ANALYTICS
WITH ANAND
LET THE DATA DO THE TALKING...

CASE STATEMENTS IN SQL

MASTER BLASTER SQL IN SNOWFLAKE

WHAT DOES THE SQL CASE STATEMENT DO?

- The CASE statement allows you to **perform an IF-THEN-ELSE check within an SQL statement**.
- It’s good for displaying a value in the SELECT query based on logic that you have defined. As the data for columns can vary from row to row, using a CASE SQL expression can help make your data more readable and useful to the user or to the application.
- It’s quite common if you’re writing complicated queries or doing any kind of ETL work.

SQL CASE STATEMENT SYNTAX

- The syntax of the SQL CASE expression is:

```
CASE [expression]
        WHEN condition_1 THEN result_1
        WHEN condition_2 THEN result_2 ...
        WHEN condition_n THEN result_n
        ELSE result
END case_name
```

PARAMETERS OF THE CASE STATEMENT

- The parameters or components of the CASE SQL statement are:
- **expression** (optional): This is the expression that the CASE statement looks for. If we're comparing this to an IF statement, this is the check done inside the IF statement (e.g. for IF x > 10, the expression would be "x > 10"
- **condtion_1/condition_n** (mandatory): These values are a result of the expression parameter mentioned. They are the possible values that expression can evaluate to. Alternatively, they can be an expression on their own, as there are two ways to write an SQL CASE statement (as explained below). They also relate to the IF statement in the IF-THEN-ELSE structure.
- **result_1/result_n** (mandatory): These values are the value to display if the related condition is matched. They come after the THEN keyword and relate to the THEN part of the IF-THEN-ELSE structure.
- **result** (optional): This is the value to display if none of the conditions in the CASE statement are true. It is the ELSE part of the IF-THEN-ELSE structure and is not required for the CASE SQL statement to work.
- **case_name** (optional): This value indicates what the column should be referred to as when displayed on the screen or from within a subquery. It's also called the column alias.

EXAMPLES OF THE CASE STATEMENT

```
SELECT first_name, last_name, country,
CASE
    WHEN country = 'USA' THEN 'North  America'
    WHEN country = 'Canada'  THEN 'North America'
    WHEN country = 'UK' THEN 'Europe'
    WHEN country = 'France' THEN 'Europe'
    ELSE 'Unknown'
END Continent
FROM customers
ORDER BY first_name, last_name;
```

The results are:

FIRST_NAME	LAST_NAME	COUNTRY	CONTINENT
Adam	Cooper	USA	North America
John	Smith	USA	North America
Mark	Allan	UK	Europe
Sally	Jones	USA	North America
Steve	Brown	Canada	North America

SEARCHED CASE WITH NUMBERS

```
SELECT first_name, last_name, employees,
    CASE
        WHEN employees < 10 THEN 'Small'
         WHEN employees >= 10 AND employees <= 50 THEN 'Medium'
        WHEN employees >= 50 THEN 'Large'
    END SizeOfCompany
    FROM customers
    ORDER BY first_name, last_name;
```

FIRST_NAME	LAST_NAME	EMPLOYEES	SIZEOFCOMPANY
Adam	Cooper	55	Large
John	Smith	4	Small
Mark	Allan	23	Medium
Sally	Jones	10	Medium
Steve	Brown	15	Medium

CASE STATEMENT WITH IN CLAUSE

```
SELECT first_name, last_name, country,
CASE
  WHEN country IN ('USA', 'Canada') THEN 'North America'
  WHEN country IN ('UK', 'France') THEN 'Europe'
   ELSE 'Unknown'
END Continent
FROM customers
ORDER BY first_name, last_name;
```

FIRST_NAME	LAST_NAME	COUNTRY	CONTINENT
Adam	Cooper	USA	North America
John	Smith	USA	North America
Mark	Allan	UK	Europe
Sally	Jones	USA	North America
Steve	Brown	Canada	North America

WHAT IF NO MATCH IS FOUND IN A CASE STATEMENT?

- If there is no match found in any of the conditions, that's where **the ELSE statement** comes in. The value used in the ELSE statement is what is returned if no match is found.
- However, this is an optional part of the SQL CASE statement. If there is no result, and there is no ELSE statement, then the value of NULL is returned.

DOES THE CASE STATEMENT SEARCH ALL CONDITIONS OR JUST FINDS THE FIRST MATCH?

A common question on SQL CASE statements is if the database evaluates all of the conditions in the CASE statement, or does it stop after finding the first match?
The answer is that it **stops after the first match**. It finds the first match, or the first expression that is evaluated to be a match, and does not continue with the rest.

It also performs something called "**short-circuit evaluation**" for Simple CASE expressions. This might not be a concern to you, but it's good to know for performance reasons. The database will evaluate the first condition, then compare it to the expression, then evaluate the second condition, then evaluate that to the expression, and so on. It doesn't evaluate all conditions before comparing the first one to the expression.

HOW MANY CONDITIONS OR ARGUMENTS CAN A CASE STATEMENT USE?

The maximum number of conditions in a CASE statement is **255**. This includes:

- The initial expression in a simple CASE statement
- The optional ELSE expression
- The expression for the WHEN condition
- The expression for the THEN result

You can use nested CASE statements so that the return value is a CASE expression. However, if you're reaching the limit of 255 expressions, I would be looking at the efficiency of the query itself, as most queries should not need 255 expressions.

BUSINESS USE CASE EXAMPLES

THANK YOU