Database Administrators

How do I drop all constraints from all tables?

29

I want to drop all default constraints, check constraints, unique constraints, primary keys and foreign keys from all tables in a SQL
Server database. I know how to get all the constraint names from `sys.objects`, but how do I populate the `ALTER TABLE` part?

sql-server   ddl   constraint

7

asked Jan 23 '15 at 18:08

Aaron Bertrand ♦
157k   19   312   516

3   @srutzky It was asked on Stack Overflow but I decided to create a cleaner, canonical version here. Anyway it is a common request, often part of a larger task of cleaning out a database (starting over, cleaning up objects that were mistakenly put into master, etc). I don't see these functional dependencies being impacted by dropping the constraints - in fact I suspect that in most cases the larger picture is truncating or dropping the tables, too. Dropping the constraints first allows that. –   Aaron Bertrand ♦   Jan 25 '15 at 20:23 ✎

2 Answers

You can derive this information easily by joining `sys.tables.object_id = sys.objects.parent_object_id` for those object types.

35

```sql
DECLARE @sql NVARCHAR(MAX);
SET @sql = N'';

SELECT @sql = @sql + N'
  ALTER TABLE ' + QUOTENAME(s.name) + N'.'
  + QUOTENAME(t.name) + N' DROP CONSTRAINT '
  + QUOTENAME(c.name) + ';'
FROM sys.objects AS c
INNER JOIN sys.tables AS t
ON c.parent_object_id = t.[object_id]
INNER JOIN sys.schemas AS s
ON t.[schema_id] = s.[schema_id]
WHERE c.[type] IN ('D','C','F','PK','UQ')
ORDER BY c.[type];

PRINT @sql;
--EXEC sys.sp_executesql @sql;
```

`PRINT` is just there for eyeballing - if you have a *lot* of constraints, it may not show the entire script because it is limited to 8K. In those cases, see this tip for other ways to validate the script before running.

Once you are happy with the output, uncomment the `EXEC`.

edited Jan 25 '15 at 18:42   answered Jan 23 '15 at 18:08

Aaron Bertrand ♦
157k   19   312   516

3   You might also want to make sure to drop foreign key constraints before primary keys; `ORDER BY (CASE WHEN c.[type] IN ('PK', 'UQ') THEN 1`

I started with the accepted answer and modified the structure to use a while loop rather than to build the full sql statement in dynamic sql. I like this better for several reasons.

5

The query is not stored in the large @sql variable. This implementation allows for a Print for each constraint that is dropped for logging purposes in the output. Execution seemed a little faster in my unit testing.

```sql
Set NoCount ON

Declare @schemaName varchar(200)
set @schemaName=''
Declare @constraintName varchar(200)
set @constraintName=''
Declare @tableName varchar(200)
set @tableName=''

While exists
(
    SELECT c.name
    FROM sys.objects AS c
    INNER JOIN sys.tables AS t
    ON c.parent_object_id = t.[object_id]
    INNER JOIN sys.schemas AS s
    ON t.[schema_id] = s.[schema_id]
    WHERE c.[type] IN ('D','C','F','PK','UQ')
    and t.[name] NOT IN ('__RefactorLog', 'sysdiagrams')
    and c.name > @constraintName
)

Begin
    -- First get the Constraint
    SELECT
        @constraintName=min(c.name)
    FROM sys.objects AS c
    INNER JOIN sys.tables AS t
    ON c.parent_object_id = t.[object_id]
    INNER JOIN sys.schemas AS s
    ON t.[schema_id] = s.[schema_id]
    WHERE c.[type] IN ('D','C','F','PK','UQ')
    and t.[name] NOT IN ('__RefactorLog', 'sysdiagrams')
    and c.name > @constraintName
```

```
        @schemaName = s.name
    FROM sys.objects AS c
    INNER JOIN sys.tables AS t
    ON c.parent_object_id = t.[object_id]
    INNER JOIN sys.schemas AS s
    ON t.[schema_id] = s.[schema_id]
    WHERE c.name = @constraintName

    -- Then Print to the output and drop the constraint
    Print 'Dropping constraint ' + @constraintName + '...'
    Exec('ALTER TABLE [' + @schemaName + N'].[' + @tableName + N'] DROP CONSTRAINT [' +
@constraintName + ']')
End

Set NoCount OFF
```

answered Jan 4 '17 at 17:40