

SQL Server 2005: T-SQL to temporarily disable a trigger

Asked 16 years, 3 months ago Modified 7 years, 3 months ago

Viewed 54k times



Is it possible to disable a trigger for a batch of commands and then enable it when the batch is done?

53



I'm sure I could drop the trigger and re-add it but I was wondering if there was another way.



sql-server

sql-server-2005

t-sql

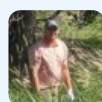


Share

Improve this question

Follow

asked Sep 23, 2008 at 20:13



Austin Salonen

50.2k ● 15 ● 111 ● 140

7 Answers

Sorted by:

Highest score (default)



```
DISABLE TRIGGER { [ schema_name . ] trigger_name [ , ...  
ON { object_name | DATABASE | ALL SERVER } [ ; ]
```

64



[http://msdn.microsoft.com/en-us/library/ms189748\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms189748(SQL.90).aspx)



followed by the inverse:

```
ENABLE TRIGGER { [ schema_name . ] trigger_name [ , ...  
ON { object_name | DATABASE | ALL SERVER } [ ; ]
```

[http://msdn.microsoft.com/en-us/library/ms182706\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms182706(SQL.90).aspx)

Share Improve this answer

Follow

edited Feb 4, 2011 at 19:24



sth

229k ● 56 ● 286 ● 368

answered Sep 23, 2008 at 20:14



Matt Rogish

24.9k ● 11 ● 78 ● 93

Thanks! I think my SQL Server 2005 naivety is showing too well on this site. – [Austin Salonen](#) Sep 23, 2008 at 20:15

No worries; be a DBA for a long time and these things become automatic :) – [Matt Rogish](#) Sep 23, 2008 at 20:16

i.e. if you use the option to modify a trigger and the script contains `ALTER TRIGGER [dbo].[trgWhatever] ON [dbo].[tblWhatever]` then you need `DISABLE TRIGGER [dbo].[trgWhatever] ON [dbo].[tblWhatever]` and `ENABLE TRIGGER [dbo].[trgWhatever] ON [dbo].[tblWhatever]` – [AjV Jsy](#) Sep 22, 2015 at 14:45

-
- 1 NOTE - I get an error 'Incorrect syntax near 'DISABLE'.' unless I place a `;` in between `PRINT 'Some message';` `DISABLE TRIGGER [dbo].....` – [AjV Jsy](#) Sep 23, 2015 at 15:22
-



41



Sometimes to populate an empty database from external data source or debug a problem in the database I need to disable ALL triggers and constraints. To do so I use the following code:

To disable all constraints and triggers:

```
sp_msforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"
sp_msforeachtable "ALTER TABLE ? DISABLE TRIGGER all"
```

To enable all constraints and triggers:

```
exec sp_msforeachtable @command1="print '?', @command2='CHECK CHECK CONSTRAINT all'
sp_msforeachtable @command1="print '?', @command2='ALTER TRIGGER all'"
```

I found that solution some time ago on [SQLServerCentral](#), but needed to modify the enable constraints part as the original one did not work fully

Share Improve this answer

edited Oct 2, 2008 at 9:16

Follow


answered Sep 23, 2008 at 21:20



kristof

53.8k ● 24 ● 88 ● 111

- 1 Very useful answer, but will the enabling query also enable ones that were previously disabled? A comment on [this answer](#) suggests using `select * from`

`sys.foreign_keys where is_disabled = 1` to find out if you have this scenario for foreign keys. – [MarkJ](#) Mar 20, 2023 at 14:45 



16



However, it is almost always a bad idea to do this. You will mess with the integrity of the database. Do not do it without considering the ramifications and checking with the dbas if you have them.

If you do follow Matt's code be sure to remember to turn the trigger back on. AND remember the trigger is disabled for everyone inserting, updating or deleting from the table while it is turned off, not just for your process, so if it must be done, then do it during the hours when the database is least active (and preferably in single user mode).

If you need to do this to import a large amount of data, then consider that bulk insert does not fire the triggers. But then your process after the bulk insert will have to fix up any data integrity problems you introduce by not firing the triggers.

Share Improve this answer

answered Sep 23, 2008 at 20:35

Follow



HLGEM

96.4k ● 15 ● 119 ● 189

- 1 Great points. The reason I needed this was when copying data from a production environment to a test environment some AKs were getting reset by the trigger but the related columns in another table weren't following suit.

– **Austin Salonen** Sep 23, 2008 at 21:12



To extend Matt's answer, here is an example given on [MSDN](#).

13



```
USE AdventureWorks;
GO
DISABLE TRIGGER Person.uAddress ON Person.Address;
GO
ENABLE Trigger Person.uAddress ON Person.Address;
GO
```

[Share](#) [Improve this answer](#)

answered Jul 4, 2013 at 5:38

[Follow](#)
**Daniel Imms**

50.1k ● 19 ● 155 ● 168



5

Another approach is to *effectively disable* the trigger without actually disabling it, using an additional state variable that is incorporated into the trigger.



```
create trigger [SomeSchema].[SomeTableIsEditableTrigge
[SomeTable]
for insert, update, delete
as
declare
    @isTableTriggerEnabled bit;

exec usp_IsTableTriggerEnabled -- Have to use USP inst
#temp
    @pTriggerProcedureIdOpt = @@procid,
    @poIsTableTriggerEnabled = @isTableTriggerEnabled

if (@isTableTriggerEnabled = 0)
    return;

-- Rest of existing trigger
go
```

For the state variable one could read some type of lock control record in a table (best if limited to the context of

the current session), use CONTEXT_INFO(), or use the presence of a particular temp table name (which is already session scope limited):

```
create proc [usp_IsTableTriggerEnabled]
    @pTriggerProcedureIdOpt  bigint          = null, -
    @pTableNameOpt           varchar(300)     = null, -
    @poIsTableTriggerEnabled bit              = null ou
begin

    set @poIsTableTriggerEnabled = 1; -- default retur

    -- Allow a particular session to disable all trigg
    -- temp tables are session scope limited).
    --
    if (object_id('tempdb..#Common_DisableTableTrigger
begin
    set @poIsTableTriggerEnabled = 0;
    return;
end

    -- Resolve table name if given trigger procedure i
    -- Google: "How to get the table name in the trigg
    --
    set @pTableNameOpt = coalesce(
        @pTableNameOpt,
        (select object_schema_name(parent_id) + '.' +
as tablename
        from sys.triggers
        where object_id = @pTriggerProcedureIdOpt)
    );

    -- Else decide based on logic involving @pTableNam
session
end
```

Then to disable all triggers:

```
select 1 as A into #Common_DisableTableTriggers;
-- do work
```

```
drop table #Common_DisableTableTriggers; -- or close c
```

A potentially major downside is that the trigger is permanently slowed down depending on the complexity of accessing of the state variable.

Edit: Adding a reference to this amazingly similar [2008 post by Samuel Vanga](#).

Share Improve this answer
Follow

edited Oct 3, 2013 at 0:47

answered Oct 2, 2013 at 23:53



[crokusek](#)

5,594 ● 3 ● 47 ● 62



5

```
ALTER TABLE table_name DISABLE TRIGGER TRIGGER_NAME
-- Here your SQL query
ALTER TABLE table_name ENABLE TRIGGER TRIGGER_NAME
```



Share Improve this answer
Follow

edited Aug 19, 2016 at 9:32



[Roman Marusyk](#)

24.5k ● 27 ● 80 ● 125

answered Aug 19, 2016 at 9:30



[ABHISHEK GANGULY](#)

51 ● 1 ● 1

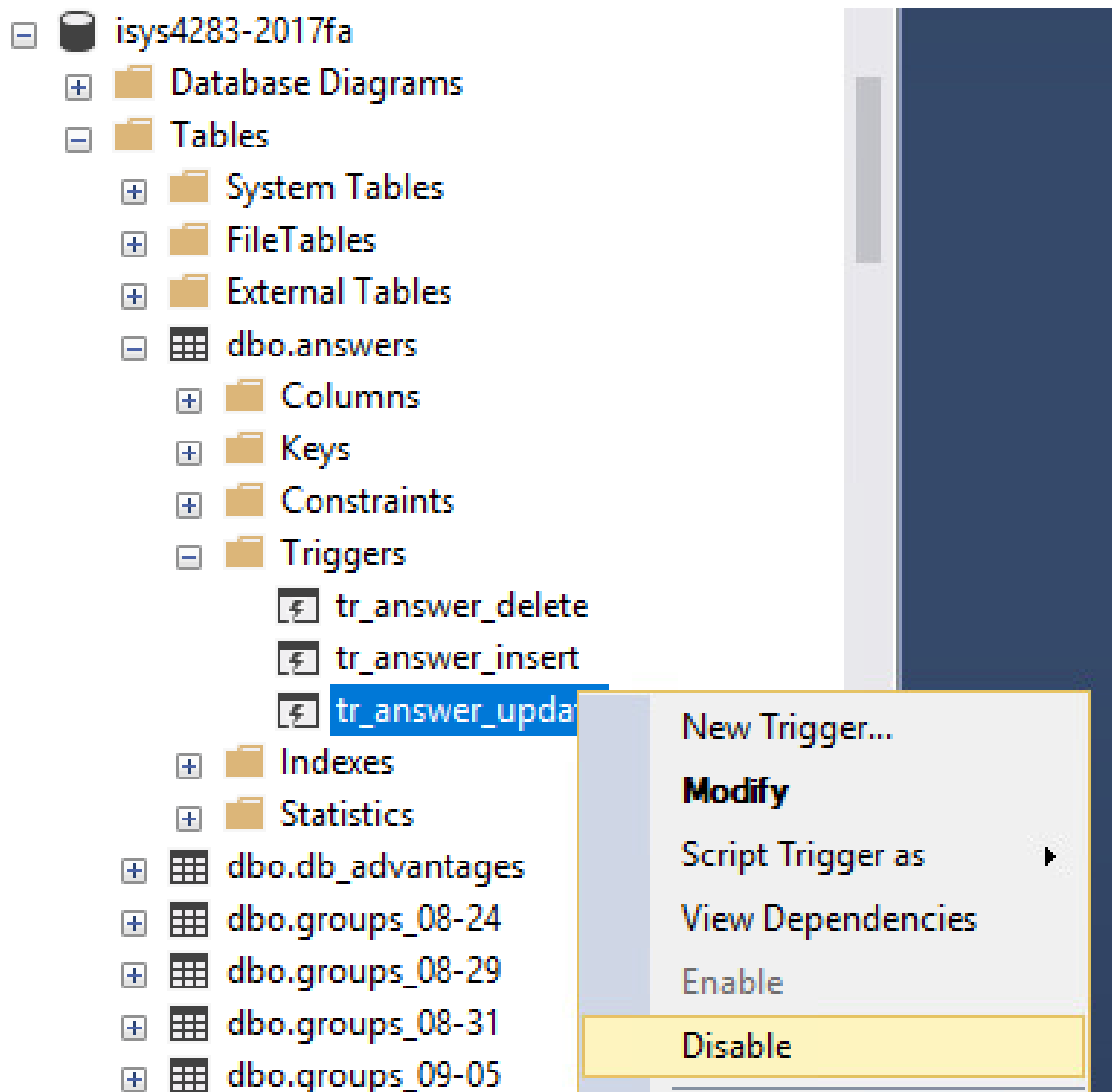


Not the best answer for batch programming, but for others finding this question in search of a quick and easy

2

way to temporarily disable a trigger, this can be accomplished in SQL Server Management Studio.

1. expand the triggers folder on the table
2. right-click the trigger
3. disable



Follow the same process to re-enable.

Share Improve this answer

answered Sep 9, 2017 at 18:00

Follow



[Jeff Puckett](#)

40.6k ● 19 ● 124 ● 172

