CTE – Common Table Expression

It is a temporary, named result set in SQL

Its created using WITH keyword.

Key Points about CTEs:

1. Declared at the start of a query using WITH.
2. Makes complex queries easier to read (step by step).
3. Can be **referenced multiple times** in the same query.

**Syntax**

WITH cte\_name AS (

-- your query here

SELECT column1, column2, ...

FROM table\_name

WHERE condition

)

-- now use the CTE like a table

SELECT \*

FROM cte\_name;

**Example**

Let’s say we have an Employees table:

| **id** | **name** | **department** | **salary** |
| --- | --- | --- | --- |
| 1 | Aditi | HR | 50000 |
| 2 | Raj | HR | 60000 |
| 3 | Meena | IT | 70000 |

Now, suppose we want only the HR employees and then find the average salary:

WITH hr\_employees AS (

SELECT name, salary

FROM Employees

WHERE department = 'HR'

)

SELECT AVG(salary) AS avg\_hr\_salary

FROM hr\_employees;

**RECURSIVE CTE** – A recursive CTE is a CTE that refers to itself.

It is used when you need to work with hierarchies like org charts, folder structures, famil trees etc.

Or to generate sequence of numbers.

**Syntax**

WITH RECURSIVE cte\_name AS (

-- 1. Anchor part (starting row(s))

SELECT ...

UNION ALL

-- 2. Recursive part (cte refers to itself)

SELECT ...

FROM cte\_name

JOIN ...

)

SELECT \* FROM cte\_name;

**Anchor query** → the starting point (like the root of a tree, or first number/row).

**Recursive query** → keeps running, each time using the previous result.

**UNION ALL** → combines results until no more rows are returned.