**Common Table Expressions**

* Introduced in SQL server 2005.
* A temporary result set, that can be referenced within a SELECT, INSERT, UPDATE, or DELETE statement, that immediately follows the CTE.
* You can also use a CTE in a CREATE VIEW statement, as part of the view’s SELECT query.
* Called as a temporary table but difference is that it does not capture any space in the metadata(server instance, the databases on that instance, and the objects within the databases)
* Not stored anywhere and can be referred multiple times after declaration but within the same query.
* Used within SQL Server to simplify complex joins and subqueries, and to provide a means to query hierarchical data such as an organizational chart.

### Advantages of using Common Table Expression

* After declaration of table you can refer the table created multiple times in scope of the same query.
* Can be used in place of views or temporary tables.
* Recursive query is easily created through CTE.
* Promote readability. Rather than lump all you query logic into one large query, create several CTE’s, which are the combined later in the statement.

**Syntax of CTE**

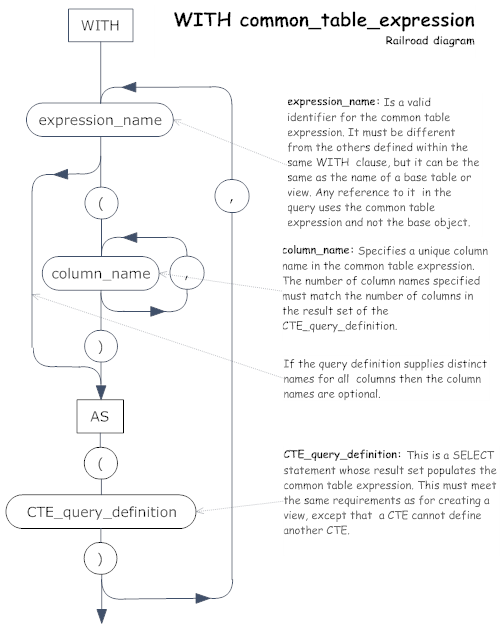
WITH CommonTableExpression\_Name [Column1,Column2,…. ]

AS

<CTE Query>

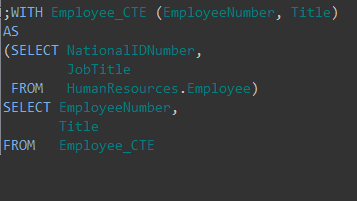
SELECT <Column1, Column2,>

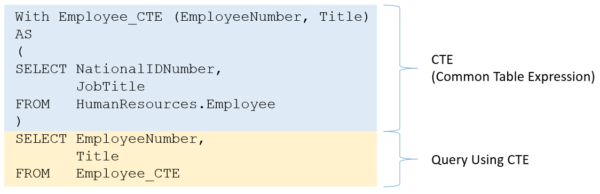
FROM CommonTableExpression\_Name

****

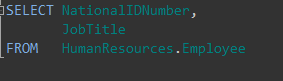
* If you include more than one CTE in your WITH clause, you must separate them with commas.
* For each CTE, you must provide a name, the AS keyword, and a SELECT statement.
* You can also provide column names (separated by commas), as long as the number of names match the number of columns returned by the result set.
* After you define your WITH clause with the necessary CTEs, you can then reference those CTEs as you would any other table.
* However, you can reference a CTE only within the execution scope of the statement that immediately follows the WITH clause.
* After you’ve run your statement, the CTE result set is not available to other statements.

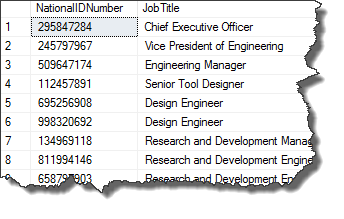
**Example**



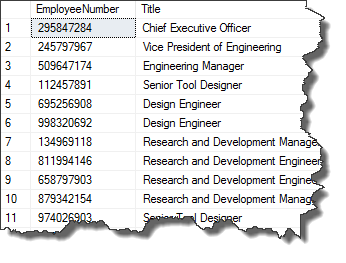


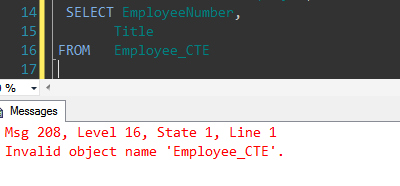
The blue portion is the CTE.  Notice it contains a query that can be run on its own in SQL.  This is called the CTE query definition:





* When we define the CTE we give the result a name as well its columns.
* In this way a CTE acts like a VIEW.
* The result and columns are named differently.
* This allows you to encapsulate complicated query logic with the common table expression.
* Now going back to the CTE, notice that the WITH statement.  There you’ll see the name and columns are defined.
* These columns correspond to the columns returned from the inner query. As such, the results from the CTE are:





* Can define more than one CTE within a WITH statement.
* Helps to simplify some very complicated queries which are ultimately joined together.
* Each complicated piece can include in their own CTE which is then referred to and joined outside the WITH clause.

Here is an example using of TWO CTE’s, it’s a simple example, but it shows how two CTE’s are defined, and then used in an INNER JOIN

