

# How to change schema of all tables, views and stored procedures in MSSQL

Asked 11 years, 5 months ago   Modified 1 year, 1 month ago

Viewed 105k times



45



Recently we were having issues on our database server and after long efforts it was decided to change the database server. So we managed to restore the database on another server, change the connection string, etc.

Everything was going as planned until we tried to access the website from a web browser.

We started getting errors about database objects not being found. Later we found out that it occurred as a result of the modified schema name. Since there are hundreds of database objects (tables, views and stored procedures) in a Kentico database, it is not feasible to change all of them manually, one-by-one. Is there a practical way of doing this?

sql

sql-server-2008

database-schema

database-migration

kentico

Share

edited Jan 14, 2014 at 15:34

Improve this question

Follow

asked Jul 10, 2013 at 12:56



[anar khalilov](#)

17.5k ● 9 ● 50 ● 63

Are you sure you just haven't set your users default schema up incorrectly? – [Nick.Mc](#) May 29, 2015 at 5:28

The question is nearly two years old but as far as I can remember it wasn't related to user's default schema.

– [anar khalilov](#) May 29, 2015 at 5:32

Oh! for some reason it appeared at the top of the list!

– [Nick.Mc](#) May 29, 2015 at 6:11

Readers - Also see this Microsoft answer:

[support.managed.com/kb/a100/...](https://support.managed.com/kb/a100/...) – [Yogi](#) May 7, 2018 at 14:58

I faced the same issue recently. what causes this the schema of objects to be changed while creating a new DB instance with existing user, and assigning privileges to the new instance? – [Mohamed Iqzas](#) Nov 14, 2019 at 2:39

5 Answers

Sorted by:

Highest score (default)



**Yes, it is possible.**

122

To change the schema of a database object you need to run the following SQL script:



```
ALTER SCHEMA NewSchemaName TRANSFER OldSchemaName.ObjectName
```





Where ObjectName can be the name of a table, a view or a stored procedure. The problem seems to be getting the list of all database objects with a given shcema name. Thankfully, there is a system table named sys.Objects that stores all database objects. The following query will generate all needed SQL scripts to complete this task:

```
SELECT 'ALTER SCHEMA NewSchemaName TRANSFER [' + SysSc
DbObjects.Name + '];'
FROM sys.Objects DbObjects
INNER JOIN sys.Schemas SysSchemas ON DbObjects.schema_
WHERE SysSchemas.Name = 'OldSchemaName'
AND (DbObjects.Type IN ('U', 'P', 'V'))
```

Where type 'U' denotes user tables, 'V' denotes views and 'P' denotes stored procedures.

Running the above script will generate the SQL commands needed to transfer objects from one schema to another. Something like this:

```
ALTER SCHEMA NewSchemaName TRANSFER OldSchemaName.CONT
ALTER SCHEMA NewSchemaName TRANSFER
OldSchemaName.Proc_Analytics_Statistics_Delete;
ALTER SCHEMA NewSchemaName TRANSFER
OldSchemaName.Proc_CMS_QueryProvider_Select;
ALTER SCHEMA NewSchemaName TRANSFER OldSchemaName.COM_
ALTER SCHEMA NewSchemaName TRANSFER OldSchemaName.CMS_
ALTER SCHEMA NewSchemaName TRANSFER OldSchemaName.Poll
```

Now you can run all these generated queries to complete the transfer operation.

Follow

answered Jul 10, 2013 at 12:56



anar khalilov

17.5k ● 9 ● 50 ● 63

- 
- 1 Nice solution. I used "WHERE SysSchemas.Name <> 'dbo'" instead to list all non-dbo bound objects. – [Mackan](#) Nov 4, 2014 at 10:14
- 

And 'SO' denotes sequences – [Edgar](#) May 30, 2016 at 17:49

---

- 1 For **all Tables** , check [this](#) and [this](#) to do it in single statement, hope helps some one. – [Shaiju T](#) Jul 3, 2016 at 9:45
- 
- 3 want to transfer functions as well, you can add 'FN','TF' to in list. 'FN' for Scalar function, and 'TF' for Table function. for more, ref: [msdn.microsoft.com/en-us/library/ms177596.aspx](https://msdn.microsoft.com/en-us/library/ms177596.aspx) – [Will Wu](#) Nov 16, 2016 at 2:25
- 
- 1 This is just perfect answer. Thanks @anar for save my time. – [Raju Paladiya](#) May 9, 2020 at 17:09
- 



9



Here's the SQL I ran, to move all tables in my database (spread across several schemas) into the "dbo" schema:

```
DECLARE
    @currentSchemaName nvarchar(200),
    @tableName nvarchar(200)

DECLARE tableCursor CURSOR FAST_FORWARD FOR
SELECT TABLE_SCHEMA, TABLE_NAME
FROM information_schema.tables
ORDER BY 1, 2
```

```

DECLARE @SQL nvarchar(400)

OPEN tableCursor
FETCH NEXT FROM tableCursor INTO @currentSchemaName, @

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @SQL = 'ALTER SCHEMA dbo TRANSFER ' + @current
@tableName
    PRINT @SQL

    EXEC (@SQL)

    FETCH NEXT FROM tableCursor INTO @currentSchemaNam
END

CLOSE tableCursor
DEALLOCATE tableCursor

```

Phew!

Share Improve this answer

answered May 29, 2015 at 13:52

Follow



Mike Gledhill

29.1k ● 8 ● 157 ● 162



You may use following script by just copy/paste for all objects

7

**NOTE:** You need to Change schema names in script !



```

DECLARE @OldSchema VARCHAR(200)
DECLARE @NewSchema VARCHAR(200)
DECLARE @SQL nvarchar(4000)
SET @OldSchema = 'dbo'
SET @NewSchema = 'Inf'

DECLARE tableCursor CURSOR FAST_FORWARD FOR
SELECT 'ALTER SCHEMA [' + @NewSchema + '] TRANSFER

```

```

'].[ ' + DbObjects.Name + '];' AS Cmd
FROM sys.Objects DbObjects
INNER JOIN sys.Schemas SysSchemas ON DbObjects.sch
SysSchemas.schema_id
WHERE SysSchemas.Name = @OldSchema
AND (DbObjects.Type IN ('U', 'P', 'V'))
OPEN tableCursor
FETCH NEXT FROM tableCursor INTO @SQL
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT @SQL
    EXEC (@SQL)
    FETCH NEXT FROM tableCursor INTO @SQL
END
CLOSE tableCursor
DEALLOCATE tableCursor
PRINT '*** Finished ***'

```

Share Improve this answer

answered Apr 13, 2017 at 7:33

Follow



**Haseeb Ahmed**

243 ● 2 ● 11



5



Thanks for the tip.. Here is my update to same, where I added a crlf to output as well as put brackets around the SchemaName and ObjectName, because some of the objects had a '-' in the name and the brackets solved that naming error.



```

SELECT 'ALTER SCHEMA NewSchemaName TRANSFER [' + SysSc
DbObjects.Name + '];'
+ CHAR(13)+ CHAR(10)+ 'GO ' + CHAR(13)+ CHAR(10)
FROM sys.Objects DbObjects
INNER JOIN sys.Schemas SysSchemas ON DbObjects.schema_
SysSchemas.schema_id
WHERE SysSchemas.Name = 'OldSchemaName'
AND (DbObjects.Type IN ('U', 'P', 'V'))

```

Share Improve this answer

edited May 29, 2015 at 5:26

Follow



anar khalilov

17.5k ● 9 ● 50 ● 63

answered May 28, 2015 at 16:28



Hank Freeman

1,212 ● 10 ● 7



1



Here is my take on combining anar's and Hank's answers.

***I also changed the names of the system views and columns as I am using SQL Server 2019 and it seems that Microsoft has changed the first-letter case for all of these objects.***

```
SELECT
    'ALTER SCHEMA NewSchemaName TRANSFER [' + ss.name
+ CHAR(13) + CHAR(10) + CHAR(13) + CHAR(10)
FROM
    sys.objects so
INNER JOIN sys.schemas ss ON
    so.schema_id = ss.schema_id
WHERE
    ss.name = 'OldSchemaName'
AND (so.type IN ('U', 'P', 'V'))
```

Share Improve this answer

edited Oct 25, 2023 at 9:10

Follow



Maged Mostafa

51 ● 4

answered Oct 25, 2023 at 7:57

