CASE Statement

Evaluates a list of conditions and returns a value when the first condition is met.

Syntax –

CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

…

ELSE result

END

Use case –

Categorising Data –

Group up the data into different categories based on certain conditions.

Example –

SELECT

Category,

SUM(total\_price) Total

FROM(

SELECT

order\_id,

total\_price,

CASE

WHEN total\_price > 20 THEN 'HIGH'

WHEN total\_price > 10 THEN 'MEDIUM'

ELSE 'LOW'

END Category

FROM pizza\_sales

)t

GROUP BY Category

ORDER BY Total DESC

Data type of the result must be matching.

Mapping Values –

Transform data from one form to another.

SELECT

EmployeeID,

FirstName,

LastName,

Gender,

CASE

WHEN Gender = 'F' THEN 'Female'

WHEN Gender = 'M' THEN 'Male'

ELSE 'Not Available'

END GenderFullText

FROM Sales

Handling NULLs –

SELECT

CustomerID,

Scores,

CASE

WHEN Score IS NULL THEN 0

ELSE Score

END ScoreClean,

AVG(Scores) OVER() AvgCust

FROM Sales

Conditional Aggregation –

SELECT

OrderID,

CustomerID,

Sales,

CASE

WHEN Sales > 30 THEN 1

ELSE 0

END SalesFlag

FROM Sales

ORDER BY CustomerID

SELECT

CustomerID,

SUM(CASE

WHEN Sales > 30 THEN 1

ELSE 0

END) TotalOrders

FROM Sales

GROUP BY CustomerID