

Auto Generate Create Table Script Based on SQL Server Query

By: [Daniel Farina \(/sqlserverauthor/111/daniel-farina/\)](#) | Updated: 2018-10-05 | [Comments \(1\)](#) | Related: [More \(/sql-server-developer-resources/\)](#) > [Functions - User Defined UDF \(/sql-server-tip-category/160/functions--user-defined-udf/\)](#)

Problem

You have to create a SQL Server table to store the results from a query. At first you think about looking at each column data type to create the table, but realize it will be a tedious task. In this tip we look at a function you can use to generate a create table script that has the correct data types for each column based on the source columns.

Solution

We as SQL Server database professionals write queries each day and some circumstances we need the query to store the data in a table so we can do further analysis or we need to get the data from the query from different sources into one common table. In such case, we can't use a SELECT INTO statement to create the destination table. This is especially important in case we are working inside a stored procedure that uses a temporary table that needs to be populated with a SELECT statement twice.

For the purposes of this tip, I will use the query from my previous tip [Create SQL Server Disk Space Report for All Servers \(/sqlservertip/5040/create-sql-server-disk-space-report-for-all-servers/\)](#) as an example.

T-SQL Script to Generate a Table based on a Query

During the course of this tip we will go through the process of creating a scalar function that takes a SELECT statement as a parameter and returns the CREATE TABLE script for the query, so you can pass it as a parameter to an EXEC or sp_executesql statement.

In order to achieve this, we will be using the Dynamic Management Function sys.dm_exec_describe_first_result_set.

The DMF sys.dm_exec_describe_first_result_set was introduced in SQL Server 2012. This function has the following parameters: @tsql, @params and @browse_information_mode. The table below has the descriptions of the parameters as found at the following Microsoft help [link \(https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-describe-first-result-set-transact-sql\)](https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-describe-first-result-set-transact-sql).

Parameter	Description
tsql	The Transact SQL script or batch.
params	It provides a declaration string for parameters for the Transact-SQL batch, which is similar to sp_executesql. Parameters may be nvarchar(n) or nvarchar(max).
browse_information_mode	<p>Specifies if additional key columns and source table information are returned. If set to 1, each query is analyzed as if it includes a FOR BROWSE option on the query. Additional key columns and source table information are returned.</p> <ul style="list-style-type: none">• If set to 0, no information is returned.• If set to 1, each query is analyzed as if it includes a FOR BROWSE option on the query. This will return base table names as the source column information.• If set to 2, each query is analyzed as if it would be used in preparing or executing a cursor. This will return view names as source column information.



SELECT statement using this function and using the query from this tip [Create SQL Server Disk Space Report for All Servers](http://www.wuolap.com/vertip/5040/create-sql-server-disk-space-report-for-all-servers/) (http://www.wuolap.com/vertip/5040/create-sql-server-disk-space-report-for-all-servers/). Note, I had to make all single quotes into double single quotes since the query is passed as a parameter.

```
USE AdventureWorks2012
GO
SELECT * FROM sys.dm_exec_describe_first_result_set ( '
SELECT @@SERVERNAME [Server] ,
       DB_NAME() [Database] ,
       MF.name [File Name] ,
       MF.type_desc [Type] ,
       MF.physical_name [Path] ,
       CAST(CAST(MF.size / 128.0 AS DECIMAL(15, 2)) AS VARCHAR(50)) + '' MB'' [File Size] ,
       CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - ( ( size / 128.0 ) - CAST(FILEPROPERTY(MF.name, ''SPACEUSED'')) AS DECIMAL(10, 2)) AS DECIMAL(10, 2)) AS VARCHAR(50)) + '' MB'' [Free Space] ,
       CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, ''SPACEUSED'')) AS DECIMAL(10, 2)) AS DECIMAL(10, 2)) AS VARCHAR(50)) + '' MB'' [Used Space] ,
       CAST(CONVERT(DECIMAL(10, 2), ( ( MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, ''SPACEUSED'')) AS DECIMAL(10, 2)) / (MF.size / 128.0) * 100 AS DECIMAL(10, 2)) AS DECIMAL(10, 2)) AS VARCHAR(50)) + '' %'' [Percent Used] ,
       IIF(MF.growth = 0, ''N/A'',
          CASE WHEN MF.is_percent_growth = 1 THEN CAST(MF.growth AS VARCHAR(50)) + ''%''
              ELSE CAST(MF.growth / 128 AS VARCHAR(50)) + '' MB''
              END) [Autogrowth] ,
       VS.volume_mount_point ,
       CAST(CAST(VS.total_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50)) + '' GB'' [Total Space] ,
       CAST(CAST(VS.available_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50)) + '' GB'' [Available Space] ,
       CAST(CAST(VS.available_bytes / CAST(VS.total_bytes AS DECIMAL(20, 2)) * 100 AS DECIMAL(20, 2)) AS DECIMAL(20, 2)) AS VARCHAR(50)) + '' %'' [Percent Available]
FROM sys.database_files MF
CROSS APPLY sys.dm_os_volume_stats(DB_ID(''?'), MF.file_id) VS',NULL, null)
```

As you can see in the next screen capture, the output of this function is a table that contains a detailed description of each column of the query that we provided as the parameter.

ORIOM.AdventureWorks2012 (ORIOM\Daniel (76))*

(/AdventureWorks2012

MENU

```

SELECT * FROM sys.dm_exec_describe_first_result_set (
SELECT
  @@SERVERNAME [Server] ,
  DB_NAME() [Database] ,
  MF.name [File Name] ,
  MF.type_desc [Type] ,
  MF.physical_name [Path] ,
  CAST(CAST(MF.size / 128.0 AS DECIMAL(15, 2)) AS VARCHAR(50)) + ' MB' [File Size] ,
  CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - ( ( size / 128.0 ) - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0 ))
  AS VARCHAR(50)) + ' MB' [File Used Space] ,
  CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0) AS VARCHAR(50))
  + ' MB' [File Free Space] ,
  CAST(CONVERT(DECIMAL(10, 2), ( ( MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0 ) /
  ( MF.size / 128.0 ) ) * 100) AS VARCHAR(50)) + '%' [% Free File Space] ,
  IIF(MF.growth = 0, 'N/A', CASE WHEN MF.is_percent_growth = 1 THEN CAST(MF.growth AS VARCHAR(50)) + '%'
  ELSE CAST(MF.growth / 128 AS VARCHAR(50)) + ' MB'
  END) [Autogrowth] ,

VS.volume_mount_point ,
CAST(CAST(VS.total_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50))
+ ' GB' [Total Volume Size] ,
CAST(CAST(VS.available_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50))
+ ' GB' [Free Space] ,
CAST(CAST(VS.available_bytes / CAST(VS.total_bytes AS DECIMAL(20, 2))
* 100 AS DECIMAL(20, 2)) AS VARCHAR(50)) + '%' [% Free]

FROM
  sys.database_files MF
CROSS APPLY sys.dm_os_volume_stats(DB_ID(''), MF.file_id) VS',NULL, null)

```

100 %

Results Messages

	is_hidden	column_ordinal	name	is_nullable	system_type_id	system_type_name	max_length	precision	scale	collation_name	user_type_id	us
1	0	1	Server	1	231	nvarchar(128)	256	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
2	0	2	Database	1	231	nvarchar(128)	256	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
3	0	3	File Name	0	231	nvarchar(128)	256	0	0	SQL_Latin1_General_CP1_CI_AS	256	A
4	0	4	Type	1	231	nvarchar(50)	120	0	0	Latin1_General_CI_AS_KS_WS	NULL	N
5	0	5	Path	1	231	nvarchar(260)	520	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
6	0	6	File Size	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
7	0	7	File Used Space	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
8	0	8	File Free Space	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
9	0	9	% Free File Space	1	167	varchar(51)	51	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
10	0	10	Autogrowth	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
11	0	11	volume_mount_point	1	231	nvarchar(256)	512	0	0	Modern_Spanish_CI_AS	NULL	N
12	0	12	Total Volume Size	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
13	0	13	Free Space	1	167	varchar(53)	53	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N
14	0	14	% Free	1	167	varchar(51)	51	0	0	SQL_Latin1_General_CP1_CI_AS	NULL	N

Query executed successfully.

ORIOM (14.0 RTM) ORIOM\Daniel (76) AdventureWorks2012 00:00:00 14 rows

In the next table you will see a description of each output column. You can see the full table at this [link](https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-describe-first-result-set-transact-sql) (<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-exec-describe-first-result-set-transact-sql>).

For our purposes we will only use the following columns: name, is_nullable, system_type_name, collation_name and is_hidden to filter because we don't want columns that don't appear in the result set.

Column	Description
is_hidden	Specifies that the column is an extra column added for browsing and informational purposes that does not actually appear in the result set.
column_ordinal	Contains the ordinal position of the column in the result set. Position of the first column will be specified as 1.
name	Contains the name of the column if a name can be determined. If not, will contain NULL.
is_nullable	Contains the following values: Value 1 if column allows NULLs. Value 0 if the column does not allow NULLs. Value 1 if it cannot be determined that the column allows NULLs.
system_type_id	Contains the system_type_id of the column data type as specified in sys.types. For CLR types, even though the system_type_name column will return NULL, this column will return the value 240.
system_type_name	Contains the name and arguments (such as length, precision, scale), specified for the data type of the column. If data type is a user-defined alias type, the underlying system type is specified here. If data type is a CLR user-defined type, NULL is returned in this column.
max_length	Maximum length (in bytes) of the column. -1 = Column data type is varchar(max), nvarchar(max), varbinary(max), or xml. For text columns, the max_length value will be 16 or the value set by sp_tableoption 'text in row'.
precision	Precision of the column if numeric-based. Otherwise returns 0.
scale	Scale of column if numeric-based. Otherwise returns 0.
collation_name	Name of the collation of the column if character-based. Otherwise returns NULL.



(L)

user_type_database	For CLR and alias types, contains the user_type_id of the data type of the column as specified in sys.types. Otherwise is NULL.	MENU
	For CLR and alias types, contains the name of the database in which the type is defined. Otherwise is NULL.	
user_type_schema	For CLR and alias types, contains the name of the schema in which the type is defined. Otherwise is NULL.	
user_type_name	For CLR and alias types, contains the name of the type. Otherwise is NULL.	
assembly_qualified_type_name	For CLR types, returns the name of the assembly and class defining the type. Otherwise is NULL.	
xml_collection_id	Contains the xml_collection_id of the data type of the column as specified in sys.columns. This column returns NULL if the type returned is not associated with an XML schema collection.	
xml_collection_database	Contains the database in which the XML schema collection associated with this type is defined. This column returns NULL if the type returned is not associated with an XML schema collection.	
xml_collection_schema	Contains the schema in which the XML schema collection associated with this type is defined. This column returns NULL if the type returned is not associated with an XML schema collection.	
xml_collection_name	Contains the name of the XML schema collection associated with this type. This column returns NULL if the type returned is not associated with an XML schema collection.	
is_xml_document	Returns 1 if the returned data type is XML and that type is guaranteed to be a complete XML document (including a root node), as opposed to an XML fragment). Otherwise returns 0.	
is_case_sensitive	Returns 1 if the column is of a case-sensitive string type. Returns 0 if it is not.	
is_fixed_length_clr_type	Returns 1 if the column is of a fixed-length CLR type. Returns 0 if it is not.	
source_server	Name of the originating server (if it originates from a remote server). The name is given as it appears in sys.servers. Returns NULL if the column originates on the local server or if it cannot be determined which server it originates on. Is only populated if browsing information is requested.	
source_database	Name of the originating database returned by the column in this result. Returns NULL if the database cannot be determined. Is only populated if browsing information is requested.	
source_schema	Name of the originating schema returned by the column in this result. Returns NULL if the schema cannot be determined. Is only populated if browsing information is requested.	
source_table	Name of the originating table returned by the column in this result. Returns NULL if the table cannot be determined. Is only populated if browsing information is requested.	
source_column	Name of the originating column returned by the result column. Returns NULL if the column cannot be determined. Is only populated if browsing information is requested.	
is_identity_column	Returns 1 if the column is an identity column and 0 if not. Returns NULL if it cannot be determined that the column is an identity column.	
is_part_of_unique_key	Returns 1 if the column is part of a unique index (including unique and primary constraints) and 0 if it is not. Returns NULL if it cannot be determined that the column is part of a unique index. Is only populated if browsing information is requested.	
is_updateable	Returns 1 if the column is updateable and 0 if not. Returns NULL if it cannot be determined that the column is updateable.	
is_computed_column	Returns 1 if the column is a computed column and 0 if not. Returns NULL if it cannot be determined if the column is a computed column.	
is_sparse_column_set	Returns 1 if the column is a sparse column and 0 if not. Returns NULL if it cannot be determined that the column is a part of a sparse column set.	
ordinal_in_order_by_list	The position of this column is in ORDER BY list. Returns NULL if the column does not appear in the ORDER BY list, or if the ORDER BY list cannot be uniquely determined.	
order_by_list_length	The length of the ORDER BY list. NULL is returned if there is no ORDER BY list or if the ORDER BY list cannot be uniquely determined. Note that this value will be the same for all rows returned by sp_describe_first_result_set.	
order_by_is_descending	If the ordinal_in_order_by_list is not NULL, the order_by_is_descending column reports the direction of the ORDER BY clause for this column. Otherwise it reports NULL.	
error_number	Contains the error number returned by the function. If no error occurred, the column will contain NULL.	
error_severity	Contains the severity returned by the function. If no error occurred, the column will contain NULL.	
error_state	Contains the state message. returned by the function. If no error occurred, the column will contain NULL.	
error_message	Contains the message returned by the function. If no error occurred, the column will contain NULL.	
error_type	Contains an integer representing the error being returned. Maps to error_type_desc. See the list under remarks.	
error_type_desc	Contains a short uppercase string representing the error being returned. Maps to error_type. See the list under remarks.	

SQL Server Function to Generate Create Table Script

Here is the code of the function.



```
FUNCTION fn_Table_Structure (@InputSQL AS NVARCHAR(4000), @TableName AS NVARCHAR(128) = NULL)
RETURNS NVARCHAR(4000)
```

```
AS
BEGIN

DECLARE @SQL AS NVARCHAR(4000)
DECLARE @name NVARCHAR(128)
DECLARE @is_nullable BIT
DECLARE @system_type_name NVARCHAR(128)
DECLARE @collation_name NVARCHAR(128)
DECLARE @NewLine NVARCHAR(2) = CHAR(13) + CHAR(10) -- CRLF

DECLARE CUR_Table CURSOR LOCAL FAST_FORWARD
FOR
    SELECT  name ,
            is_nullable ,
            system_type_name ,
            collation_name
    FROM    sys.dm_exec_describe_first_result_set(@InputSQL, NULL, NULL)
    WHERE   is_hidden = 0
    ORDER BY column_ordinal ASC

OPEN CUR_Table

FETCH NEXT FROM CUR_Table INTO @name, @is_nullable, @system_type_name,
    @collation_name

SET @SQL = 'CREATE TABLE [' + ISNULL(@TableName, 'TableName') + '] ( '
    + @NewLine

WHILE @@FETCH_STATUS = 0
    BEGIN
        SET @SQL += @NewLine + '[' + @name + ']' + ' ' + @system_type_name
            + CASE WHEN @collation_name IS NOT NULL
                THEN ' COLLATE ' + @collation_name + ' '
                ELSE ''
            END + CASE WHEN @is_nullable = 0 THEN ' NOT NULL '
                ELSE ''
            END + ', '
        FETCH NEXT FROM CUR_Table INTO @name, @is_nullable, @system_type_name,
            @collation_name
    END

SET @SQL = LEFT(@SQL, LEN(@SQL) - 1) + @NewLine + ')'


CLOSE CUR_Table
DEALLOCATE CUR_Table

RETURN @SQL
end
```

The way this function works by declaring a cursor for each row returned by the sys.dm_exec_describe_first_result_set Dynamic Management Function.

Parameters

This function receives the following parameters:

 (L)	Description MENU
InputSQL	The Transact SQL script or batch that will be used to create the table structure.
TableName	The name of the table you want to generate from the create table script.

The function returns the create table statement based on the query passed as the parameter. It includes the definition of nullable columns as well as the collation for the string columns.

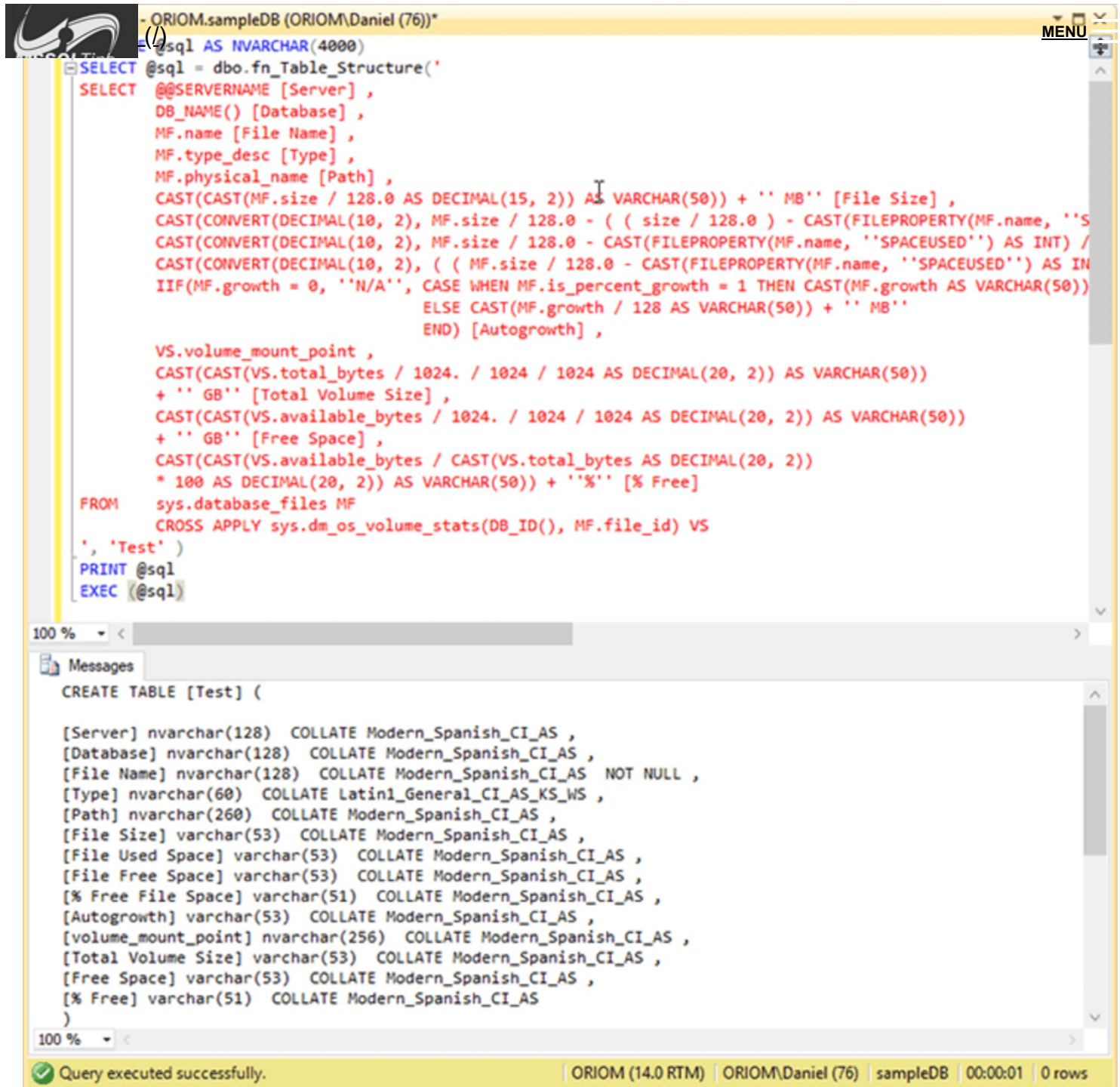
Here is an example of its use. Note, since the query is passed as a parameter, I had to make each single quote into two single quotes for the original query.

```

DECLARE @sql AS NVARCHAR(4000)
SELECT @sql = dbo.fn_Table_Structure('
SELECT @@SERVERNAME [Server] ,
        DB_NAME() [Database] ,
        MF.name [File Name] ,
        MF.type_desc [Type] ,
        MF.physical_name [Path] ,
        CAST(CAST(MF.size / 128.0 AS DECIMAL(15, 2)) AS VARCHAR(50)) + ' ' MB' [File Size] ,
        CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - ( ( size / 128.0 ) - CAST(FILEPROPERTY(MF.name
        CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT)
        CAST(CONVERT(DECIMAL(10, 2), ( ( MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED')
        IIF(MF.growth = 0, 'N/A',
            CASE WHEN MF.is_percent_growth = 1 THEN CAST(MF.growth AS VARCHAR(50)) + '% '
            ELSE CAST(MF.growth / 128 AS VARCHAR(50)) + ' MB'
            END) [Autogrowth] ,
        VS.volume_mount_point ,
        CAST(CAST(VS.total_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50)) + ' ' GB' [
        CAST(CAST(VS.available_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50)) + ' ' GB
        CAST(CAST(VS.available_bytes / CAST(VS.total_bytes AS DECIMAL(20, 2)) * 100 AS DECIMAL(20, 2))
FROM sys.database_files MF
        CROSS APPLY sys.dm_os_volume_stats(DB_ID(), MF.file_id) VS
', 'Test' )
PRINT @sql
EXEC (@sql)

```

On the next image you can see the execution of the function. This will display the results as well as create the table.



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a T-SQL query that uses the `fn_Table_Structure` function to generate a `CREATE TABLE` script for a table named `[Test]`. The query selects various file and volume statistics from the `sys.database_files` and `sys.dm_os_volume_stats` views, formatting them into a table structure with appropriate data types and collations. The bottom pane shows the output of the query execution, which is the generated `CREATE TABLE` script. The status bar at the bottom indicates that the query was executed successfully.

```

-- ORION.sampleDB (ORION\Daniel (76))*
-- sql AS NVARCHAR(4000)
SELECT @sql = dbo.fn_Table_Structure(
    SELECT @@SERVERNAME [Server] ,
           DB_NAME() [Database] ,
           MF.name [File Name] ,
           MF.type_desc [Type] ,
           MF.physical_name [Path] ,
           CAST(CAST(MF.size / 128.0 AS DECIMAL(15, 2)) AS VARCHAR(50)) + ' MB' [File Size] ,
           CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - ( ( size / 128.0 ) - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0 ) AS DECIMAL(10, 2)) AS VARCHAR(50)) [File Used Space] ,
           CAST(CONVERT(DECIMAL(10, 2), MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0 ) AS DECIMAL(10, 2)) AS VARCHAR(50)) [File Free Space] ,
           CAST(CONVERT(DECIMAL(10, 2), ( ( MF.size / 128.0 - CAST(FILEPROPERTY(MF.name, 'SPACEUSED') AS INT) / 128.0 ) AS DECIMAL(10, 2)) AS DECIMAL(10, 2)) AS VARCHAR(50)) + '%' [% Free] ,
           IIF(MF.growth = 0, 'N/A', CASE WHEN MF.is_percent_growth = 1 THEN CAST(MF.growth AS VARCHAR(50))
                                     ELSE CAST(MF.growth / 128 AS VARCHAR(50)) + ' MB'
                                END) [Autogrowth] ,
           VS.volume_mount_point ,
           CAST(CAST(VS.total_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50))
           + ' GB' [Total Volume Size] ,
           CAST(CAST(VS.available_bytes / 1024. / 1024 / 1024 AS DECIMAL(20, 2)) AS VARCHAR(50))
           + ' GB' [Free Space] ,
           CAST(CAST(VS.available_bytes / CAST(VS.total_bytes AS DECIMAL(20, 2))
               * 100 AS DECIMAL(20, 2)) AS VARCHAR(50)) + '%' [% Free]
    FROM sys.database_files MF
    CROSS APPLY sys.dm_os_volume_stats(DB_ID(), MF.file_id) VS
    , 'Test' )
PRINT @sql
EXEC (@sql)

```

```

CREATE TABLE [Test] (
    [Server] nvarchar(128) COLLATE Modern_Spanish_CI_AS ,
    [Database] nvarchar(128) COLLATE Modern_Spanish_CI_AS ,
    [File Name] nvarchar(128) COLLATE Modern_Spanish_CI_AS NOT NULL ,
    [Type] nvarchar(60) COLLATE Latin1_General_CI_AS_KS_WS ,
    [Path] nvarchar(260) COLLATE Modern_Spanish_CI_AS ,
    [File Size] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [File Used Space] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [File Free Space] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [% Free File Space] varchar(51) COLLATE Modern_Spanish_CI_AS ,
    [Autogrowth] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [volume_mount_point] nvarchar(256) COLLATE Modern_Spanish_CI_AS ,
    [Total Volume Size] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [Free Space] varchar(53) COLLATE Modern_Spanish_CI_AS ,
    [% Free] varchar(51) COLLATE Modern_Spanish_CI_AS
)

```

Query executed successfully. | ORION (14.0 RTM) | ORION\Daniel (76) | sampleDB | 00:00:01 | 0 rows

Next Steps

- If you don't understand why I included the collation on the `CREATE TABLE` script, take a look at this tip [Create SQL Server temporary tables with the correct collation](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/) ([/sqlservertip/2440/create-sql-server-temporary-tables-with-the-correct-collation/](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/)).
- Take a look at [SQL Server Dynamic Management Views and Functions Tips Category](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/) ([/sql-server-tip-category/31/dynamic-management-views-and-functions/](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/)).
- Are you new to cursors? Take a look at this tip for an introduction: [SQL Server Cursor Example](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/) ([/sqlservertip/1599/sql-server-cursor-example/](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/)).
- Take a look at [SQL Server Functions - User Defined UDF Tips Category](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/) ([/sql-server-tip-category/160/functions---user-defined-udf/](https://sqlservertip.com/2014/04/create-sql-server-temporary-tables-with-the-correct-collation/)) for more ideas.



About the author



(/sqlserverauthor/111/daniel-farina/) Daniel Farina was born in Buenos Aires, Argentina. Self-educated, since childhood he showed a passion for learning.

View all my tips (/sqlserverauthor/111/daniel-farina/)

Related Resources

- [More Database Developer Tips...](/sql-server-developer-resources/) (/sql-server-developer-resources/)

Follow

- [Get Free SQL Tips](/get-free-sql-server-tips/?ref=GetFooterMenu) (/get-free-sql-server-tips/?ref=GetFooterMenu)
- [Twitter](https://twitter.com/mssqltips) (https://twitter.com/mssqltips)
- [LinkedIn](https://www.linkedin.com/groups/2320891/) (https://www.linkedin.com/groups/2320891/)
- [Facebook](https://www.facebook.com/mssqltips/) (https://www.facebook.com/mssqltips/)
- [Pinterest](https://www.pinterest.com/mssqltips/) (https://www.pinterest.com/mssqltips/)
- [RSS](https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips) (https://feeds.feedburner.com/MSSQLTips-LatestSqlServerTips)

Learning

- [DBAs](/sql-server-dba-resources/) (/sql-server-dba-resources/)
- [Developers](/sql-server-developer-resources/) (/sql-server-developer-resources/)
- [BI Professionals](/sql-server-business-intelligence-resources/) (/sql-server-business-intelligence-resources/)
- [Careers](/sql-server-professional-development-resources/) (/sql-server-professional-development-resources/)
- [Today's Tip](/todays-sql-server-tip/) (/todays-sql-server-tip/)

Resources

- [Tutorials](/sql-server-tutorials/) (/sql-server-tutorials/)
- [Webcasts](/sql-server-webcasts/) (/sql-server-webcasts/)
- [Whitepapers](/sql-server-whitepapers/) (/sql-server-whitepapers/)
- [Tools](/sql-server-tools/) (/sql-server-tools/)

Search

- [Tip Categories](/sql-server-categories/) (/sql-server-categories/)
- [Search By TipID](/search-tip-id/) (/search-tip-id/)
- [Authors](/sql-server-mssqltips-authors/) (/sql-server-mssqltips-authors/)

Community

- [First Timer?](/learn-more-about-mssqltips/) (/learn-more-about-mssqltips/)
- [Pictures](/mssqltips-community/1/) (/mssqltips-community/1/)
- [Contribute](/contribute/) (/contribute/)
- [Event Calendar](/sql-server-event-list/) (/sql-server-event-list/)
- [User Groups](/sql-server-user-groups/) (/sql-server-user-groups/)
- [Author of the Year](/mssqltips-author-of-year/) (/mssqltips-author-of-year/)

More Info

- [Join](/get-free-sql-server-tips/?ref=JoinFooterMenu) (/get-free-sql-server-tips/?ref=JoinFooterMenu)
- [About](/about/) (/about/)
- [Copyright](/copyright/) (/copyright/)
- [Privacy](/privacy/) (/privacy/)
- [Disclaimer](/disclaimer/) (/disclaimer/)
- [Feedback](/feedback/) (/feedback/)
- [Advertise](/advertise/) (/advertise/)