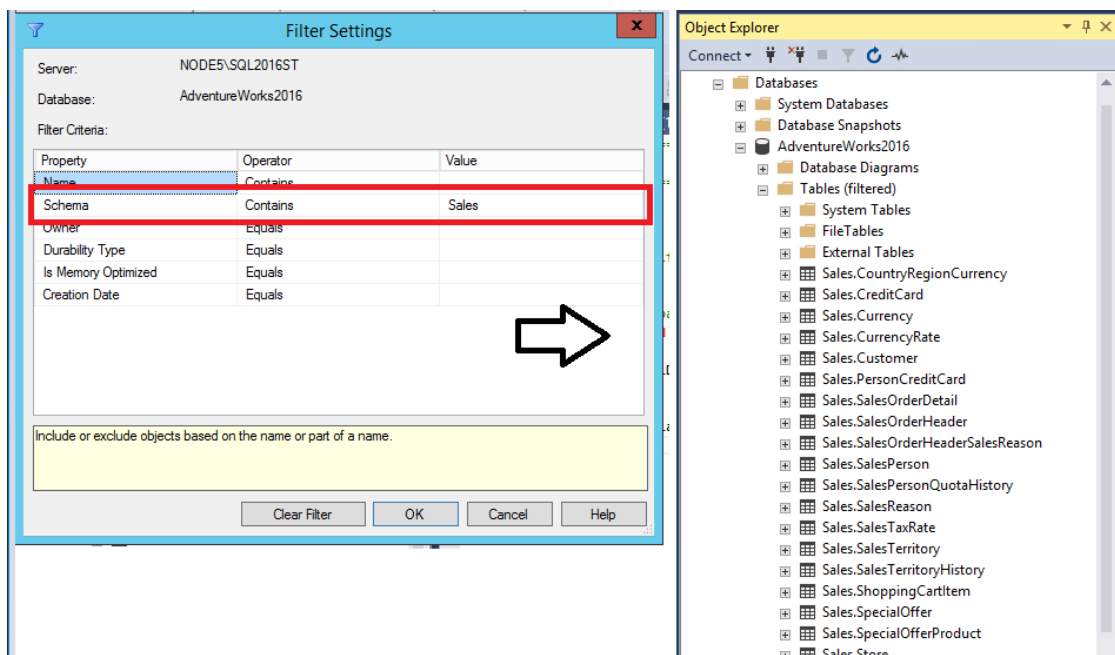


4. In the **Filter Settings** window, you can modify some of the following filter settings:

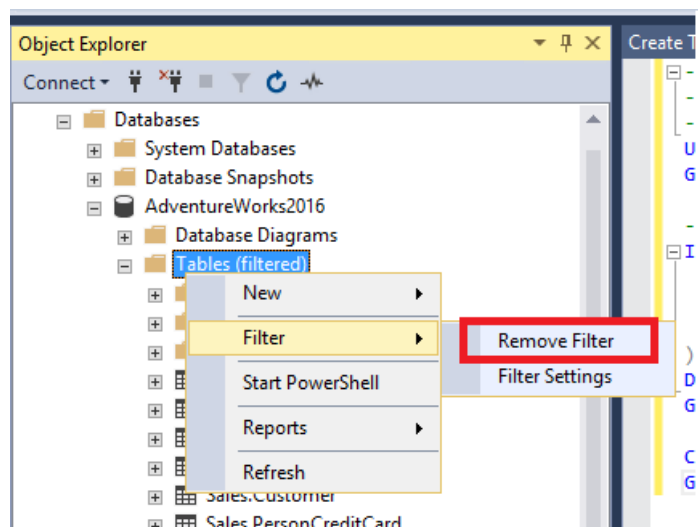
- Filter by name:



- Filter by schema:



5. To clear the filter, right-click **Tables**, and then select **Remove Filter**.

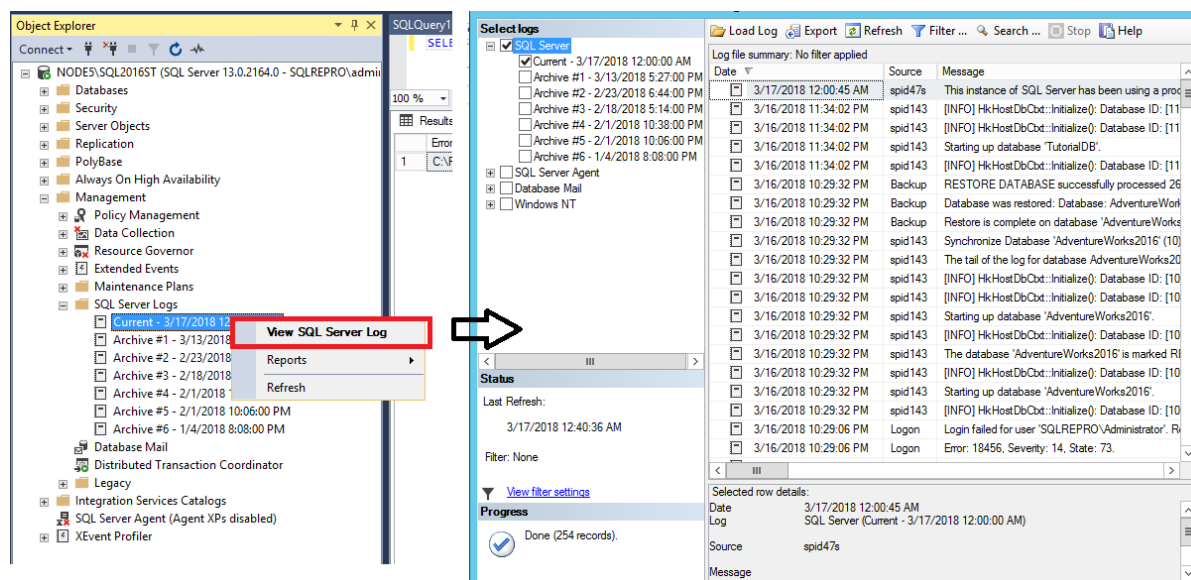


Access your SQL Server error log

The error log is a file that contains details about things that occur in your SQL Server instance. You can browse and query the error log in SSMS. The error log is a .log file that's located on your disk.

Open the error log in SSMS

1. Connect to your SQL server.
2. Expand **Management > SQL Server Logs**.
3. Right-click the **Current** error log, and then select **View SQL Server Log**:

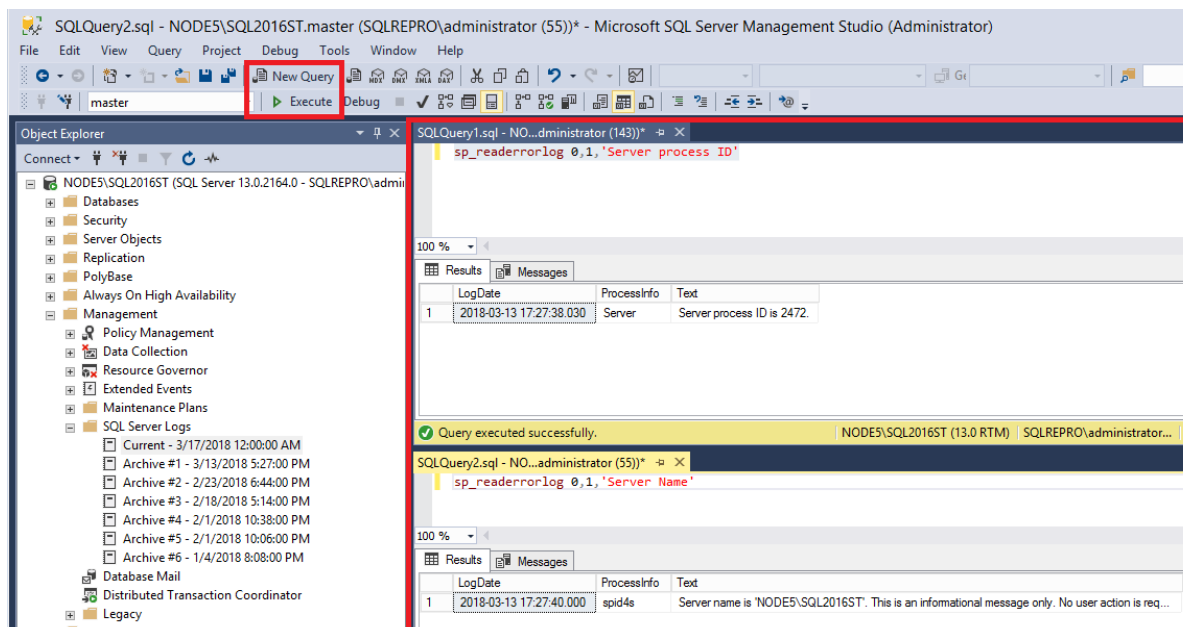


Query the error log in SSMS

1. Connect to your SQL server.
2. Open a New Query window.
3. Paste the following T-SQL code in your query window:

```
sp_readerrorlog 0,1,'Server process ID'
```

4. Modify the text in the single quotes to text you want to search for.
5. Execute the query, and then review the results:

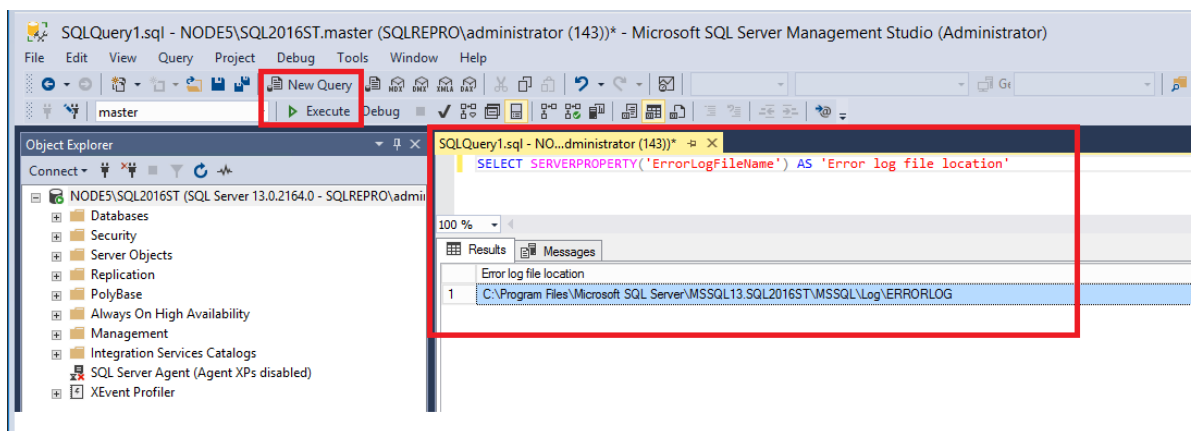


Find the error log location if you're connected to SQL Server

1. Connect to your SQL server.
2. Open a New Query window.
3. Paste the following T-SQL code in your query window, and then select **Execute**:

```
SELECT SERVERPROPERTY('ErrorLogFileName') AS 'Error log file location'
```

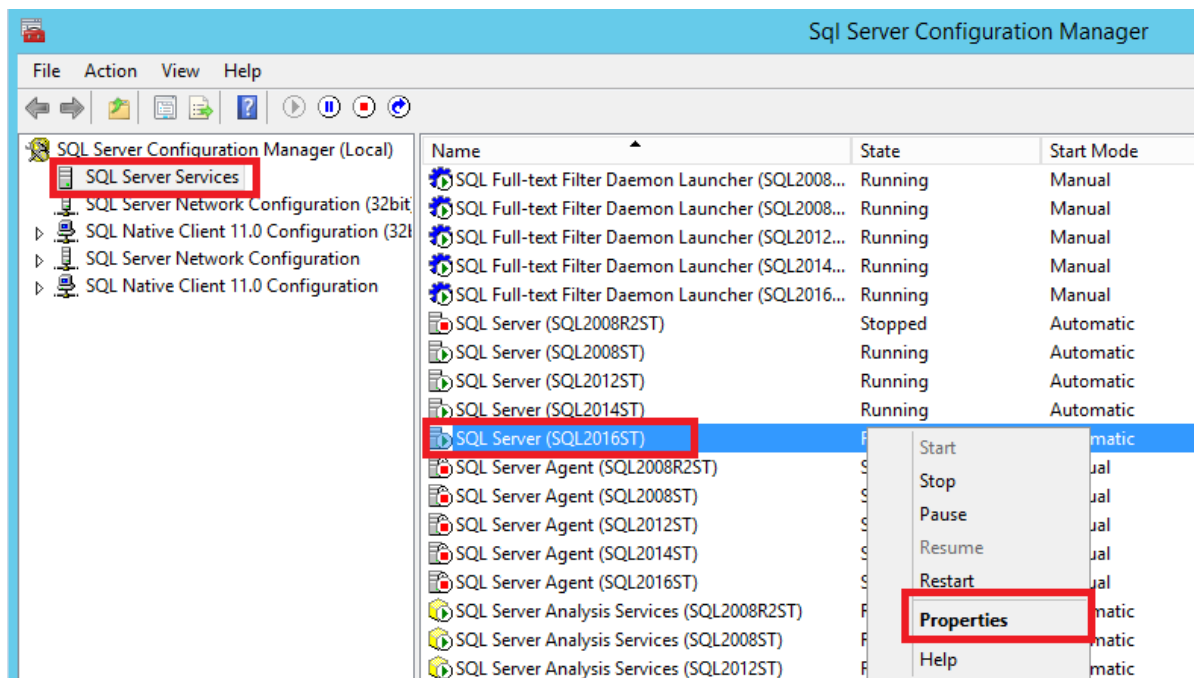
4. The results show the location of the error log in the file system:



Find the error log location if you can't connect to SQL Server

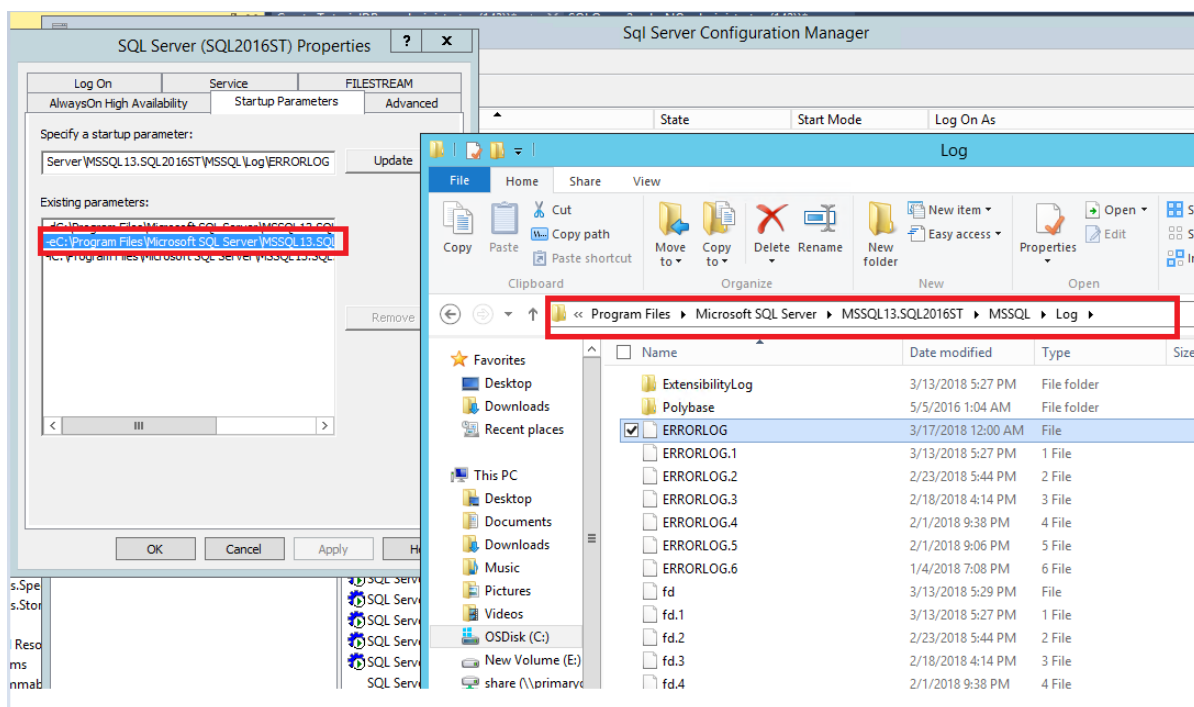
The path for your SQL Server error log can vary depending on your configuration settings. The path for the error log location can be found in the startup parameters within the SQL Server Configuration Manager. Follow the steps below to locate the relevant startup parameter identifying the location of your SQL Server error log. *Your path may vary from the path indicated below.*

1. Open SQL Server Configuration Manager.
2. Expand **Services**.
3. Right-click your SQL Server instance, and then select **Properties**:



4. Select the **Startup Parameters** tab.

5. In the **Existing Parameters** area, the path after "-e" is the location of the error log:



There are several errorlog.* files in this location. The file name that ends with *.log is the current error log file. File names that end with numbers are previous log files. A new log is created every time the SQL server restarts.

6. Open the errorlog.log file in Notepad.

Find SQL Server instance name

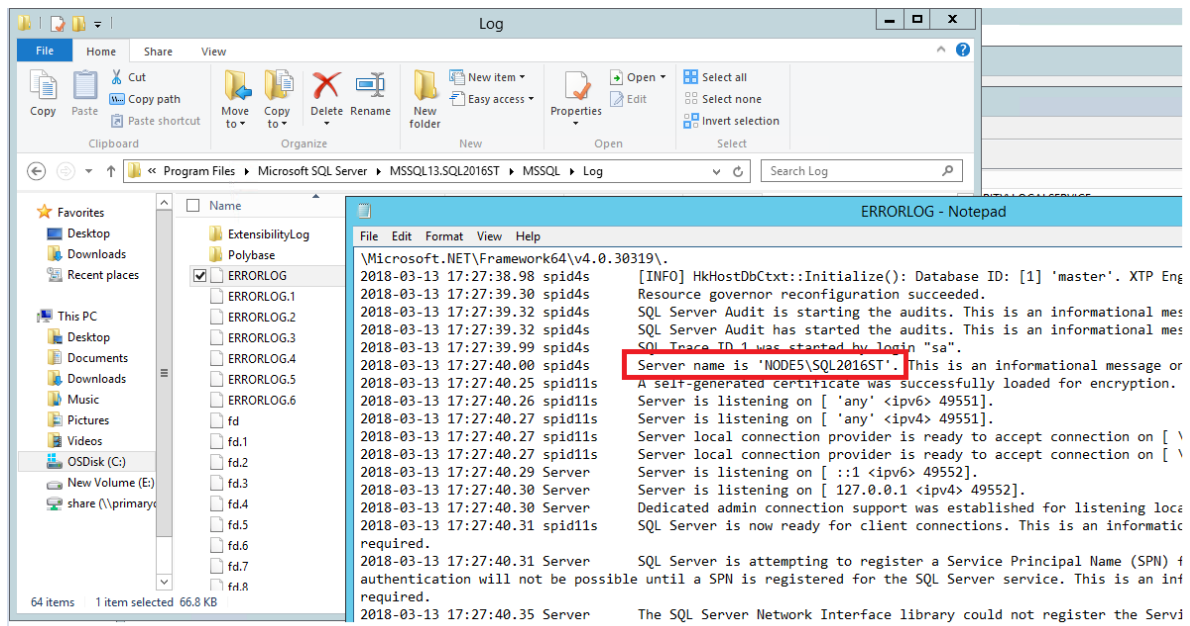
You have a few options for finding the name of your SQL server before and after you connect to SQL Server.

Before you connect to SQL Server

1. Follow the steps to locate the [SQL Server error log on disk](#). Your path may vary from the path in the image below.

2. Open the errorlog.log file in Notepad.
3. Search for the text *Server name is*.

Whatever is listed in the single quotes is the name of the SQL Server instance that you'll be connecting to:

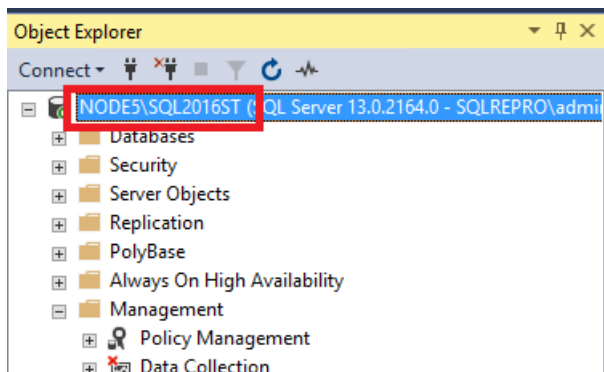


The format of the name is HOSTNAME\INSTANCENAME. If you see only the host name, then you've installed the default instance and your instance name is MSSQLSERVER. When you connect to a default instance, the host name is all you need to enter to connect to your SQL server.

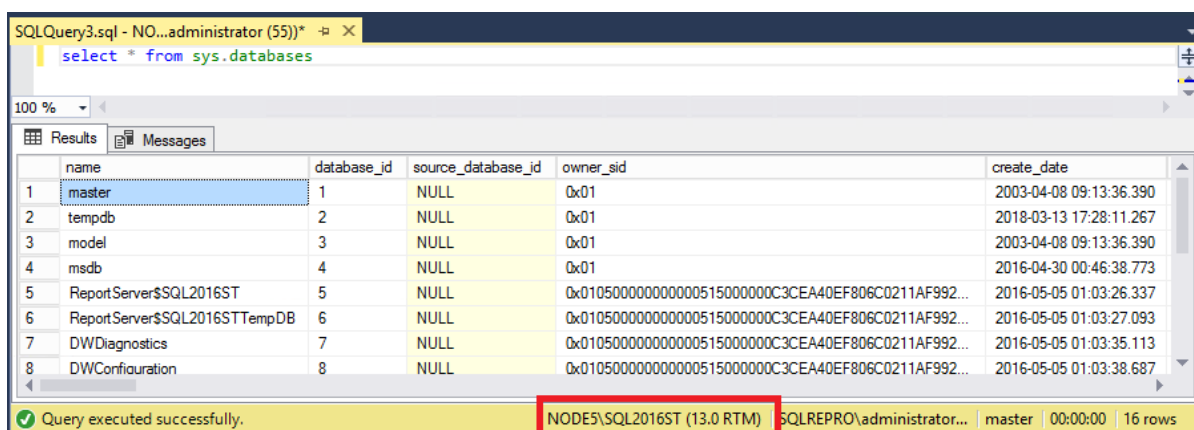
When you're connected to SQL Server

When you're connected to SQL Server, you can find the server name in three locations:

1. The name of the server is listed in Object Explorer:

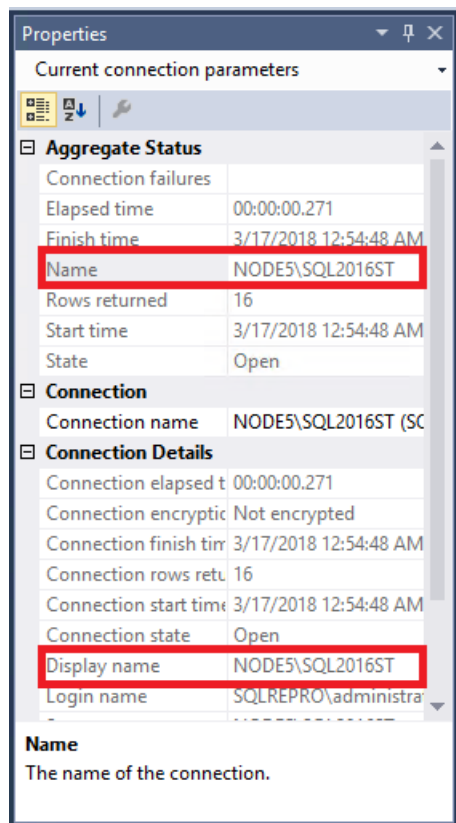


2. The name of the server is listed in the Query window:



3. The name of the server is listed in **Properties**.

- In the **View** menu, select **Properties Window**:



If you're connected to an alias or Availability Group listener

If you're connected to an alias or to an Availability Group listener, that information appears in Object Explorer and Properties. In this case, the SQL Server name might not be readily apparent, and must be queried:

1. Connect to your SQL server.
2. Open a New Query window.
3. Paste the following T-SQL code in the window:

```
select @@Servername
```

4. View the results of the query to identify the name of the SQL Server instance you're connected to:

