Remove All Foreign Keys

Sedat Salman | Posted on 11 October 2011 | Comments | Briefcase | Print

To clear test data from a database we could use truncate commands but sometimes we need to get rid of foreign keys.

we could get constaint list from **INFORMATION_SCHEMA.TABLE_CONSTRAINTS** table.

for example :

we could remove foreign key by using this table and the following query.

```
while(exists(select 1 from INFORMATION_SCHEMA.TABLE_CONSTRAINTS where CONSTRAINT_TYPE
begin
        declare @sql nvarchar(2000)
        SELECT TOP 1 @sql=('ALTER TABLE ' + TABLE_SCHEMA + '.[' + TABLE_NAME
        + '] DROP CONSTRAINT [' + CONSTRAINT_NAME + ']')
        FROM information_schema.table_constraints
        WHERE CONSTRAINT_TYPE = 'FOREIGN KEY'
        exec (@sql)
end
```

SQL Server Tips

Comments

Posted by Anonymous on 12 October 2011

Pingback from Dew Drop – October 12, 2011 | Alvin Ashcraft's Morning Dew

Posted by Yitzchok Lavi on 23 October 2011

Surely as you remove them you would want to save the removed constraints so that you could restore them?

See stackoverflow.com/.../can-foreign-key-constraints-be-temporarily-disabled-using-t-sql where this is discussed.

I personally would prefer to disable the constraints and re-enable them rather than dropping and recreating them (disabling the prevent use of TRUNCATE TABLE - you'd have to use DELETE, but the constraints wouldn't be enforced)

Posted by Sam Meshesha on 23 November 2016

```
begin
set nocount on
set transaction isolation level read uncommitted
declare @table_schema varchar(255)
declare @table_name varchar(255)
declare @foregin_key varchar(255)
declare @cur_ cursor
declare @sql_command nvarchar(4000)
set @cur_ = cursor local scroll for
select
s.[name] as [table_schema],
```

```
ut.[name] as [table_name],
fk.[name] as [foregin_key]
from sys.objects as fk
join sys.objects as ut
on fk.parent_object_id = ut.object_id
and ut.[type] = 'U'
join sys.schemas as s
on ut.schema_id = s.schema_id
where fk.[type] = 'F'
open @cur_
fetch first from @cur_
into @table_schema, @table_name, @foregin_key
while (@@fetch_status = 0)
begin
set @sql_command = N'ALTER TABLE ' + QUOTENAME(@table_schema) + '.' + QUOTENAME(@table_name) +
N' DROP CONSTRAINT ' + QUOTENAME(@foregin_key)
execute sp_executesql @sql_command, N''
fetch next from @cur_
into @table_schema, @table_name, @foregin_key
end
close @cur_
deallocate @cur_
end
```

Leave a Comment

Please register or log in to leave a comment.