Multiple CTEs in a Query

Day 10 of Common Table Expression Month (June) at SteveStedman.com, today we will be taking a look at how to use multiple CTEs in a query.

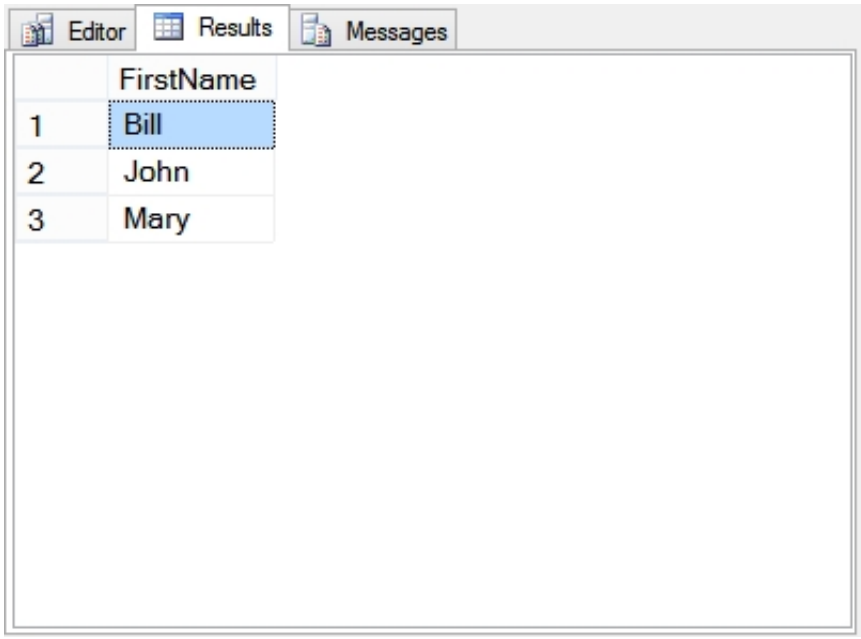
These queries will be using that database that was set up in a previous posting on the CTE_DEMO Sample Database, if you haven't set up the sample database, download it and set it up now.

One of the questions that comes up when discussing CTE Scope is can I do multiple CTEs. The answer is yes you can do multiple CTEs but the scope is is still constrained to a single SQL Statement.

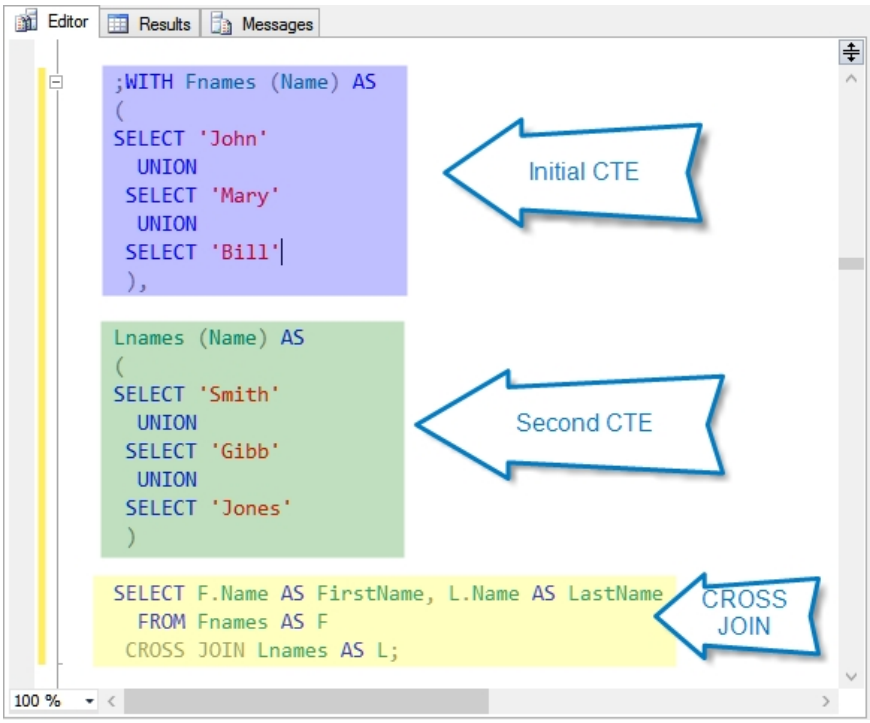
Multiple CTEs in a Single Statement

Lets take a look when multiple CTEs in a single statement might be useful. Take the example of trying to generate random names in order to fill up tables in a test database to simulate a full database during the development phase of a new project. This example is explained in much further detail in chapter 6 in my Book on Common Table Expressions. Lets take a look here at this example from my SQL Saturday presentation. The following is a CTE that returns a list of first names.

When run we see the following:



Now if we want to expand on that we can add a second CTE for last names and CROSS JOIN the results of the two CTEs. A refresher on CROSS JOIN, the CROSS JOIN takes every row from one side and matches it up with every rows for the other side, with no ON clause. This effectively creates a Cartesian product of the two tables, or joins up every possible combination.



Here you will notice that the initial CTE is separated from the new CTE with a comma, and that the second CTE doesn't start with the WITH keyword. Then in the final SELECT statement it just references both of the CTE statements declared above. When run we see the following results.

Editor | Results | Messages

	FirstName	LastName
1	Bill	Gibb
2	John	Gibb
3	Mary	Gibb
4	Bill	Jones
5	John	Jones
6	Mary	Jones
7	Bill	Smith
8	John	Smith
9	Mary	Smith

Here you can see that every first name is joined up with every last name, with two input tables of 3 rows each the result set has 3x3 or 9 rows of output.

You could also add a third CTE with middle names, and possibly a 4th with login names, and very quickly end up with a very large result set multiplying each time there is another cross join added.

Related Links:

- My Book on Common Table Expressions

- cte_demo sample database
- CTE Scope

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.