**=> What is normalization ?**

**Defination :** Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process:

* eliminating redundant data (for example, storing the same data in more than one table) -and ensuring data dependencies make sense (only storing related data in a table).

Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored. There are several benefits for using Normalization in Database.

**Benefits :**

1. Eliminate data redundancy
2. Improve performance
3. Query optimization
4. Faster update due to less number of columns in one table
5. Index improvement

There are diff. - diff. types of Normalizations form available in the Database. Lets see one by one.

**1. First Normal Form (1NF)**

 First normal form (1NF) sets the very basic rules for an organized database:

* Eliminate duplicative columns from the same table.
* Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).
  1. Remove repetative groups
  2. Create Primary Key

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Name** | **State** | **Country** | **Phone1** | **Phone2** | **Phone3** |
| John | 101 | 1 | 488-511-3258 | 781-896-9897 | 425-983-9812 |
| Bob | 102 | 1 | 861-856-6987 |  |  |
| Rob | 201 | 2 | 587-963-8425 | 425-698-9684 |  |
| **PK** |  |  | **[ Phone Nos ]** | | |
| ? |  |  |  | ? |  |
| **ID** | **Name** | **State** | **Country** | **Phone** |  |
| 1 | John | 101 | 1 | 488-511-3258 |  |
| 2 | John | 101 | 1 | 781-896-9897 |  |
| 3 | John | 101 | 1 | 425-983-9812 |  |
| 4 | Bob | 102 | 1 | 861-856-6987 |  |
| 5 | Rob | 201 | 2 | 587-963-8425 |  |
| 6 | Rob | 201 | 2 | 425-698-9684 |  |
|  |  |  |  |  |  |

2. **Second Normal Form (2NF)**Second normal form (2NF) further addresses the concept of removing duplicative data:

·         Meet all the requirements of the first normal form.

·         Remove subsets of data that apply to multiple rows of a table and place them in separate tables.

·         Create relationships between these new tables and their predecessors through the use of foreign keys.

Remove columns which create duplicate data in a table and related a new table with Primary Key – Foreign Key relationship

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | State | Country | Phone |  |  |  |
| 1 | John | 101 | 1 | 488-511-3258 |  |  |  |
| 2 | John | 101 | 1 | 781-896-9897 |  |  |  |
| 3 | John | 101 | 1 | 425-983-9812 |  |  |  |
| 4 | Bob | 102 | 1 | 861-856-6987 |  |  |  |
| 5 | Rob | 201 | 2 | 587-963-8425 |  |  |  |
| 6 | Rob | 201 | 2 | 425-698-9684 |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| ID | Name | State | Country |  | PhoneID | ID | Phone |
| 1 | John | 101 |  |  | 1 | 1 | 488-511-3258 |
| 2 | Bob | 102 |  | 2 | 1 | 781-896-9897 |
| 3 | Rob | 201 |  | 3 | 1 | 425-983-9812 |
|  |  |  |  |  | 4 | 2 | 587-963-8425 |
|  |  |  |  |  | 5 | 3 | 587-963-8425 |
|  |  |  |  |  | 6 | 3 | 425-698-9684 |

3. **Third Normal Form (3NF)**

Third normal form (3NF) goes one large step further:

·         Meet all the requirements of the second normal form.

·         Remove columns that are not dependent upon the primary key.

  Country can be derived from State also… so removing country

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | State | Country |
| 1 | John | 101 | 1 |
| 2 | Bob | 102 | 1 |
| 3 | Rob | 201 | 2 |

After 3NF

|  |  |  |
| --- | --- | --- |
| ID | Name | State |
| 1 | John | 101 |
| 2 | Bob | 102 |
| 3 | Rob | 201 |

**4. Fourth Normal Form (4NF)**

Finally, fourth normal form (4NF) has one additional requirement:

·         Meet all the requirements of the third normal form.

·         A relation is in 4NF if it has no multi-valued dependencies.

If PK is composed of multiple columns then all non-key attributes should be derived from FULL PK only. If some non-key attribute can be derived from partial PK then remove it

The 4NF also known as BCNF NF

|  |  |  |  |
| --- | --- | --- | --- |
| TeacherID | StudentID | SubjectID | StudentName |
| 101 | 1001 | 1 | John |
| 101 | 1002 | 2 | Rob |
| 201 | 1002 | 3 | Bob |
| 201 | 1001 | 2 | Rob |
|  |  |  |  |
|  |  |  |  |
| TeacherID | StudentID | SubjectID | StudentName |
| 101 | 1001 | 1 | X |
| 101 | 1002 | 2 | X |
| 201 | 1001 | 3 | X |
| 201 | 1002 | 2 | X |

Concept and basics of DBCC Commands in SQL Serveri

**Basic Commands:**

These are a good place to begin if you have not worked with DBCC commands previously starting from the upcoming ones:

**DBCC HELP:**

* Provides syntax for a specific DBCC command, or lists all commands.
* By default, only supported commands listed
* This command has no impact on performance or data
* Requires sysadmin role

We have two options to run this DBCC Command:

When you just run it as it is, it will list all of the supported DBCC commands



|  |  |
| --- | --- |
| 1 | DBCC HELP ('?'); |

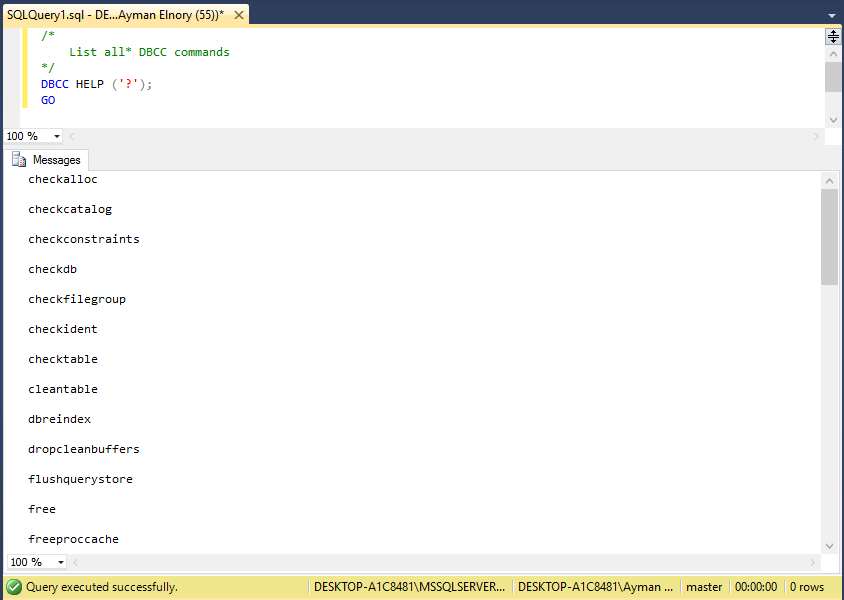
<img style="margin: 0px auto; display: block;" src="/wp-content/uploads/2017/03/word-image-264.png" />

Figure 1 DBCC HELP command

When you run it, providing a specific command, it will give you the syntax for that:



|  |  |
| --- | --- |
| 1  2 | DBCC HELP (CHECKDB);  GO |

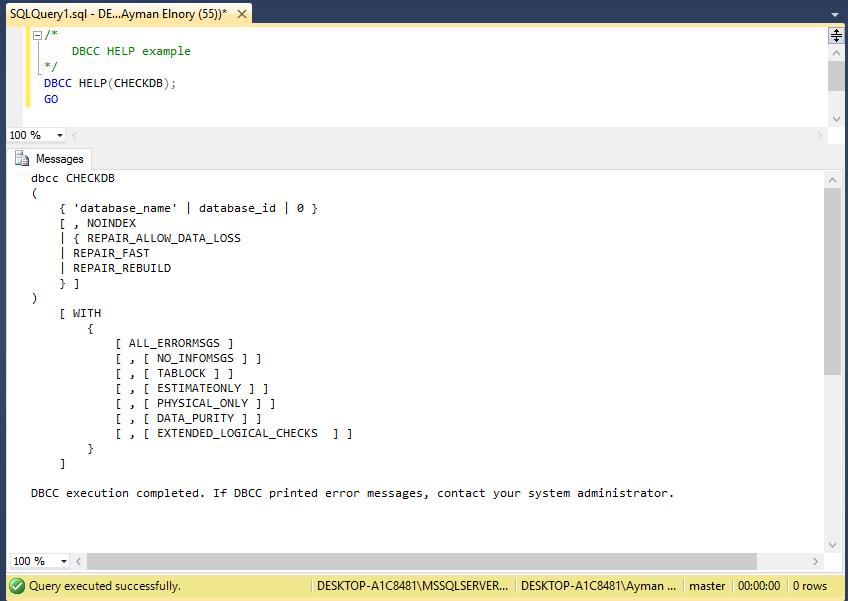
<img style="margin: 0px auto; display: block;" src="/wp-content/uploads/2017/03/word-image-265.png" />

Figure 2 DBCC HELP (CHECKDB) command

It will be very helpful if you don’t have internet access to get easy the syntax of any DBCC command just execute DBCC Help.

**DBCC TRACEON, DBCC TRACEOFF, and DBCC TRACESTATUS:**

If you have worked with trace flags before, you probably know that you can apply them as a startup parameter for the SQL Server service. Well, that is all well and good until you realize that for them to take effect, you have to restart the service. That is not always something you can easily do so that I can list some of the useful features of these commands:

* DBCC TRACEON and DBCC TRACEOFF can be used to enable/disable a trace flag without requiring a service restart.
* With DBCC TRACEON and DBCC TRACEOFF, you can enable/disable trace flags at either the session or the global level.
* Be aware that when you enable or disable a trace flag, it’s quite possible that you can affect performance, depending on what functionality the trace flag changes.
* Both commands (DBCC TRACEON and DBCC TRACEOFF) require the sysadmin role.
* DBCC TRACESTATUS provides status for a specific trace flag, or all of them, and notes whether they are enabled for a session or globally.
* Running DBCC TRACESTATUS does not affect performance or data, nor does it alter the configuration of the instance.
* DBCC TRACESTATUS only needs the public role to be run.

Now, Let us see some examples of using these commands:

Use the following code to list all trace flags enabled just for this connection:



|  |  |
| --- | --- |
| 1  2 | DBCC TRACESTATUS ();  GO |

Use this code to list all trace flags enabled globally:



|  |  |
| --- | --- |
| 1  2 | DBCC TRACESTATUS (-1);  GO |

Follow the following scenario to test both of DBCC TRACEON and DBCC TRACEOFF

Run a full backup operation for one of your databases:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | DECLARE @BackupPath NVARCHAR (100);  SET @BackupPath = 'C:\Backup\SQL\_Shack2014\_' +  REPLACE (CONVERT (nvarchar (19), SYSDATETIME (), 126), ':','') + '.bak';    BACKUP DATABASE [SQL\_Shack2014]  TO DISK = @BackupPath  WITH NOFORMAT,  INIT,  COMPRESSION,  STATS = 05;  GO |

Then run a transaction log backup operation on the same database:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | DECLARE @BackupPath NVARCHAR (100);  SET @BackupPath = 'C:\Backup\SQL\_Shack2014\_' +  REPLACE (CONVERT (nvarchar (19), SYSDATETIME (), 126), ':','') + '.trn';    BACKUP LOG [SQL\_Shack2014]  TO DISK = @BackupPath  WITH NOFORMAT,  INIT,  COMPRESSION,  STATS = 05;  GO |

Now, when you check the ERRORLOG, you will notice entries to the SQL Server error log for every successful backup operation (local connection) as in the following figure:

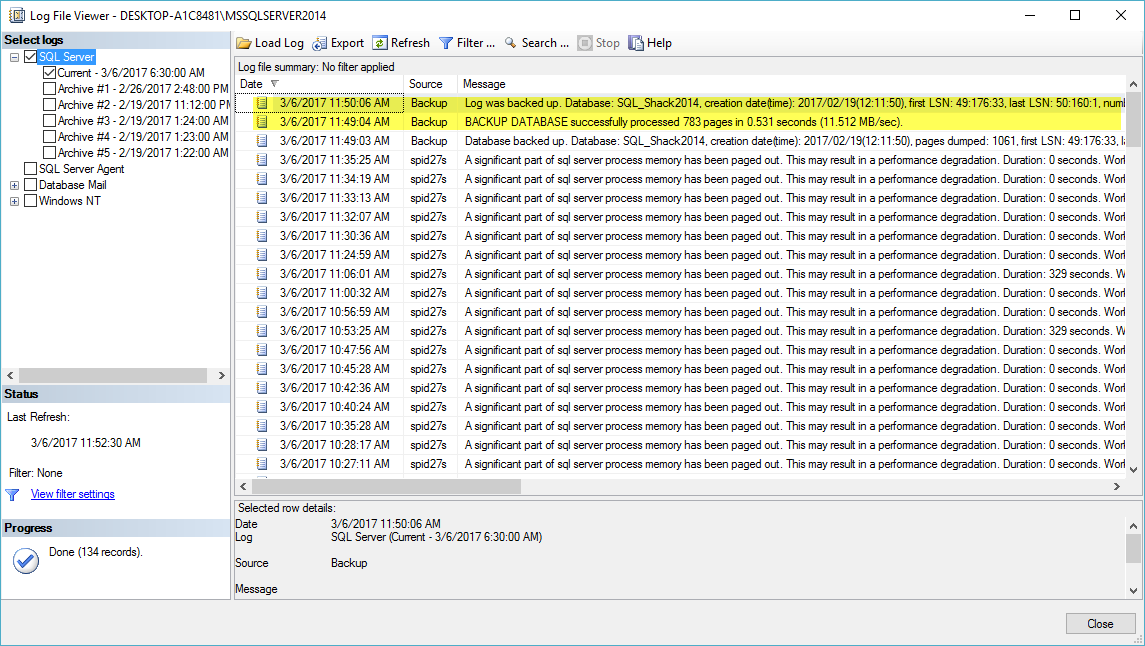
[](https://www.sqlshack.com/wp-content/uploads/2017/03/word-image-266.png)<img src="/wp-content/uploads/2017/03/word-image-266.png" />

Figure 3 Log file viewer (1)

We can suppress all successful in SQL Server error log by enabling trace flag 3226 using DBCC TRACEON command:

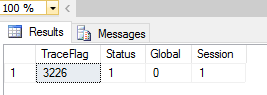


|  |  |
| --- | --- |
| 1  2 | DBCC TRACEON (3226);  GO |

Then we can use DBCC TRACESTATUS command to verify that the trace is enabled:



|  |  |
| --- | --- |
| 1  2 | DBCC TRACESTATUS ();  GO |

<img style="margin: 0px auto; display: block;" src="/wp-content/uploads/2017/03/word-image-267.png" />

Run a transaction log Backup operation again and check SQL Server log to see the impact of enabling this trace flag:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | DECLARE @BackupPath NVARCHAR(100);  SET @BackupPath = 'C:\Backup\SQL\_Shack2014\_' +  REPLACE (CONVERT (nvarchar (19), SYSDATETIME (), 126), ':','') + '.trn';    BACKUP LOG [SQL\_Shack2014]  TO DISK = @BackupPath  WITH NOFORMAT,  INIT,  COMPRESSION,  STATS = 05;  GO |

You will notice that there is a log record for enabling the trace flag 3226, and the last transaction log backup has not logged as the following figure:

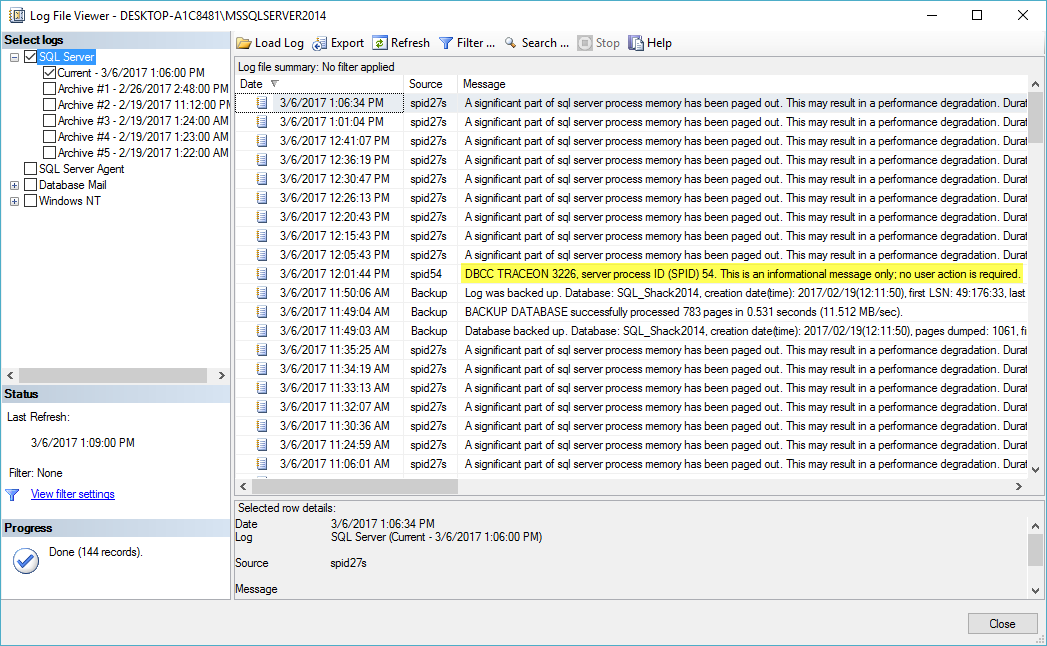
[](https://www.sqlshack.com/wp-content/uploads/2017/03/word-image-268.png)<img src="/wp-content/uploads/2017/03/word-image-268.png" />

Figure 4 Log file viewer (2)

You can disable this trace flag using the following code:



|  |  |
| --- | --- |
| 1  2 | DBCC TRACEOFF (3226);  GO |

To make these commands affect on SQL server instance globally which will be effective till the next restart of the instance, you can use the following codes:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*  Enable trace flag (3226) globally  \*/  DBCC TRACEON (3226, -1);  GO  /\*  Verify again that the trace flag is enabled globally  \*/  DBCC TRACESTATUS (-1);  GO  /\*  Turn off 3226 globally  \*/  DBCC TRACEOFF (3226,-1);  GO |

**Informational Commands:**

These commands will not change the configuration of our environment, but they will give us information about our environment.

**DBCC SQLPERF:**

* ProvidesTransaction log space usage for all log files on an instance
* Used to clear out data related to weight statistics, latch statistics, or spinlock statistics
* To run this command, you need to have( **VIEW SERVER STATE, ALTER SERVER STATE** )
* Execution does not affect system performance

To check log space utilization:



|  |  |
| --- | --- |
| 1 | DBCC SQLPERF (LOGSPACE); |

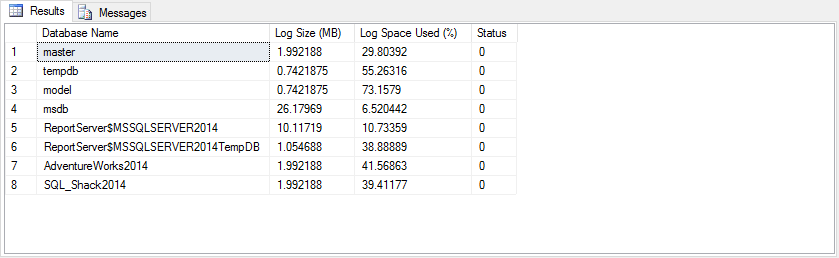
[](https://www.sqlshack.com/wp-content/uploads/2017/03/word-image-269.png)<img src="/wp-content/uploads/2017/03/word-image-269.png" />

Figure 5 Result of DBCC SQLPERF Command

To clear wait statistics:



|  |  |
| --- | --- |
| 1 | DBCC SQLPERF ("sys.dm\_os\_wait\_stats", CLEAR); |

To clear latch statistics:



|  |  |
| --- | --- |
| 1 | DBCC SQLPERF ("sys.dm\_os\_latch\_stats", CLEAR); |

To clear spinlock statistics:



|  |  |
| --- | --- |
| 1 | DBCC SQLPERF ("sys.dm\_os\_spinlock\_stats", CLEAR); |

**DBCC SHOW\_STATISTICS:**

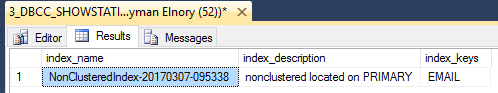
* Displays statistics object for an index, indexed view, or column statistic
* Use when troubleshooting and understanding the estimates from a plan and comparing against actual values
* Viewing statistics does not introduce a performance load on the system
* Require sysadmin, db\_owner, or db\_ddladmin roles, or tablet ownership

To implement this DBCC Command go through the following steps:

First, you need to view all indexes contained in the specified table:



|  |  |
| --- | --- |
| 1  2  3  4 | USE [SQL\_Shack2014];  GO  sp\_helpindex '[dbo]. [Students]';  GO |

<img style="margin: 0px auto; display: block;" src="/wp-content/uploads/2017/03/word-image-270.png" />

Then you can use DBCC SHOW\_STATISTICS against the index name you got in step 1:



|  |  |
| --- | --- |
| 1  2 | DBCC SHOW\_STATISTICS ('[dbo]. [Students]','NonClusteredIndex-20170307-095338');  GO |

The result displays the header, histogram, and density vector based on data stored in the statistics object.

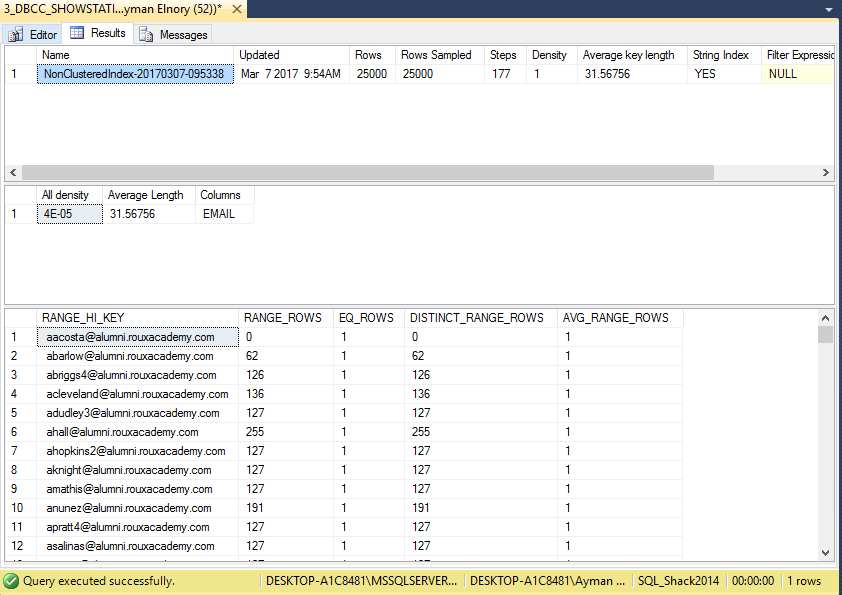
[](https://www.sqlshack.com/wp-content/uploads/2017/03/word-image-271.png)<img src="/wp-content/uploads/2017/03/word-image-271.png" />

Figure 6 Result of DBCC SHOW\_STATISTICS Command

And you can specify what section of the result you want to see by using one of the following options:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | DBCC SHOW\_STATISTICS ('[dbo]. [Students]','NonClusteredIndex-20170307-095338') WITH STAT\_HEADER;  GO    DBCC SHOW\_STATISTICS ('[dbo]. [Students]','NonClusteredIndex-20170307-095338') WITH DENSITY\_VECTOR;  GO    DBCC SHOW\_STATISTICS ('[dbo]. [Students]','NonClusteredIndex-20170307-095338') WITH HISTOGRAM;  GO |

**DBCC USEROPTIONS:**

* Returns options set for the current connection like isolation level or QUOTED\_IDENTIFIER
* Use to verify setting for connection to confirm they are correct, or consistent across different connect methods
* Only displays information, does not modify settings
* Require public role



|  |  |
| --- | --- |
| 1  2 | DBCC USEROPTIONS;  GO |

