advance
CTE   # with clause

WITH clause / CTE (common table expressions)

- It works like one time variables in other languages like CPP, python etc.
- It runs a query and store its ~~ka~~ result in RAM until the occution of that particular query.
- returns the result set.
- increase readiblity and speed.
- Time span : current Query Execution only.
- very helpfull in recursive queries.

syntax:

```
WITH CTE_NAME AS (
        -- query
) main query.
```

```
Eg:  WITH  employeedepartments  AS (
            SELECT  e.id, e.name
            FROM  employee  AS e
            JOIN  department  d  ON  e.id = d.id
      )
      SELECT  *  FROM  employeedepartments;
```

- CTE are also used to write recursive queries.

advance

recursion.

Recursive Queries

- Syntax :

```
WITH RECURSIVE cte_name AS (
        -- Base/Anchor member.
       [
       [ Query
        UNION OR UNION ALL
        -- Recursive member.
       [
       [ Query ( inside FROM there must be
                 cte_name as it reference
                 same result set)
)
  main query.
```

- Anchor member
  ↳ provide base result and starting point

- Recursive member
  ↳ references CTE itself & iteratively adds rows to result set.
  ↳ It also have termination condition, or query will end when no result is found.

recursive query

Eg:

```
WITH RECURSIVE eh AS (
    -- Anchor
    SELECT id, name, mid, 1 AS Level
    FROM employees
    WHERE (manager) mid IS NULL

    UNION ALL

    -- Recursive
    SELECT e.id, e.name, e.mid, eh.level+1
    FROM employees e
    JOIN eh AS eh ON e.mid = eh.id
)
SELECT id, name, mid, level
FROM eh;
```

It is same like other recursion concept of other languages.