Common Table Expressions - Terminating the Previous Statement

Day 4 of Common Table Expression Month (June) at SteveStedman.com, today I will cover the topic of terminating the previous T-SQL statement.

These queries will be using that database that was set up in a previous posting on the CTE_DEMO Sample Database, if you haven't set up the sample database, download it and set it up now.

To start with having come from a background years ago of writing C programs, the concept of terminating statements is natural to me. I always feel a bit uncomfortable of a SQL Statement isn't properly terminated with a semicolon. If I am doing a T-SQL code review for someone else, I usually will make at least one comment about terminating T-SQL statements, unless it has already been done. I know that is not a requirement to terminate every SQL Statement, but it does make it easier to read. However in most cases, other than readability the termination of the previous statement really doesn't matter. . . In most cases.

For instance, the following two sets of queries work exactly the same with our without the semicolons.

Batch 1:

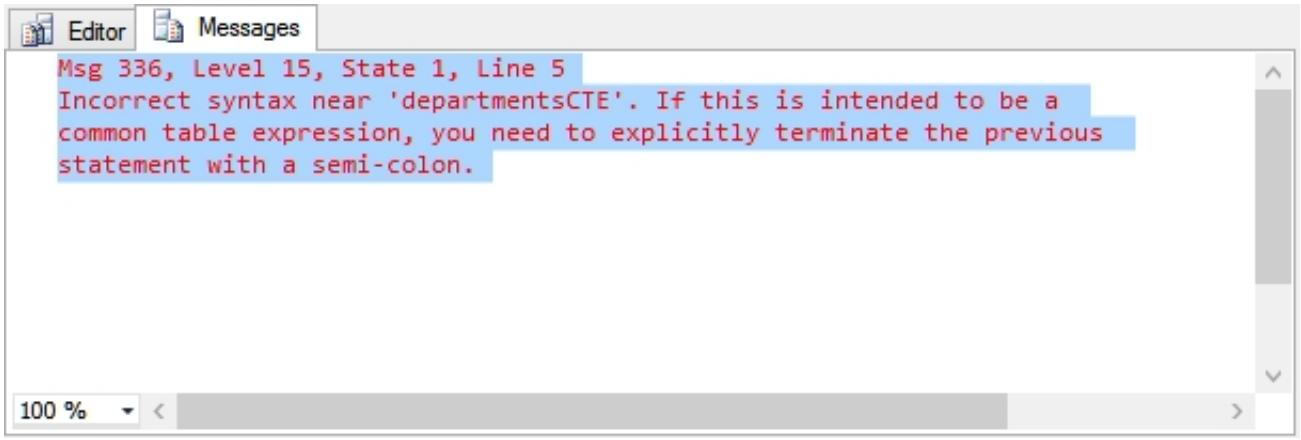
Batch 2:

In most cases, when writing multiple T-SQL statements in a batch, the semicolon to terminate the lines really doesn't matter, but there are a few cases in SQL Server where it does matter, and common table expressions are one of them.

Improper Termination

To start with lets take a look at the sample code from our previous posting titled "Writing Your First Common Table Expression with SQL Server". If we tried to run this SQL without any semicolons, it would run fine if it was run one line at a time, but if we run it as a batch, it throws an error.

When run produces the following results:




```
Msg 336, Level 15, State 1, Line 5
Incor
rect synta
x near 'departmentsC
TE'. If this is intended to be a
co
mmon
 table e
xpression, you n
eed to explicitly terminate the previous   statement with a semi-colon.
```

Which is a descriptive enough message stating that we need to explicitly terminate the previous statement with a semi-colon.

Proper Termination

When there is another T-SQL Statement in a batch prior to a CTE, that statement must be terminated with a semi-colon like this:

Note, the only required semi-colon is the one at the end of the SELECT * FROM Departments; query, and we get the desired results.

Editor | Results | Messages

	id	department	parent	archived
1	1	Camping	NULL	0
2	2	Cycle	NULL	0
3	3	Snowsports	NULL	0
4	4	Fitness	NULL	0
5	5	Tents	1	0
6	6	Backpacks	1	0
7	7	Sleeping Bags	1	0
8	8	Cooking	1	0
9	9	Bikes	2	0

	id	department	parent
1	1	Camping	NULL
2	2	Cycle	NULL
3	3	Snowsports	NULL
4	4	Fitness	NULL
5	5	Tents	1
6	6	Backpacks	1
7	7	Sleeping ...	1
8	8	Cooking	1
9	9	Bikes	2

GO Keyword in SSMS

When working in SQL Server Management Studio, an alternative is to terminate the batch, and the GO keyword is used to break a number of T-SQL statements into different batches, like this:

Editor | Results | Messages

	id	department	parent	archived
1	1	Camping	NULL	0
2	2	Cycle	NULL	0
3	3	Snowsports	NULL	0
4	4	Fitness	NULL	0
5	5	Tents	1	0
6	6	Backpacks	1	0
7	7	Sleeping Bags	1	0
8	8	Cooking	1	0
9	9	Bikes	2	0

	id	department	parent
1	1	Camping	NULL
2	2	Cycle	NULL
3	3	Snowsports	NULL
4	4	Fitness	NULL
5	5	Tents	1
6	6	Backpacks	1
7	7	Sleeping ...	1
8	8	Cooking	1
9	9	Bikes	2

Clean Code

Being an advocate for clean code, and clearing specifying your intentions, the way that I would write is with a semi-colon at the end of every T-SQL statement, and an extra semi-colon at the beginning of any CTE just to be safe.

Editor | Results | Messages

	id	department	parent	archived
1	1	Camping	NULL	0
2	2	Cycle	NULL	0
3	3	Snowsports	NULL	0
4	4	Fitness	NULL	0
5	5	Tents	1	0
6	6	Backpacks	1	0
7	7	Sleeping Bags	1	0
8	8	Cooking	1	0
9	9	Bikes	2	0

	id	department	parent
1	1	Camping	NULL
2	2	Cycle	NULL
3	3	Snowsports	NULL
4	4	Fitness	NULL
5	5	Tents	1
6	6	Backpacks	1
7	7	Sleeping ...	1
8	8	Cooking	1
9	9	Bikes	2

How you decide to do it is up to your own coding standards, and as long as it is clearly defined and supports clean code it is just fine.

Related Links:

- My Book on Common Table Expressions

- cte_demo sample database

Common Table Expressions Book

If you enjoyed this posting, and want to learn more about common table expressions, please take a look at my book on CTE's at Amazon.com. The book is titled Common Table Expressions - Joes 2 Pros® - A CTE Tutorial on Performance, Stored Procedures, Recursion, Nesting and the use of Multiple CTEs.