CASE STATEMENTS IN SQL - ‘if-then-else’ & NULL FUNCTION

The `CASE` expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. 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.

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

Example:

- The following SQL goes through conditions and returns a value when the first condition is met:

```
SELECT column_name(s),
CASE
    WHEN Quantity > 30 THEN 'The quantity is greater than 30'
    WHEN Quantity = 30 THEN 'The quantity is 30'
    ELSE 'The quantity is under 30'
END AS QuantityText
FROM OrderDetails;
```

- Here i have 6 values Phone, Web, Fax, Refferal, Email, Postal Mail, i have to write a query which indicate Phone & web as inbound, Fax as Electronic, Refferal, Email, Postal Mail as Outbound in a new column.

```
-- This is a bad code, because it will be difficult for other coder
-- to understand.
SELECT *,
CASE
    WHEN SUBMITTED_VIA LIKE '%l' THEN 'Outbound'
    WHEN SUBMITTED_VIA LIKE '%x' THEN 'Electronic'
    ELSE 'Inbound'
END AS SUBMISSIO_TYPE
FROM CONSUMER_COMPLAINTS;
```

```
-- Optimal Code

SELECT *,
  CASE
      WHEN SUBMITTED_VIA IN ('Referral', 'Postal Mail', 'Email') THEN 'Outbound'
      WHEN SUBMITTED_VIA IN ('Phone', 'Web') THEN 'Inbound'
      ELSE 'Electronic'
  END AS SUBMISSIO_TYPE
FROM CONSUMER_COMPLAINTS;
```

- What if there is `NULL` value in data

```
-- Wrong code
SELECT *,
  CASE
  WHEN COLUMN_NAME IN ('NULL', 'SOMETHING U WANT TO CONVERT') THEN 'NA'
  ELSE COLUMN_NAME
END AS SUB_COLUMN_NAME
FROM TABLE_NAME;

-- In this case if there are values like NULL & something u want
-- then it will convert it into value NA
-- else it will return same data in that new column


-- Rigth code
SELECT *,
  CASE
  WHEN COLUMN_NAME = 'SOMETHING U WANT TO CONVERT' OR COLUMN_NAME IS NULL THEN 'NA'
  ELSE COLUMN_NAME
END AS SUB_COLUMN_NAME
FROM TABLE_NAME;
```

- If there is no column name in that table, and that column by doing some functionality on 2-3 columns then how to apply if-then-esle statement on that

new column

```
-- Example Doing Datediff on 2 columns and name it as process_day, then applied
-- case statement. Then casestatement can read easily.
SELECT DATEDIFF('DAY', ORDER_DATE, SHIP_DATE) AS PROCESS_DAY,
    CASE
    WHEN PROCESS_DAY <= 3 THEN '5'
    WHEN PROCESS_DAY <= 6 AND PROCESS_DAY > 3 THEN '4'
    WHEN PROCESS_DAY < 10 AND PROCESS_DAY > 6 THEN '3'
    ELSE '2'
    END AS RATING
FROM SALES_DATA;
```

1. NULL

   A field with a NULL value is a field with no value.

   If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.

   Note: A NULL value is different from a zero value or a field that contains spaces. A field with a NULL value is one that has been left blank during record creation!

   It is not possible to test for NULL values with comparison operators, such as =, <, or <>.

   We will have to use the `IS NULL` and `IS NOT NULL` operators instead.

```
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```