

CASE EXPRESSION

The CASE expression goes through conditions and returns a value when the first condition is met (like if-then-else statement). If no conditions are true, it returns the value in the ELSE clause.

If there is no ELSE part and no conditions are true, it returns NULL.

Also called CASE STATEMENT.

CASE Statement Syntax

General CASE Syntax

```
CASE  
  WHEN condition1 THEN result1  
  WHEN condition2 THEN result2  
  WHEN conditionN THEN resultN  
  ELSE other_result
```

END;

Example 1)

```
SELECT customer_id, amount,
```

```
  CASE  
    WHEN amount > 100 THEN 'Expensive Product'  
    WHEN amount = 100 THEN 'Moderate Product'  
    ELSE 'Inexpensive product'
```

```
  END AS ProductStatus
```

```
FROM payment
```

Example 2

```
SELECT customer_id,
```

```
  CASE amount
```

```
    WHEN 500 THEN 'Prime Customer'
```

```
    WHEN 100 THEN 'Plus Customer'
```

```
    ELSE 'Regular Customer'
```

```
  END AS CustomerStatus
```

```
FROM payment.
```

COMMON TABLE EXPRESSION (CTE)

- ① A common table expression, or CTE, is a temporary named result set created from a simple SELECT statement that can be used in a subsequent SELECT statement.
- ② We can define CTEs by adding a WITH clause directly before SELECT, INSERT, UPDATE, DELETE, or MERGE statement.
- ③ The WITH clause can include ~~can~~ one or more CTEs separated by commas.

Syntax

```
WITH my-cte AS (  
    SELECT a, b, c  
    FROM Table1  
)  
SELECT a, c  
FROM my-cte.
```

CTE query

Main query

The name of this CTE is my-cte, and the CTE query is SELECT a, b, c FROM Table1. The CTE starts with the WITH keyword, after which you specify the name of your CTE, then the content of the query in parentheses. The main query comes after the closing parenthesis and refers to the CTE. Here, the main query (also known as the output query) is SELECT a, c FROM my-cte.

1) CTE - Example
 WITH my-cte AS (
 SELECT *, AVG (amount) OVER (ORDER BY
 p.customer-id) AS "Average-Price",
 COUNT (address-id) OVER (ORDER BY
 c.customer-id) AS "Count"
 FROM payment as p
 INNER JOIN customer AS c
 ON p.customer-id = c.customer-id
)
 SELECT first_name, last_name
 FROM my-cte.

① Example Multiple CTEs
 WITH my-cte AS (
 SELECT *, AVG (amount) OVER (ORDER BY p.customer-id)
 AS 'Average-Price',
 COUNT (address-id) OVER (ORDER BY c.customer-id)
 AS 'Count'
 FROM payment as p
 INNER JOIN customer AS c
 ON p.customer-id = c.customer-id
),
 my-ca AS (
 SELECT *
 FROM customer as c
 INNER JOIN address AS a
 ON a.address-id = c.address-id
 INNER JOIN country as cc
 ON cc.city-id = a.city-id
)
 SELECT cp.first_name, cp.last_name, ca.city, ca.
 country, cp.amount FROM my-ca as ca, my-cte as
 cp

② Example Advance

WITH my-cte AS (

SELECT mode, MAX(amount) AS highest-price,

SUM(amount) AS total-price

FROM payment

GROUP BY mode

)

SELECT payment.* , my.highest-price, my.total-price

FROM payment

JOIN my-cte my

ON payment.mode = my.mode

ORDER BY payment.mode