Keeping it simple and how to do multiple CTE in a query

Ask Question

▲

121

▼

★

19

I have this simple T-SQL query, it emits a bunch of columns from a table and also joins information from other *related* tables.

My data model is simple. I have a scheduled event, with participants. I need to know how many participants participate in each event.

My solution to this is to add a CTE that groups scheduled events and counts the number of participants.

This will allow me to join in that information per scheduled event. Keeping the query simple.

I like to keep my queries simple, however, If I ever in the future need to have additonal temporary results accessible during my simple query, what do I do?

I would really like it, if I could have multiple CTEs but I can't, right? What are my options here?

I've ruled out views and doing things at the application data layer. I prefer to isolated my SQL queries.

sql-server   tsql   common-table-expression

2 Answers

▲

234

▼

✔

You can have multiple `CTE`s in one query, as well as reuse a `CTE`:

```
WITH    cte1 AS
        (
        SELECT  1 AS id
        ),
        cte2 AS
        (
        SELECT  2 AS id
        )
SELECT  *
FROM    cte1
UNION ALL
SELECT  *
FROM    cte2
UNION ALL
SELECT  *
FROM    cte1
```

Note, however, that `SQL Server` may reevaluate the `CTE` each time it is accessed, so if you are using values like `RAND()`, `NEWID()` etc., they may change between the `CTE` calls.
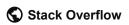
edited Mar 31 '16 at 11:57

Appulus
12.3k   9   31   41

answered Jan 26 '10 at 16:24

Quassnoi
339k   72   534   565

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

(Transact-SQL). You can see this are in the syntax section (take special note of the `[ ,...n ]` in `[ WITH <common_table_expression> [ ,...n ] ]`. Example C, "Using multiple CTE definitions in a single query," calls this out explicitly. Sadly, this example is not provided in the documentation for SQL 2008 and older (i.e., the example wasn't provided when the OP posted the question). – Brian Jul 10 '14 at 15:01 ✎
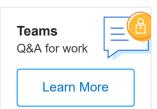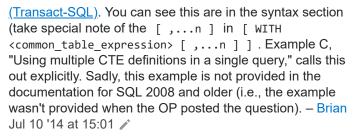
I get double the amount of records on this :/ – Tom Stickel Apr 8 at 20:18

@TomStickel try only using the half of the query, before the last `UNION ALL` – Quassnoi Apr 9 at 2:03 ✎

@Quassnoi Yes that worked. I has done that after writing the comment. Not sure why that 2nd union is even there... – Tom Stickel Apr 9 at 16:19

▲

72

▼

You certainly are able to have multiple CTE's in a single query expression. You just need to separate them with a comma. Here is an example. In the example below, there are two CTE's. One is named `CategoryAndNumberOfProducts` and the second is named `ProductsOverTenDollars`.

```sql
WITH CategoryAndNumberOfProducts (CategoryID, CategoryName
(
    SELECT
        CategoryID,
        CategoryName,
        (SELECT COUNT(1) FROM Products p
         WHERE p.CategoryID = c.CategoryID) as NumberOfProdu
    FROM Categories c
),
```

```
        UnitPrice
    FROM Products p
    WHERE UnitPrice > 10.0
)

SELECT c.CategoryName, c.NumberOfProducts,
       p.ProductName, p.UnitPrice
FROM ProductsOverTenDollars p
    INNER JOIN CategoryAndNumberOfProducts c ON
        p.CategoryID = c.CategoryID
ORDER BY ProductName
```

edited Mar 31 '14 at 4:23

Subin Jacob
2,929   5   27   57

answered Jan 26 '10 at 16:26

Randy Minder
30.6k   37   150   267

---

4   @JohnLeidegren: posting a correct answer within 2 minutes of
     the first correct answer merits an upvote, which I've given, at
     least. – Peter Majeed Mar 20 '12 at 15:52