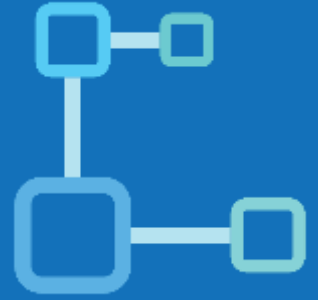netlink
digital solutions

www.netlink.com

DATABASE

PostgreSQL CASE

The PostgreSQL CASE expression is the same as IF/ELSE statement in other programming languages. It allows you to add if-else logic to the query to form a powerful query.

Since CASE is an expression, you can use it in any places where an expression can be used e.g.,<u>SELECT</u>, <u>WHERE</u>, <u>GROUP BY</u>, and <u>HAVING</u> clause.

The CASE expression has two forms:

1) General PostgreSQL CASE expression

2) Simple PostgreSQL CASE expression

PostgreSQL CASE

1) General PostgreSQL CASE expression

The following illustrates the general form of the CASE statement:

```
CASE
    WHEN condition_1  THEN result_1
    WHEN condition_2  THEN result_2
    [WHEN ...]
    [ELSE else_result]
END
```

1. In this syntax, each condition (condition_1, condition_2…) is a boolean expression that returns either true or false.

2. When a condition evaluates to false, the CASE expression evaluates the next condition from the top to bottom until it finds a condition that evaluates to true.

3. If a condition evaluates to true, the CASE expression returns the corresponding result that follows the condition. For example, if the condition_2 evaluates to true, the CASE expression returns the result_2. Also, it immediately stops evaluating the next expression.

4. In case all conditions evaluate to false, the CASE expression returns the result (else_result) that follows the ELSE keyword. If you omit the ELSE clause, the CASE expression returns NULL.

2) Simple PostgreSQL CASE expression

PostgreSQL provides another form of
the CASE expression called simple form as follows:

```
CASE expression
    WHEN value_1 THEN result_1
    WHEN value_2 THEN result_2
    [WHEN ...]
ELSE
  else_result
END
```

1. The CASE first evaluates the expression and compares the result with each value(value_1, value_2, …) in the WHEN clauses sequentially until it finds the match.

2. Once the result of the expression equals a value (value1, value2, etc.) in a WHEN clause, the CASE returns the corresponding result in the THEN clause.

3. If CASE does not find any matches, it returns the else_result in that follows the ELSE, or NULL value if the ELSE is not available.