How to delete all rows from all tables in a SQL Server database?

Asked 9 years, 9 months ago Active 11 months ago Viewed 162k times



How to delete all rows from all tables in a SQL Server database?

sql-server-2005





58



asked Dec 14 '09 at 9:19 surajit khamrai

See codeguru.com/forum/showthread.php?t=458182 and scroll down... – Wim ten Brink Dec 14 '09 at 9:24

by drop database will be deleted i just want to reset data - surajit khamrai Dec 14 '09 at 9:27

11 Answers



Note that TRUNCATE won't work if you have any referential integrity set.

251

In that case, this will work:



```
EXEC sp MSForEachTable 'DISABLE TRIGGER ALL ON ?'
EXEC sp MSForEachTable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'
EXEC sp_MSForEachTable 'DELETE FROM ?'
EXEC sp MSForEachTable 'ALTER TABLE ? CHECK CONSTRAINT ALL'
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email







- 1 nice. good thinking.... Preet Sangha Dec 14 '09 at 9:28
- 1 and what about triggers ??????? surajit khamrai Dec 14 '09 at 9:30
- 1 Actually, that's only for DDL triggers. In which case: EXECP sp_MSForEachTable 'DISABLE TRIGGER ALL ON ?' Mark Rendle Dec 14 '09 at 9:34
- 9 Not available in SQL Azure :(Akash Kava Apr 17 '13 at 12:31
- Got it if a backup file already exists then it looks like SSMS appends to it rather than replacing it (I didn't realize this). So I deleted the file and now the 'empty' database backup file is only 3.7 MB Ben Dec 8 '14 at 15:34



In my recent project my task was to clean an entire database by using sql statement and each table having many constraints like Primary Key and Foreign Key. There are more than 1000 tables in database so its not possible to write a delete query on each and ever table.



By using a stored procedure named <u>sp_MSForEachTable</u> which allows us to easily process some code against each and every table in a single database. It means that it is used to process a single T-SQL command or a different T-SQL commands against every table in the database.

So follow the below steps to truncate all tables in a SQL Server Database:

Step 1- Disable all constraints on the database by using below sql query :

EXEC sys.sp msforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'

Step 2- Execute a Delete or truncate operation on each table of the database by using below sql command:

EXEC sys.sp msforeachtable 'DELETE FROM ?'

Step 3- Enable all constraints on the database by using below sql statement:

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email







4,533 5 33



401 4 4

1 You can simply execute step 2 multiple times so that first time it deletes tables with non dependencies, 2nd time to delete those tables failed in first time, 3rd time to delete faild in 2nd time, etc – user586399 Jun 3 '16 at 22:10

any ideas on how to do this on sql server azure ? - Zapnologica Jan 30 '18 at 8:36

This approach will work also in Azure as it uses only plain SQL: sqlrelease.com/delete-all-rows-from-all-tables – Jakob Lithner Nov 19 '18 at 12:37 🖍



I had to delete all the rows and did it with the next script:

17

```
DECLARE @Nombre NVARCHAR(MAX);
DECLARE curso CURSOR FAST FORWARD
FOR
Select Object name(object id) AS Nombre from sys.objects where type = 'U'
OPEN curso
FETCH NEXT FROM curso INTO @Nombre
WHILE (@@FETCH_STATUS <> -1)
BEGIN
IF (@@FETCH_STATUS <> -2)
BEGIN
DECLARE @statement NVARCHAR(200);
SET @statement = 'DELETE FROM ' + @Nombre;
print @statement
execute sp executesql @statement;
END
FETCH NEXT FROM curso INTO @Nombre
END
CLOSE curso
DEALLOCATE curso
```

Hope this helps!

answered Oct 3 '12 at 17:27

M' Conza Oviada

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email







Here is a solution that:

9

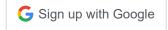
- 1. Drops constraints (thanks to this post)
- 2. Iterates through INFORMATION_SCHEMA.TABLES for a particular database
- 3. SELECTS tables based on some search criteria
- 4. Deletes all the data from those tables
- 5. Re-adds constraints
- 6. Allows for ignoring of certain tables such as sysdiagrams and RefactorLog

I initially tried EXECUTE sp_MSforeachtable 'TRUNCATE TABLE ?', but that deleted my diagrams.

```
USE <DB name>;
-- Disable all constraints in the database
EXEC sp msforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"
declare @catalog nvarchar(250);
declare @schema nvarchar(250);
declare @tbl nvarchar(250);
DECLARE i CURSOR LOCAL FAST FORWARD FOR select
                                        TABLE CATALOG,
                                        TABLE SCHEMA,
                                        TABLE NAME
                                        from INFORMATION_SCHEMA.TABLES
                                        where
                                        TABLE TYPE = 'BASE TABLE'
                                        AND TABLE NAME != 'sysdiagrams'
                                        AND TABLE NAME != ' RefactorLog'
OPEN i;
FETCH NEXT FROM i INTO @catalog, @schema, @tbl;
WHILE @@FETCH STATUS = 0
   BEGIN
        DECLARE @sql NVARCHAR(MAX) = N'DELETE FROM [' + @catalog + '].[' + @schema + '].
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email





```
CLOSE i;
DEALLOCATE i;
-- Re-enable all constraints again
EXEC sp msforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"
```

edited May 3 '18 at 10:35

answered Mar 1 '17 at 9:17



This is great but it doesn't take non-dbo schemas into account. - influent Jan 19 '18 at 22:19

I've never used non dbo schemas, so I wouldn't catch that. But why doesn't it work? I'm not specifying schema anywhere so does it default to dbo only?

– Zach Smith Jan 20 '18 at 18:32

If you have a table, for example, called test. Table 1, where "test" is the schema, your deletes will fail if trying to execute "DELETE FROM Table 1". It needs to be DELETE FROM test. Table 1. – influent Jan 22 '18 at 19:03

2 @influent - now it takes non-dbo schemas into account - Zach Smith Jan 30 '18 at 12:19

Unfortunately this seems to fail if there are FK constraints. The ALTER TABLE bit to disable constraints fails. - Douglas Gaskell Nov 10 '18 at 2:48 /



Set nocount on



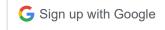
Exec sp_MSForEachTable 'Alter Table ? NoCheck Constraint All'



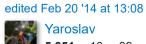
Exec sp_MSForEachTable
'
If ObjectProperty(Object_ID(''?''), ''TableHasForeignRef'')=1
Begin
-- Just to know what all table used delete syntax.
Print ''Delete from '' + ''?''
Delete From ?
End
Else
Begin
-- Just to know what all table used Truncate syntax.

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email







answered Jan 3 '13 at 8:08





In my case, I needed to set QUOTED_IDENTIFIER on. This led to a slight modification of Mark Rendle's answer above:

3

```
EXEC sp_MSForEachTable 'DISABLE TRIGGER ALL ON ?'

GO

EXEC sp_MSForEachTable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'

GO

EXEC sp_MSForEachTable 'SET QUOTED_IDENTIFIER ON; DELETE FROM ?'

GO

EXEC sp_MSForEachTable 'ALTER TABLE ? CHECK CONSTRAINT ALL'

GO

EXEC sp_MSForEachTable 'ENABLE TRIGGER ALL ON ?'

GO
```

answered Apr 9 '18 at 14:43





You could delete all the rows from all tables using an approach like Rubens suggested, or you could just drop and recreate all the tables. Always a good idea to have the full db creation scripts anyway so that may be the easiest/quickest method.

1



answered Dec 14 '09 at 9:26



seems OP is concerned about referential integrity and triggers; this case, your got best solution. I'm dropping my answer =) – Rubens Farias Dec 14 '09 at 9:31

Join Stack Overflow to learn, share knowledge, and build your career.









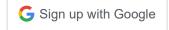
0

For some requirements we might have to skip certain tables. I wrote the below script to add some extra conditions to filter the list of tables. The below script will also display the pre delete count and post delete count.

```
IF OBJECT ID('TEMPDB..#TEMPRECORDCOUNT') IS NOT NULL
DROP TABLE #TEMPRECORDCOUNT
CREATE TABLE #TEMPRECORDCOUNT
        TABLENAME NVARCHAR (128)
        , PREDELETECOUNT BIGINT
        , POSTDELETECOUNT BIGINT
INSERT INTO #TEMPRECORDCOUNT (TABLENAME, PREDELETECOUNT, POSTDELETECOUNT)
SELECT O.name TableName
        ,DDPS.ROW COUNT PREDELETECOUNT
        ,NULL FROM sys.objects 0
INNER JOIN (
            SELECT OBJECT ID, SUM(row count) ROW COUNT
           FROM SYS.DM DB PARTITION STATS
            GROUP BY OBJECT ID
          ) DDPS ON DDPS.OBJECT ID = 0.0BJECT ID
WHERE O.type = 'U' AND O.name NOT LIKE 'OC%' AND O.schema id = 1
DECLARE @TableName NVARCHAR(MAX);
DECLARE TableDeleteCursor CURSOR FAST FORWARD
FOR
SELECT TableName from #TEMPRECORDCOUNT
OPEN TableDeleteCursor
FETCH NEXT FROM TableDeleteCursor INTO @TableName
WHILE (@@FETCH STATUS <> -1)
BEGIN
IF (@@FETCH STATUS <> -2)
BEGIN
DECLARE @STATEMENT NVARCHAR(MAX);
SET @STATEMENT = ' DISABLE TRIGGER ALL ON ' + @TableName +
                 '; ALTER TABLE ' + @TableName + ' NOCHECK CONSTRAINT ALL' +
                 ': DELETE FROM ' + @TableName +
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email





```
END
CLOSE TableDeleteCursor
DEALLOCATE TableDeleteCursor
UPDATE T
SET T.POSTDELETECOUNT = I.ROW COUNT
 FROM #TEMPRECORDCOUNT T
INNER JOIN (
                SELECT O.name TableName, DDPS.ROW_COUNT ROW_COUNT
                FROM sys.objects 0
                INNER JOIN (
                        SELECT OBJECT ID, SUM(row count) ROW COUNT
                        FROM SYS.DM_DB_PARTITION_STATS
                        GROUP BY OBJECT ID
                       ) DDPS ON DDPS.OBJECT ID = O.OBJECT ID
                WHERE O.type = 'U' AND O.name NOT LIKE 'OC%' AND O.schema id = 1
           ) I ON I.TableName COLLATE DATABASE DEFAULT = T.TABLENAME
SELECT * FROM #TEMPRECORDCOUNT
ORDER BY TABLENAME ASC
```

answered Mar 16 '17 at 16:10



Balasubramanian S



This answer builds on Zach Smith's answer by resetting the identity column as well:



1. Disabling all constraints



2. Iterating through all tables except those you choose to exclude

- 3. Deletes all rows from the table
- 4. Resets the identity column if one exists
- 5. Re-enables all constraints

Join Stack Overflow to learn, share knowledge, and build your career.







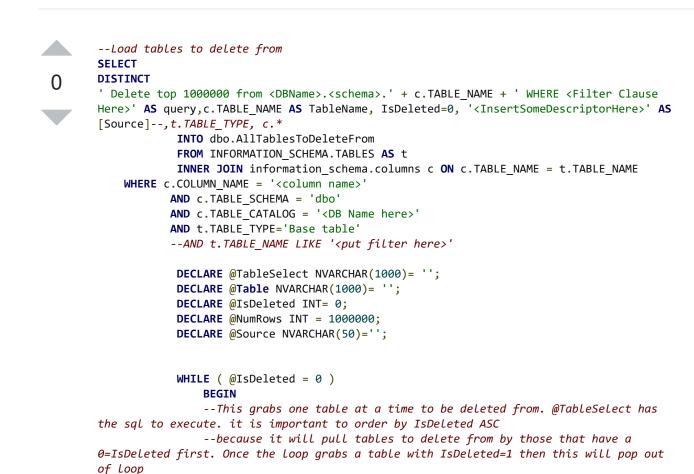
```
declare @catalog nvarchar(250);
declare @schema nvarchar(250);
declare @tbl nvarchar(250);
DECLARE i CURSOR LOCAL FAST FORWARD FOR select
                                        TABLE CATALOG,
                                        TABLE SCHEMA,
                                        TABLE NAME
                                        from INFORMATION SCHEMA.TABLES
                                        where
                                        TABLE TYPE = 'BASE TABLE'
                                        AND TABLE NAME != 'sysdiagrams'
                                        AND TABLE NAME != ' RefactorLog'
                                        -- Optional
                                        -- AND (TABLE SCHEMA = 'dbo')
OPEN i;
FETCH NEXT FROM i INTO @catalog, @schema, @tbl;
WHILE @@FETCH STATUS = 0
   BEGIN
        DECLARE @sql NVARCHAR(MAX) = N'DELETE FROM [' + @catalog + '].[' + @schema + '].
[' + @tbl + '];'
        /* Make sure these are the commands you want to execute before executing */
        PRINT 'Executing statement: ' + @sql
        -- EXECUTE sp executesql @sql
        -- Reset identity counter if one exists
        IF ((SELECT OBJECTPROPERTY( OBJECT ID(@catalog + '.' + @schema + '.' + @tbl),
'TableHasIdentity')) = 1)
        BEGIN
           SET @sql = N'DBCC CHECKIDENT ([' + @catalog + '.' + @schema + '.' + @tbl +
'], RESEED, 0)'
            PRINT 'Executing statement: ' + @sql
            -- EXECUTE sp executesql @sql
        END
        FETCH NEXT FROM i INTO @catalog, @schema, @tbl;
   END
CLOSE i;
DEALLOCATE i;
-- Re-enable all constraints again
EXEC sp msforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



For one reason or another this mostly fails as it throws FK constraint errors. - Douglas Gaskell Nov 10 '18 at 2:46



```
SELECT TOP 1
     @TableSelect = query,
     @IsDeleted = IsDeleted,
     @Table = TableName,
     @Source=[a].[Source]
FROM     dbo.AllTablesToDeleteFrom a
WHERE a.[Source]='SomeDescriptorHere'--use only if needed
```

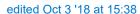
Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email





```
WHILE (@NumRows = 1000000) -- only delete a million rows at a time?
                    BEGIN
                    EXEC sp executesql @TableSelect;
                    SET @NumRows = @@ROWCOUNT;
                    --IF @NumRows = 1000000 --can do something here if needed
                    --One wants this loop to continue as long as a million rows is
deleted. Once < 1 million rows is deleted it pops out of loop
                    --and grabs next table to delete
                          BEGIN
                    --SELECT @NumRows; -- can add this in to see current number of
deleted records for table
                            INSERT INTO dbo.DeleteFromAllTables
                                    ( tableName,
                                      query,
                                      cnt,
                                      [Source]
                            SELECT @Table,
                                    @TableSelect,
                                    @NumRows,
                                    @Source;
                          END:
                END;
SET @NumRows = 1000000;
UPDATE a
SET
        a.IsDeleted = 1
FROM
        dbo.AllTablesToDeleteFrom a
WHERE
       a.TableName = @Table;
--flag this as deleted so you can move on to the next table to delete from
END;
```









Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email







edited Apr 23 '12 at 19:56



MADCookie

answered Apr 23 '12 at 19:25



this doesn't accomplishes the initial requirement, to delete ALL TABLES... - franko_camron Jan 28 '16 at 16:32

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



