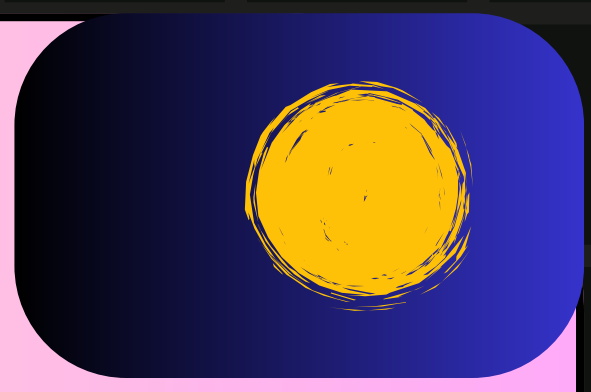
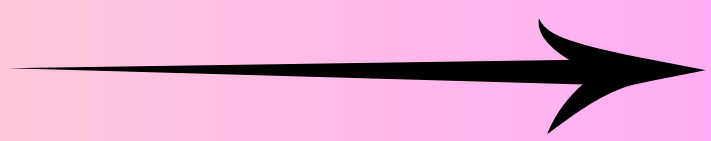


Exploring the Power of CTEs in SQL!

@Vaishnavi Dauale



1

What is a CTE ?

- A CTE (Common Table Expression) is a one-time result set that exists only for the duration of a single query. It was first introduced with SQL Server in 2005 and is widely used because of its temporary nature—CTEs are not stored anywhere permanently.

2

- CTEs allow us to refer to data within a single SELECT, INSERT, UPDATE, DELETE, CREATE VIEW, or MERGE statement execution.

- Database administrators often prefer CTEs as an alternative to subqueries or views for their simplicity and efficiency.

3

🔹Why Use CTEs ?

- Like database views and derived tables, CTEs can make it easier to write and manage complex queries by making them more readable and simple. We can accomplish this characteristic by breaking down the complex queries into simple blocks that can reuse in rewriting the query.

4

Syntax of CTE:

```sql
WITH cte_name (column1, column2, ...) AS (
    SELECT ...
    FROM ...
    WHERE ...
)
SELECT column1, column2, ...
FROM cte_name
WHERE ...
```

5

🔹Types of CTEs :

1. Recursive CTE :

- Used for hierarchical data or recursive queries, such as organizational charts.

6

```sql
WITH RECURSIVE EmployeeHierarchy AS (
    SELECT employee_id, manager_id, first_name, last_name, 1 AS level
    FROM employees
    WHERE manager_id IS NULL

    UNION ALL

    SELECT e.employee_id, e.manager_id, e.first_name, e.last_name, eh.level
    FROM employees e
    INNER JOIN EmployeeHierarchy eh ON e.manager_id = eh.employee_id
)
SELECT employee_id, first_name, last_name, level
FROM EmployeeHierarchy;
```

7

2 Non-Recursive CTE:

- Used for simplifying complex queries without recursion.

```
WITH HighSalaryEmployees AS (
    SELECT e.employee_id, e.first_name, e.last_name, e.salary, d.department_name
    FROM employees e
    JOIN departments d ON e.department_id = d.department_id
    WHERE e.salary > 60000
)
SELECT employee_id, first_name, last_name, salary, department_name
FROM HighSalaryEmployees;
```

8

💠Advantages of CTEs :

- CTE facilitates code maintenance easier.
- CTE increases the readability of the code.
- It increases the performance of the query.
- CTE makes it possible to implement recursive queries easily.

91

🔹Disdvantages of CTEs :

- CTE members are unable to use the keyword clauses like Distinct, Group By, Having, Top, Joins, etc.
- The CTE can only be referenced once by the Recursive member.
- We cannot use the table variables and CTEs as parameters in stored procedures.
- We already know that the CTE could be used in place of a view, but a CTE cannot be nested, while Views can.

Note 🎯

It should keep in mind while writing the CTE query definition; we cannot use the following clauses:

1. **ORDER BY unless you also use as TOP clause**
2. **INTO**
3. **OPTION clause with query hints**
4. **FOR BROWSE**

11

Thank You

@Vaishnavi Dauale