SQL | Case Statement

Control statements form the heart of most languages since they control the execution of other sets of statements. These are also found in SQL and should be exploited for uses such as query filtering and query optimization by carefully selecting tuples that match our requirements.

In this article, we explore the Case-Switch statement in SQL. The CASE statement is SQL's way of handling if/then logic.

There can be two valid ways of going about the case-switch statements.

The first takes a variable called case_value and matches it with some statement_list.

Syntax:

CASE case_value

WHEN when_value THEN statement_list

[WHEN when_value THEN statement_list] ...

[ELSE statement_list]

END CASE

The second considers a search_condition instead of variable equality and executes the statement_list accordingly.

Syntax:

CASE

WHEN search_condition THEN statement_list

[WHEN search_condition THEN statement_list] ...

[ELSE statement_list]

END CASE

Example:

CREATE TABLE:

Below is a selection from the “Customer” table in the sample database:

```
CREATE TABLE Customer(
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(50),
    LastName VARCHAR(50),
    Country VARCHAR(50),
    Age int(2),
  Phone int(10)
);
-- Insert some sample data into the Customers table
INSERT INTO Customer (CustomerID, CustomerName, LastName, Country, Age, Phone)
VALUES (1, 'Shubham', 'Thakur', 'India','23','xxxxxxxxxx'),
       (2, 'Aman ', 'Chopra', 'Australia','21','xxxxxxxxxx'),
       (3, 'Naveen', 'Tulasi', 'Sri lanka','24','xxxxxxxxxx'),
       (4, 'Aditya', 'Arpan', 'Austria','21','xxxxxxxxxx'),
       (5, 'Nishant. Salchichas S.A.', 'Jain', 'Spain','22','xxxxxxxxxx');
```

Output:

CustomerID	CustomerName	LastName	Country	Age	Phone
1	Shubham	Thakur	India	23	xxxxxxxxxx
2	Aman	Chopra	Australia	21	xxxxxxxxxx
3	Naveen	Tulasi	Sri lanka	24	xxxxxxxxxx
4	Aditya	Arpan	Austria	21	xxxxxxxxxx
5	Nishant. Salchichas S.A.	Jain	Spain	22	xxxxxxxxxx

Adding Multiple Conditions to a CASE statement

Query:

By adding multiple conditions in SQL

```
SELECT CustomerName, Age,
CASE
    WHEN Age> 22 THEN 'The Age is greater than 20'
    WHEN Age = 21 THEN 'The Age is 21'
    ELSE 'The Age is over 30'
END AS AgeText
FROM Customer ;
```

Output:

CustomerName	Age	QuantityText
Shubham	23	The Age is greater than 20
Aman	21	The Age is 21
Naveen	24	The Age is greater than 20
Aditya	21	The Age is 21
Nishant. Salchichas S.A.	22	The Age is over 30

CASE Statement With ORDER BY Clause

Query:

By using Order by Clause in SQL

```
SELECT CustomerName, Country
FROM Customer
ORDER BY
(CASE
    WHEN Country  IS 'India' THEN Country
    ELSE Age
END);
```

Output:

CustomerName	Country
Aman	Australia
Aditya	Austria
Nishant. Salchichas S.A.	Spain
Naveen	Sri lanka
Shubham	India

Some important points about CASE statements:

1. There should always be a SELECT in the case statement.
2. END. ELSE is an optional component but WHEN THEN these cases must be included in the CASE statement.
3. We can make any conditional statement using any conditional operator (like WHERE) between WHEN and THEN. This includes stringing together multiple conditional statements using AND and OR.
4. We can include multiple WHEN statements and an ELSE statement to counter with unaddressed conditions.

Last Updated : 13 Apr, 2023

29

Similar Reads

1. Difference between Structured Query Language (SQL) and Transact-SQL (T-SQL)
2. Configure SQL Jobs in SQL Server using T-SQL
3. SQL INSERT INTO Statement
4. SQL | DELETE Statement
5. SQL | UPDATE Statement
6. SQL | INSERT IGNORE Statement
7. SQL | DESCRIBE Statement
8. SQL | MERGE Statement
9. MERGE Statement in SQL Explained
10. Delete statement in MS SQL Server

Previous Next

Article Contributed By :

GeeksforGeeks