Advanced Querying Techniques in SQL Server

Navigating Data Hierarchies using Recursive Queries

Gerald Britton
Pluralsight Author

@GeraldBritton www.linkedin.com/in/geraldbritton

Version Check

This course was created by using:

- SQL Server 2022 in a Docker container
- Azure Data Studio version 1.44

Version Check

This course is 100% applicable to:

- SQL Server 2014 and above

Not Applicable

This course is NOT applicable to:

- SQL Server release 2012 and below

Overview

- Review the WITH statement
- Idiosyncrasies in T-SQL
- Using WITH iteratively
- Looking at recursion in math and T-SQL
- Simple examples of recursive T-SQL
- Navigating a hierarchy
- Performance issues

```
[;]WITH
     -- First CTE
     cte1 [(col1, col2, …)]
     AS (
          SELECT …),

     -- Second CTE
     cte2 [(col1, col2, …)]
     AS (
          SELECT …),

     -- Other CTEs

-- Outer query using the CTEs
SELECT | INSERT | UPDATE | DELETE
```

- Keyword WITH starts things off
- Common Table Expression (CTE)
- Optional list of column names
- Query definition

- Optional second common table expression

- Additional CTEs, as required

- Final query, using the common table expressions

Common Table Expressions vs. Subqueries

WITH statement and CTEs

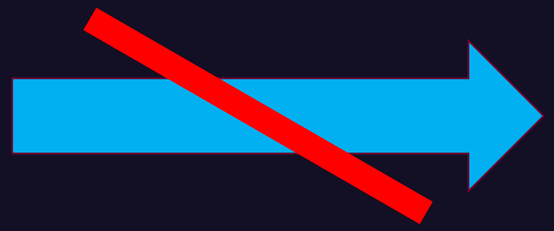
```
[;]WITH
      -- First CTE
      cte1 [(col1, col2, …)]
      AS (
            SELECT …),

      -- Second CTE
      cte2 [(col1, col2, …)]
      AS (
            SELECT …),

      -- Other CTEs

-- Outer query using the CTEs
SELECT …
```

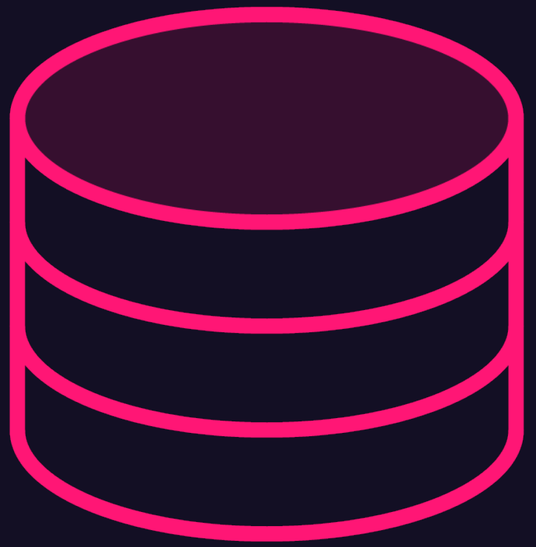
SELECT using subqueries

```
-- Outer query using subqueries
SELECT …
FROM
    -- First subquery
    (SELECT …) AS sub1 [(col1, col2, …)]

    JOIN | UNION

    -- Second subquery
    (SELECT …) AS sub2 [(col1, col2, …)]
```

CTEs Are Not Materialized

```
[;]WITH
      -- First CTE
      cte1 [(col1, col2, …)]
      AS (
            SELECT …),
      -- Second CTE
      cte2 [(col1, col2, …)]
      AS (
            SELECT …),

      -- Other CTEs
-- Outer query using the CTEs
SELECT | INSERT | UPDATE | DELETE
```





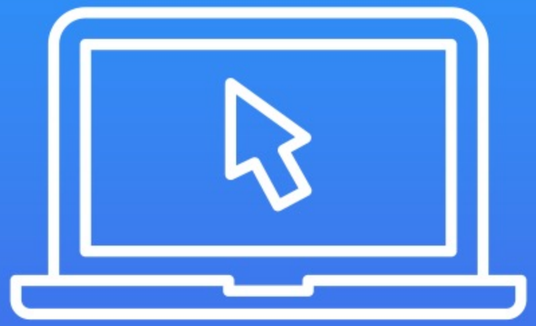
Common Table Expressions Queries

Any valid SELECT statement

Simple or complex

Cannot be nested

Demo 1



Iterative (non-recursive) `WITH` statements

Foundations of Recursion

Mathematical foundation

E.W. Dijkstra – *Recursive Programming*

- http://oai.cwi.nl/oai/asset/9253/9253A.pdf
- *“Go To Statement Considered Harmful.”*

$$F_n = \begin{cases} 0 & if\ n = 0 \\ 1 & if\ n = 1 \\ F_{n-1} + F_{n-2} & if\ n > 1 \end{cases}$$

Recurrence

$$F_{10} = F_9 + F_8 = F_8 + F_7 + F_7 + F_6 = \ \ldots$$

$$S_n = \begin{cases} 0 & if\ n = 0 \\ n + S_{n-1} & if\ n > 0 \end{cases}$$

```
[;]WITH
    -- possible Other CTEs

    -- Recursive CTE

    recurs [(col1, col2, …)]
    AS (
        SELECT …
        UNION ALL
        SELECT …
        FROM recurs
        )
),

-- Outer query using the recursive

SELECT | INSERT | UPDATE | DELETE
FROM recurs
```

- Keyword `WITH` starts things off
- Possible other CTEs

- Recursive CTE

- Query definition
- Base case
- Recursive case

- Final query, using the common table expressions

Recursive Execution Process

Split the CTE expression into anchor and recursive members

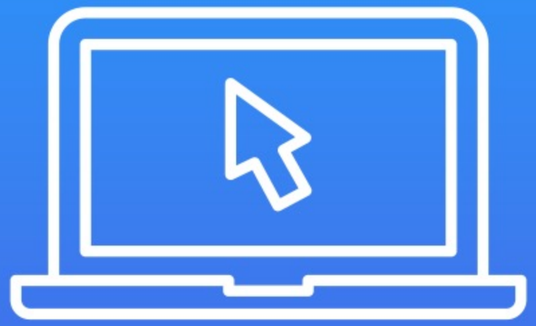
Run the anchor member(s) creating the first invocation or base result set

Run the recursive member(s) with the current result set as input and generating a new result set as output

Repeat until an empty set is returned, when a terminating condition is met

Return the UNION ALL of all the result sets returned

Demo 2

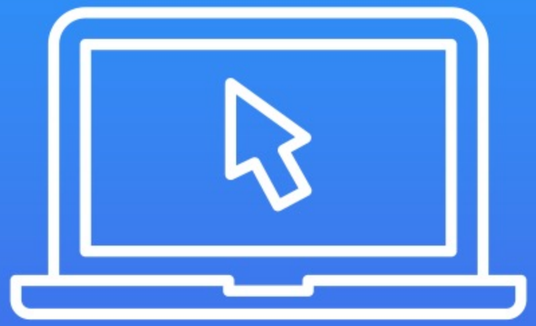


Writing recursive SQL

Recursive CTE Restrictions

- SELECT DISTINCT
- GROUP BY, HAVING
- Scalar aggregation (SUM, AVG, MIN, MAX etc.)
- PIVOT (SQL Server 2012 or later)
- TOP
- LEFT, RIGHT, OUTER JOIN (INNER JOIN is allowed)
- Subqueries

Demo 3



Hierarchy navigation

- AdventureWorks
- Building a bicycle
- Show the parts hierarchy

Summary

`WITH` statement

- Non-recursive
- Recursive

Limitations of recursive queries

- Only current rowset in recursive call

Recursive query for hierarchy navigation

- Powerful, succinct expression

Performance gotchas

- Sometimes row by row
- Custom-designed instead
- SQL Server maximum recursion (32,767)