

SQL

SIMPLIFIED FOR ALL

COMMON TABLE EXPRESSIONS (CTE)

Created by :

Abhinav KR

Source:

we3school
stackoverflow
snowflake

SQL

SIMPLIFIED FOR ALL

Common Table Expression (CTE)

what is it?

- ☞ A temporary view existing only for the duration of a single SQL statement.
- ☞ Similar to a view but a CTE is embedded into the query.

Example of a CTE statement:

```
☞ WITH demo AS
(
  SELECT field_1 AS txt,
  field_2 AS txt_2,
  COUNT(*) AS frequency
  FROM table
  WHERE condition = 'TRUE'
  GROUP BY 1,2
)
```

```
SELECT txt,
       AVG(frequency)
FROM demo
GROUP BY txt
ORDER BY 2 DESC;
```

Why should I use CTE?

- ☞ Improves readability.
- ☞ Can be referenced multiple times in a query.
- ☞ Can be used to create recursive queries.
- ☞ Can be used in place of a view in scenarios when the metadata definition need not be stored.
- ☞ can be used in finding duplicates in our data

Are CTE & Sub-query the same thing?

- ☞ A CTE can be used recursively but a sub-query cannot.
- ☞ A subquery can be used in the WHERE clause but a CTE can't.
- ☞ CTEs can be used multiple times in a query.
- ☞ CTEs Must Be Named but this is not always mandatory for subqueries

SQL

SIMPLIFIED FOR ALL

Common Table Expression (CTE)

How to write a CTE?

👉 Let's start with the inner query first:

```
SELECT field_1 AS txt,  
       field_2 AS txt_2,  
       COUNT(*) AS frequency  
FROM table  
WHERE condition = 'TRUE'  
GROUP BY 1,2
```

👉 Now add a 'WITH' syntax & add an alias to the whole inner query and make sure to enclose the whole inner query with paranthesis:

```
WITH demo AS  
(  
  SELECT field_1 AS txt,  
         field_2 AS txt_2,  
         COUNT(*) AS frequency  
  FROM table  
  WHERE condition = 'TRUE'  
  GROUP BY 1,2  
)
```

👉 At this stage, we can use the above query as a temporary table to pull the required columns and we are done with it:

```
WITH demo AS  
(  
  SELECT field_1 AS txt,  
         field_2 AS txt_2,  
         COUNT(*) AS frequency  
  FROM table  
  WHERE condition = 'TRUE'  
  GROUP BY 1,2  
)  
SELECT txt,  
       AVG(frequency)  
FROM demo  
GROUP BY txt  
ORDER BY 2 DESC;
```

