# How do I preview a destructive SQL query?

37

22

When writing destructive queries (e.g., DELETE or UPDATE) in SQL Server Management Studio I always find myself wishing I could preview the results of the query without actually running it. Access very conveniently allows you to do this, but I prefer to code my SQL by hand which, sadly, Access is very poor at.

So my question is twofold:

1. Is there an add-on for SSMS, or a separate tool that is furnished with good SQL hand-coding facilities that can also preview the result of a destructive query, similar to Access?
2. Are there any techniques or best practices for doing previews "by hand"; e.g., somehow using transactions?

It seems to me that doing this sort of thing is fundamentally important but yet I can't seem to find anything via Google (I'm probably just looking for the wrong thing - I am terribly ignorant on this matter). At present I'm taking a rather hairy belt and braces approach of commenting in and out select/delete/update lines and making sure I do backups. There's got to be a better way, surely?

Can anyone help?

sql sql-server

asked Jan 21 '09 at 12:28

Charles Roper
**12.1k** 18 61 93

stackoverflow.com/questions/281339/… – Rockcoder Jan 21 '09 at 12:30

## 5 Answers

I would use the OUTPUT clause present in SQL SERVER 2008 onwards...

**OUTPUT Clause (Transact-SQL)**

```
BEGIN TRANSACTION

DELETE [table] OUTPUT deleted.* WHERE [woof]

ROLLBACK TRANSACTION
```

INSERTs and UPDATEs can use the 'inserted' table too. The MSDN article covers it all.

**EDIT:**

This is just like other suggestions of SELECT then DELETE inside a transaction, except that it actually does both together. So you open a transaction, delete/insert/update with an OUTPUT clause, and the changes are made while ALSO outputting what was done. Then you can choose to rollback or commit.

edited Mar 9 '16 at 19:26
BJones
**1,662** 1 11 22

answered Jan 21 '09 at 12:40
MatBailie
**61.7k** 15 78 114

I prefer Kevin's solution because you can see the results without changing data. Select 100000 records is a whole bunch better for the database performance than delete 100000 records, output the results of deleted and then rollback. – HLGEM Jan 21 '09 at 19:15

2 But you're still open to mistakes. Like forgetting to highlight a where clause, etc. In this example you'd begin a transaction, make the change (which you fully intend to commit) OUPUT the changes and then leave the connection open with the transaction open. Once you have reviewed the output you can either COMMIT or ROLLBACK. This is much safer and much more reliable as it removes much more human error. It does mean the table is locked, so you need to be carefull still. – MatBailie Aug 13 '09 at 21:54

1 Order needs to be: DELETE [table] OUTPUT deleted.* WHERE [woof] – fractor Jan 14 '16 at 13:42

6

When deleting:

```
BEGIN TRANSACTION

   DELETE FROM table1
   OUTPUT deleted.*
   WHERE property1 = 99

ROLLBACK TRANSACTION
```

```
BEGIN TRANSACTION

   UPDATE table1
   SET table1.property1 = 99
   OUTPUT inserted.*

ROLLBACK TRANSACTION
```

edited Mar 10 '16 at 9:56

answered Aug 18 '15 at 4:57

kravits88

**7,795** 1 39 47

---

1

When I want to see what will be deleted, I just change the "delete" statement to a "select *". I like this better than using a transaction because I don't have to worry about locking.

answered Jan 21 '09 at 13:59



BankZ

**1,739** 1 9 8

1 I always try to put things in transactions any way. Just incase. Like highlighting everything except the where clause, hitting f5 and then staring blankly at the empty coffee mug... – MatBailie Jan 21 '09 at 15:01

---

24

I live in fear of someone doing this to my databases so I always ask my Team to do something like the following:

```
BEGIN TRAN

DELETE FROM X
-- SELECT * FROM X
FROM Table A as X JOIN Table B ON Blah blah blah
WHERE blah blah blah

ROLLBACK TRAN
COMMIT TRAN
```

it. If you delete the same number of records you expected, highlight the COMMIT TRAN and run it. If anything looks wonky, highlight the ROLLBACK TRAN and run it.

I do this with any UPDATE or DELETE statement. It's saved me a couple times, but it ALWAYS provides peace of mind.

answered Jan 21 '09 at 12:46

Kevin Buchan
**1,686** 2 21 29

I would put the DELETE and the SELECT the other way around for added safety. – Mr. Shiny and New 安宇 Jan 21 '09 at 14:33

2 The reason I do it in the order listed is so that I can highlight the BEGIN TRAN and the DELETE statement together. If the DELETE were commented out, I'd have to do them separately, but that's just developer preference. Thanks for the comment. – Kevin Buchan Jan 21 '09 at 14:41

3 This all works OK until someone forgets to COMMIT - and you end up blocking other users. I've seen that happen too many times, so be careful when using this strategy. – Scott Ivey Jan 21 '09 at 16:25

2

When you are in the context of the transaction, you can rollback changes any time before the transaction is committed. (Either by calling commit tran explicitly or if a condition arises that will cause the server to implicitly commit the transaction)

```
create table x (id int, val varchar(10))

insert into x values (1,'xxx')
begin tran

delete from x
select * from x

rollback tran
select * from x
```

edited Jan 21 '09 at 12:42

answered Jan 21 '09 at 12:35

cmsjr
**40.5k** 9 63 61