

CASE statement in SQL Server

This article will show you in detail how to use the CASE statement handling function in SQL Server with specific syntax and examples to better visualize and capture functions.

This article will show you in detail how to use the CASE statement handling function in SQL Server with specific syntax and examples to better visualize and capture functions.

Describe

CASE statement used to set branch conditions in SQL Server, similar to the function of IF-THEN-ELSE statement.

CASE has 2 formats:

1. Simple CASE function, also called **Simple CASE**.
2. The search function CASE is also called **Searched CASE**.

Inside:

1. Simple CASE is to compare an expression with a set of simple expressions to determine the result.
2. Searched CASE is to evaluate a set of Boolean expressions to determine results.
3. Both formats support the ELSE argument (but not required).

Syntax

To use CASE statements in SQL Server, we use the following syntax:

Simple CASE

```
CASE biethuc_dauvao
  WHEN biethuc_1 THEN ketqua_1
  WHEN biethuc_2 THEN ketqua_2
  .
  WHEN biethuc_n THEN ketqua_n
ELSE ketqua_khac
END
```

Or Searched CASE

```

CASE
  WHEN dieukien_1 THEN ketqua_1
  WHEN dieukien_2 THEN ketqua_2
  .
  WHEN dieukien_n THEN ketqua_n
ELSE ketqua_khac
END

```

Parameters :

1. *bieuthuc_dauvao*: The expression will be compared to each value provided.
2. *bieuthuc_1, bieuthuc_2, bieuthuc_n*: expressions will be used to compare in turn with the input expression. When an expression matches *bieuthuc_dauvao*, CASE will execute the next statement and no further comparison.
3. *dieukien_1, dieukien_2, dieukien_n* : conditions reviewed, approved in the order listed. When a condition is determined to be true, CASE returns the result and no further conditions are evaluated. All conditions must be the same data type.
4. *ketqua_1, ketqua_2, ketqua_n* : the result returned after considering the condition is true. All values ?? must be the same data type.

Note :

1. If no *match* is found for TRUE, the CASE statement will return the result in the ELSE clause.
2. If there is no ELSE clause and there is no TRUE condition then the CASE statement will return NULL.
3. Conditions are evaluated in the order listed. When a condition is determined to be true, the CASE statement returns the result and no further conditions are evaluated.
4. CASE can be used in later versions of SQL Server: SQL Server 2017, SQL Server 2016, SQL Server 2014, SQL Server 2012, SQL Server 2008 R2, SQL Server 2008, and SQL Server 2005.

For example

Take a look and explore some examples of CASE statements in SQL Server.

Example 1: Simple CASE

Suppose you need to check the section of the TipsMake.com website, based on the Code column of the conversion table:

1. If Code = '01' is 'Laptrinh-TipsMake.com'.
2. If Code = '02' is 'Congnghe-TipsMake.com'.
3. If Code = '03' is 'Cuocsong-TipsMake.com'.
4. On the contrary, that is not from 1 to 3 then in the section 'Khoahoc-TipsMake.com'.

```

SELECT tenchuyenmuc, Code
  (CASE code
  WHEN 01 THEN 'Laptrinh-TipsMake.com'
  WHEN 02 THEN 'Congnghe-TipsMake.com'
  WHEN 03 THEN 'Cuocsong-TipsMake.com'
  ELSE 'Khoahoc-TipsMake.com'
  END) AS Chuyenmuc

```

```
FROM chuyenmuc
ORDER BY Code
```

Results returned:

tenchuyenmuc Code Chuyenmuc SQL Server 01 Laptrinh-TipsMake.com Linux 02 Congnghe-TipsMake.com Python 01 Laptrinh-TipsMake.com JavaScript 01 Laptrinh-TipsMake.com Android 02 Congnghe-TipsMake.com Giai tri 03 Cuocsong-TipsMake.com Cuocsong-TipsMake.com Science Curve 04 Khoahoc-TipsMake.com Kham pha Science 05 Khoahoc-TipsMake.com

Example 2: Searched CASE

```
SELECT tenchuyenmuc ,
CASE
WHEN code = 01 THEN 'Laptrinh-TipsMake.com'
WHEN code = 02 THEN 'Congnghe-TipsMake.com'
WHEN code = 03 THEN 'Cuocsong-TipsMake.com'
ELSE 'Khoahoc-TipsMake.com'
END
FROM chuyenmuc ;
```

You may also not use ELSE conditions like this:

```
SELECT tenchuyenmuc , Code
(CASE code
WHEN 01 THEN 'Laptrinh-TipsMake.com'
WHEN 02 THEN 'Congnghe-TipsMake.com'
WHEN 03 THEN 'Cuocsong-TipsMake.com'
END) AS Chuyenmuc
FROM chuyenmuc
ORDER BY Code
```

Or:

```
SELECT tenchuyenmuc ,
CASE
WHEN code = 01 THEN 'Laptrinh-TipsMake.com'
WHEN code = 02 THEN 'Congnghe-TipsMake.com'
WHEN code = 03 THEN 'Cuocsong-TipsMake.com'
END
FROM chuyenmuc ;
```

When the ELSE clause is omitted, if no condition is true, the CASE statement returns NULL.

Example 3: Comparing two conditions

Here is an example that illustrates how to use CASE statements to compare different conditions:

```
SELECT
CASE
WHEN code = 1 THEN 'Laptrinh-TipsMake.com'
WHEN code = 2 THEN 'Congnghe-TipsMake.com'
END
FROM chuyenmuc ;
```

Remember that the conditions are compared in turn in the order listed. When a condition is determined to be true, the CASE statement will return the result immediately and no other conditions will be evaluated. So you need to be careful when choosing the order that lists your conditions.

See more specific examples of CASE functions in SQL Server

Previous article: [Check version information in SQL Server](#)

Next lesson: [COALESCE command in SQL Server](#)

You finished reading the article "**CASE statement in SQL Server**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.