2021-10-07T18:00:00+02:00 | Tihomir Babic

How to Use CASE in ORDER BY in SQL

This article will show you how and when to use CASE in an ORDER BY clause.

Have you ever used a `CASE` statement? I'm sure you have, at least in a SELECT statement. But have you ever used it in an `ORDER BY` clause? No? You will, once I show you how!

Don't worry if you've never used a `CASE` statement. I'll show and explain it to you with a short example. Then I'll move to other uses of the `CASE` statement, especially in an `ORDER BY` clause.

If you want multiple knowledge sources, here's an article explaining what CASE is. And here's another one explaining how you can use CASE to add logic to a SELECT.

What Is a CASE Statement?

To put it very simply, it's an SQL statement that goes through and returns values according to the conditions specified. It is SQL's way of writing the `IF-THEN-ELSE` logic and consists of five keywords: `CASE`, `WHEN`, `THEN`, `ELSE`, and `END`.

When used in a `SELECT` statement, it works like this: if it is the ***case when*** the condition is met, ***then*** return a certain value, or ***else*** return some other value, and ***end*** checking the conditions. The syntax looks like this:

```
CASE
 WHEN <condition> THEN <value>,
 WHEN <other condition=""> THEN <value>
 ELSE <value>
END AS <column name="">
</column></value></value></other></value></condition>
```

I think this syntax is best explained by how it works in an example. The most basic use of `CASE` is in a `SELECT` statement, so let's start with an example.

CASE in SELECT

Here's the table **`films`** I'll use in this example:

id	film_title	year	director
1	True Grit	2010	The Coen Brothers
2	Da 5 Bloods	2020	Spike Lee
3	Alien	1979	Ridley Scott
4	The Bridges Of Madison County	1995	Clint Eastwood
5	Get Out	2017	Jordan Peele
6	Annie Hall	1977	Woody Allen
7	Goodfellas	1990	Martin Scorsese
8	Dr.Strangelove	1964	Stanley Kubrick
9	You Were Never Really Here	2017	Lynne Ramsay
10	Albert Nobbs	2011	Rodrigo Garcia

Now, say we want to write a query that, along with the columns from the table, shows an additional column with the century in which the title was filmed. An example query looks like this:

```
SELECT
*,
    CASE
        WHEN year < 2001 THEN '20th-century film'
        ELSE '21st-century film'
    END AS century
FROM films;
```

This query selects all the columns from the table **`films`**. Then it uses a CASE statement to put values in a new column called century. This statement means: when the column year is below (i.e., older than) 2001, then the value in the column century should be '20th-century film'. If it's not, then the value should be '21st-century film'.

To learn more about functions in SQL, check out LearnSQL.com's [Standard SQL Functions] interactive course!

Let's see the results of the query:

id	film_title	year	director	century
1	True Grit	2010	The Coen Brothers	21st-century film
2	Da 5 Bloods	2020	Spike Lee	21st-century film
3	Alien	1979	Ridley Scott	20th-century film
4	The Bridges Of Madison County	1995	Clint Eastwood	20th-century film
5	Get Out	2017	Jordan Peele	21st-century film
6	Annie Hall	1977	Woody Allen	20th-century film
7	Goodfellas	1990	Martin Scorsese	20th-century film
8	Dr.Strangelove	1964	Stanley Kubrick	20th-century film
9	You Were Never Really Here	2017	Lynne Ramsay	21st-century film
10	Albert Nobbs	2011	Rodrigo Garcia	21st-century film

Is CASE Used Only in SELECT Statements?

Nope! Although it is most often used there, `CASE` is not limited to `SELECT` statements. For example, you can use it in clauses like `IN`, `WHERE`, `HAVING`, and `ORDER BY`.

Using a `CASE` statement in a query once doesn't mean you have hit your quota for using it. You can use it multiple times in a single query. Or you can use it just once. This means you don't need to have `CASE` in `SELECT` if you just want to use it in an ORDER BY.

Speaking of which, the main point of this article is to show you how the `CASE` statement works in an `ORDER BY` clause. I'm getting there now.

When Is a CASE Statement Used in an ORDER BY?

The `ORDER BY` clause is used to sort the result in either ascending or descending order. Want a refresher on how it works? No problem! This article explains what an ORDER BY does.

Introducing a `CASE` statement here can enhance your `ORDER BY` by allowing you to order results by some other (and multiple) criteria. For example, you can use it when you want to sort string values according to some criteria other than alphabetical order, such as by the hierarchical positions of job titles in a company.

Using a `CASE` statement also allows you to sort data according to multiple criteria. For example, you may want to sort your users by country and name, but by country and state instead if they are from the U.S.

I'll show you a practical example, and you'll immediately know what I'm talking about.

CASE in ORDER BY

Here, we'll be working with the table **`shops`**, which contains the following data:

id	shop_name	country	state	city
1	Zoltan's shop	Croatia	NULL	Zagreb
2	Ante Portas	Croatia	NULL	Rijeka
3	Green Mile 1	USA	Tennessee	Memphis
4	Jan Pieter's Pita	Croatia	NULL	Split
5	Green Mile 2	USA	Tennessee	Chattanooga
6	Green Mile 3	USA	New Orleans	Louisiana
7	Krystyna's shop	Hungary	NULL	Pecs
8	Quinn & Sandy	Hungary	NULL	Gyor
9	Green Mile 4	USA	California	San Francisco
10	Fragola	Croatia	NULL	Osijek

We have to order the data by `country` first. Then, every shop within the same country should be sorted by `city`. If the shop is in the U.S., we need to sort it next by the column `state`. The code that solves this little problem is:

```
SELECT *
FROM shops
ORDER BY country,
    CASE
        WHEN country = 'USA' THEN state
        ELSE city
    END;
```

It selects all the columns from the table **`shops`**. It then orders the result first by country, then by the criteria in the `CASE` statement. It says if `country = 'USA'`, then the result is sorted by state. If it's not, then it is sorted by `city`.

It looks like this code indeed returns the desired result:

id	shop_name	country	state	city
10	Fragola	Croatia	NULL	Osijek
2	Ante Portas	Croatia	NULL	Rijeka
4	Jan Pieter's Pita	Croatia	NULL	Split
1	Zoltan's shop	Croatia	NULL	Zagreb
8	Quinn & Sandy	Hungary	NULL	Gyor
7	Krystyna's shop	Hungary	NULL	Pecs
9	Green Mile 4	USA	California	San Francisco
6	Green Mile 3	USA	New Orleans	Louisiana
5	Green Mile 2	USA	Tennessee	Chattanooga
3	Green Mile 1	USA	Tennessee	Memphis

Other Uses of the CASE Statement

You can also use the `CASE` statement with keywords other than the ones I have mentioned. For example, you can use it with `DESC`. Let's say I put it in the above code:

```
SELECT *
FROM shops
ORDER BY country,
    CASE
        WHEN country = 'USA' THEN state
        ELSE city
    END DESC;
```

In the previous example, I've omitted the `DESC` keyword. The result is sorted in ascending order by default. By writing `DESC` after the `CASE` statement, you get the shops within the countries ordered by `city` in descending order, not ascending as in the previous case. U.S. shops are ordered by `state` but this time in descending order.

We can see the result here:

id	shop_name	country	state	city
1	Zoltan's shop	Croatia	NULL	Zagreb
4	Jan Pieter's Pita	Croatia	NULL	Split
2	Ante Portas	Croatia	NULL	Rijeka
10	Fragola	Croatia	NULL	Osijek
7	Krystyna's shop	Hungary	NULL	Pecs
8	Quinn & Sandy	Hungary	NULL	Gyor
5	Green Mile 2	USA	Tennessee	Chattanooga
3	Green Mile 1	USA	Tennessee	Memphis
6	Green Mile 3	USA	New Orleans	Louisiana
9	Green Mile 4	USA	California	San Francisco

You can also use `CASE` statements with `UPDATE`, `INSERT`, and `DELETE`. These are called data modifying statements. Here's an article explaining how to use CASE with those statements. You can also use it for conditional summarization.

Do you want to learn all the basics of SQL in one place? Go through our SQL From A to Z track!

Want to Learn More About the CASE Statement?

It probably comes to you as a surprise that `CASE` statements aren't used only in `SELECT` statements. You've seen it come in handy when used with an `ORDER BY`. It opens up the possibilities for ordering query results.

That's not the end. The `CASE` statement can be used with some other statements and keywords, such as data modifying statements, or even `DESC`, `IN`, `WHERE`, and `HAVING`.

Now it's time to learn more and practice what you learn. You get all that with our Standard SQL Functions course. This course is a part of a much wider track named SQL from A to Z. What are you waiting for?

Viewed using Just Read