CTE - Common Table Expression –

Temporary, named result set, that can be used multiple times within your query to simplify and organise complex query.

Why CTE?

Efficiency

Readability

Modularity

Reusability

Standalone CTE – defined and used independently

CTE Syntax –

WITH CTE\_Name AS

( SELECT…. FROM…. WHERE….)

SELECT…. FROM CTE\_Name WHERE….

Example –

WITH CTE\_TotalSales AS

(

SELECT userid, SUM(product\_id) AS TotalSales

FROM sales

GROUP BY userid

)

--Main Query

SELECT u.userid, u.signup\_date, cts.TotalSales

FROM users u

LEFT JOIN CTE\_TotalSales cts

ON cts.userid = u.userid

Multiple CTE –

WITH CTE\_Name1 AS

(SELECT…. FROM…. WHERE….)

, CTE\_Name2 AS

(SELECT…. FROM…. WHERE….)

Example –

WITH CTE\_TotalSales AS

(

SELECT userid, SUM(product\_id) AS TotalSales

FROM sales

GROUP BY userid

)

, CTE\_LastOrder AS

(

SELECT userid, MAX(created\_date) LastOrder

FROM sales

GROUP BY userid

)

--Main Query

SELECT u.userid, u.signup\_date, cts.TotalSales, cts1.LastOrder

FROM users u

LEFT JOIN CTE\_TotalSales cts

ON cts.userid = u.userid

LEFT JOIN CTE\_LastOrder cts1

ON cts.userid = u.userid

Nested CTE –

WITH CTE\_Name1 AS

( SELECT…. FROM…. WHERE….)

, CTE\_Name2 AS

(SELECT…. FROM CTE\_Name1 WHERE….)

Example –

WITH CTE\_TotalSales AS

(

SELECT userid, SUM(product\_id) AS TotalSales

FROM sales

GROUP BY userid

)

--Last Order

, CTE\_LastOrder AS

(

SELECT userid, MAX(created\_date) LastOrder

FROM sales

GROUP BY userid

)

--Rank Customer

, CTE\_CustRank AS

(

SELECT userid, TotalSales,

RANK() OVER(ORDER BY TotalSales DESC) AS CustRank

FROM CTE\_TotalSales

)

--Segment the customer

, CTE\_CustSeg AS

(

SELECT userid,

CASE WHEN TotalSales > 12 THEN 'High'

WHEN TotalSales > 9 THEN 'Medium'

ELSE 'Low'

END CustomerSegments

FROM CTE\_TotalSales

)

--Main Query

SELECT u.userid, u.signup\_date, cts.TotalSales, cts1.LastOrder, cts2.CustRank, cts3.CustomerSegments

FROM users u

LEFT JOIN CTE\_TotalSales cts

ON cts.userid = u.userid

LEFT JOIN CTE\_LastOrder cts1

ON cts1.userid = u.userid

LEFT JOIN CTE\_CustRank cts2

ON cts2.userid = u.userid

LEFT JOIN CTE\_CustSeg cts3

ON cts3.userid = u.userid

Non-recursive CTE –

Syntax –

WITH CTE\_Name AS

(SELECT…. FROM…. WHERE….

UNION ALL/UNION

SELECT…. FROM CTE\_Name WHERE [BREAK CONDITION])

Example –

--Sequence of number from 1 to 20

WITH CTE\_Series AS

(

--Anchor

SELECT 1 AS MyNum

UNION ALL

--Recursive

SELECT MyNum+1

FROM CTE\_Series

WHERE MyNum < 1000

)

--Main Query

SELECT MyNum FROM CTE\_Series

OPTION (MAXRECURSION 1000)

Example –

WITH CTE\_Employee AS

(

--Anchor Query

SELECT Brand, Location, Idea, 1 AS Level FROM Shark

WHERE Location = 'Delhi'

UNION ALL

--Recursive

SELECT S.Brand, S.Location, S.Idea, Level+1

FROM Shark S

INNER JOIN CTE\_Employee cts4

ON S.Location = cts4.Location

)

--Main Query

SELECT \* FROM CTE\_Employee

OPTION (MAXRECURSION 12000)