Nested CTEs

Day 11 of Common Table Expression Month (June) at SteveStedman.com, today we will be taking a look at how to use multiple nested CTEs in a query.

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.

Russian Nesting Dolls

If you have ever seen the Matryoshka dolls known as the Russian nesting dolls or babushka dolls they are very interesting. They start with a large wooden doll that when opened contains a slightly smaller doll, then inside of that one an even smaller doll. These dolls often have five to eight dolls nested inside of each other.

The interesting thing about these dolls is that they have one specific way that they can all fit together; the smallest doll must be the first one put inside of the next smallest. If you were to lose one of the medium dolls, they could still be placed together; there would just be some extra space inside.

Nested CTEs often remind me of the Russian nesting dolls. Think of the first CTE in a multiple CTE query as the smallest doll. That first CTE can’t reference any other CTEs, much like the smallest doll can’t fit any other dolls inside of it.

Think of the second CTE in a multiple CTE query as the second smallest doll in the Russian dolls.

It can only have one doll fit inside of it. CTEs are similar in that the only CTE that can be referenced by the second CTE in a multiple CTE query is the first CTE.

The third CTE is like the third smallest Russian doll. The third smallest doll can only have the smallest or second smallest doll placed inside of it. With the third CTE it can only access the first or the second CTE declared, and not any of the CTEs declared later.

The same concept applies for all of the nested CTEs in a nested CTE query. Any CTE query can access the CTEs declared prior to that one, but not the ones after.

Nested CTEs

Take an example of a multiple CTE query with 4 nested CTEs as follows with the … being replaced with a T-SQL query.

Nested CTEs Sample Code

At this point the CTEs would be considered nested since one CTE references the other.

Now we will add another CTE to get the total number of items per invoice. This CTE will not be nested, as it will only be accessed from the outside query, and not from inside of any CTE, and it is does not reference any of the other CTEs.

To walk through what is happening with the example shown in Figure 6.8 follow these steps:

1. The outer SELECT statement is accessing OrderItemsPerCustomerCTE and CustomersOrdersWithNamesCTE.
2. CustomersOrdersWithNamesCTE references a table and it also references CustomersWithMostOrdersCTE which is considered nested.
3. CustomersWithMostOrdersCTE doesn’t reference any CTEs, it only references tables.
4. OrderItemsPerCustomerCTE doesn’t reference any CTEs, it also only references tables.

In this result we have ended up with 3 CTEs, two nested, and one not nested. If the need arises, we can have nested CTEs that are many levels deep.

Nesting CTEs can be a great way to accomplish queries that would otherwise have been difficult to create.

Related Links:

- My Book on Common Table Expressions

- cte_demo sample database

Nested CTEs in the 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.