# **MySQL Control Flow Functions and Expressions**

The control flow functions allow you to add if-then-else logic to SQL queries without using the procedural code. The following show the most commonly used MySQL control flows functions:

* [CASE](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-case-function/) – return the corresponding result in THEN branch if the condition in the WHEN branch is satisfied, otherwise, return the result in the ELSE branch.
* [IF](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-if-function/) – return a value based on a given condition.
* [IFNULL](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-ifnull/)– return the first argument if it is not NULL , otherwise returns the second argument.
* [NULLIF](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-nullif/)– return NULL if the first argument is equal to the second argument, otherwise, returns the first argument.

# **MySQL CASE Expression**

**Summary**: in this tutorial, you will learn how to use the MySQL CASE expression to add if-else logic to queries.

## **Introduction to MySQL CASE expression**

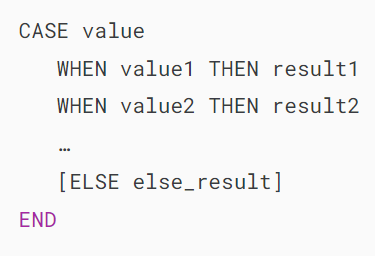
MySQL CASE expression is a control flow structure that allows you to add if-else logic to a query. Generally speaking, you can use the CASE expression anywhere that allows a valid expression e.g., [SELECT](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/), [WHERE](https://www.mysqltutorial.org/mysql-basics/mysql-where/) and [ORDER BY](https://www.mysqltutorial.org/mysql-basics/mysql-order-by/) clauses.

The CASE expression has two forms: simple CASE and searched CASE.

Note that MySQL has a [CASE](https://www.mysqltutorial.org/mysql-stored-procedure/mysql-case-statement/) statement that you can use only in stored programs such as stored procedures, stored functions, events, and triggers, which is not the CASE expression covered in this tutorial.

### **Simple CASE expression**

The following illustrates the syntax of a simple  CASE expression:

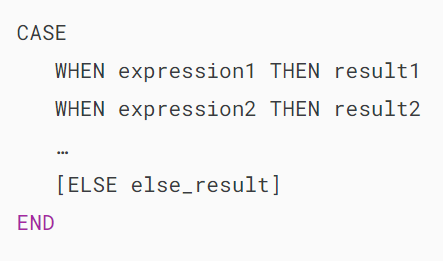


In this syntax, CASE matches the value with the value1, value2, etc., for equality and return the corresponding result1, result2,… If the value does not equal to any value1, value2, … CASE returns the result in the ELSE clause if the ELSE clause is specified.

The CASE compares the value with values in the WHEN clauses for equality, you cannot use it with NULL because NULL = NULL returns false.

### **Searched CASE expression**

The following shows the syntax of a searched CASE expression:



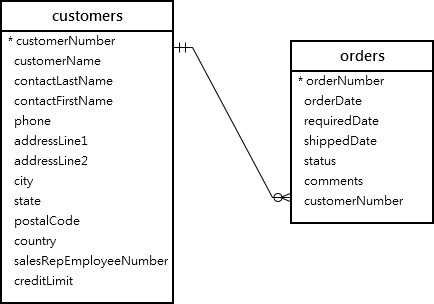
In this syntax, the CASE evaluates expressions specified in the WHEN clauses. If an expression evaluates to true. CASE returns the corresponding result in the THEN clause. Otherwise, it returns the result specified in the ELSE clause. In case the ELSE clause is not available, then the CASE expression returns NULL .

The CASE expression returns a result whose [data type](https://www.mysqltutorial.org/mysql-basics/mysql-data-types/) depends on the context where it is used. For example, if the CASE expression is used in the character string context, it returns the result as a character string. If the CASE expression is used in a numeric context, it returns the result as an integer, a decimal, or a real value.

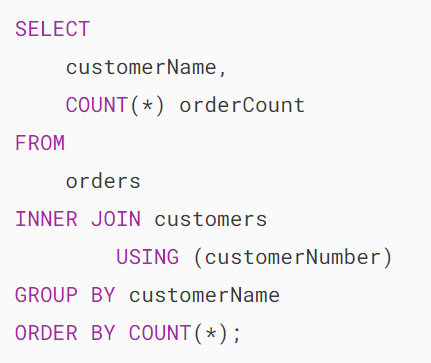
## **MySQL CASE expression examples**

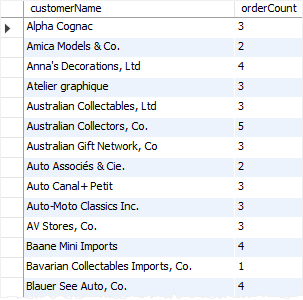
### **1) Using CASE expression in the SELECT clause example**

See the following orders and customers tables:

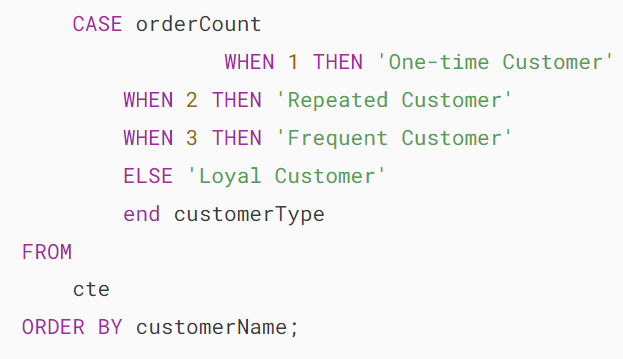
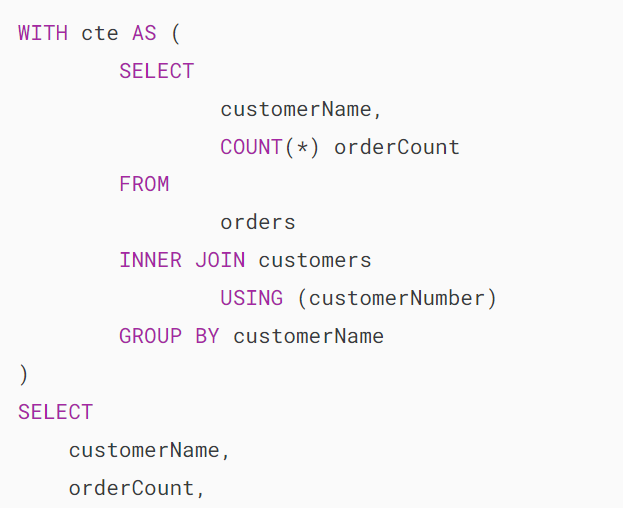


The following statement returns the customers and their orders:





This example uses the CASE expression in the SELECT clause to return the type of customers based on the number of orders that customers ordered:



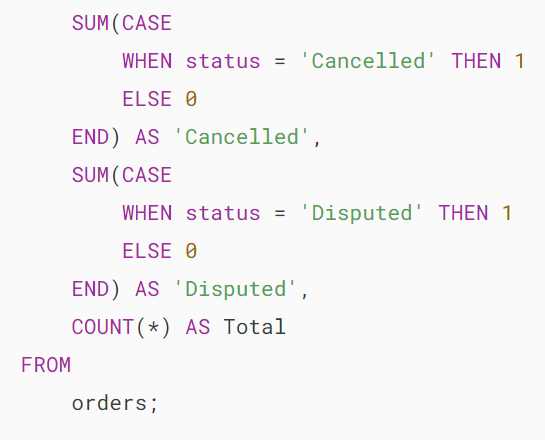
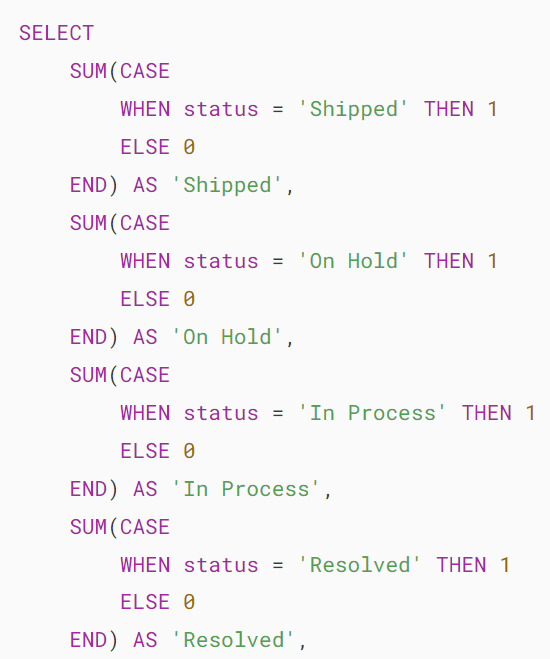
### **2) Using CASE expression in the ORDER BY clause example**

The following example uses the CASE expression to sort customers by state if the state is not NULL,  or sort the country in case the state is NULL:



### **3) Using CASE expression with an aggregate function example**

The following example uses the CASE expression with the [SUM()](https://www.mysqltutorial.org/mysql-aggregate-functions/mysql-sum/) function to calculate the total of sales orders by order status:



Here is the output:

MySQL CASE expression with aggregate function example

How it works.

* First, the CASE statement returns 1 if the status equals the corresponding status such as Shipped, on hold, in Process, Cancelled, Disputed, and zero otherwise.
* Second, the SUM() function returns the total number of orders per order status.

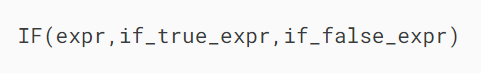
# **MySQL IF Function**

**Summary**: in this tutorial, you will learn how to use **MySQL IF function** that returns a value based on a given condition.

## **Introduction to MySQL IF function**

MySQL IF function is one of the MySQL control flow functions that returns a value based on a condition. The IF function is sometimes referred to as IF ELSE or IF THEN ELSE function.

The syntax of the MySQL IF function is as follows:



If the expr evaluates to TRUE i.e., expr is not NULL and expr is not 0, the IF function returns the if\_true\_expr , otherwise, it returns if\_false\_expr The IF function returns a numeric or a string, depending on how it is used.

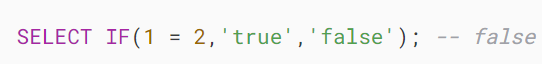
The IF function that we cover in this tutorial is different from the [IF statement](https://www.mysqltutorial.org/mysql-stored-procedure/mysql-if-statement/).

## **MySQL IF function Examples**

Let’s practice with several examples to see how the MySQL IF function works.

### **A simple IF function example**

You can use the IF function directly in the [SELECT statement](https://www.mysqltutorial.org/mysql-basics/mysql-select-from/) without the FROM and other clauses as follows:

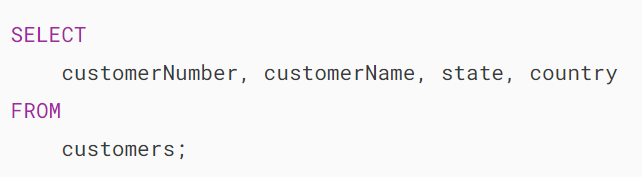


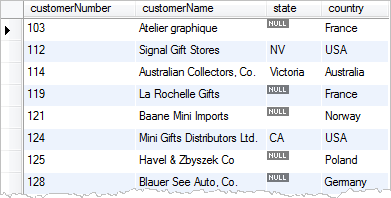
### 

### **Displaying N/A instead of NULL using MySQL IF function**

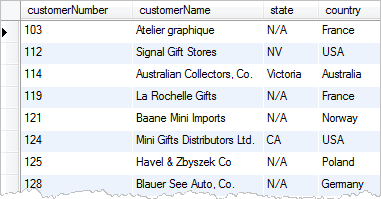
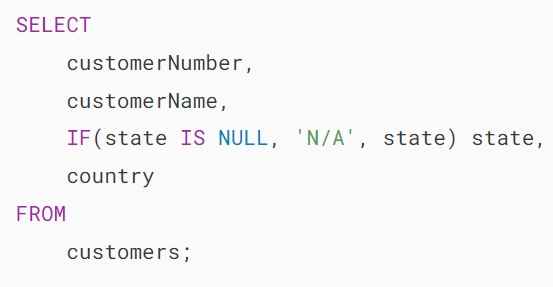
Let’s take a look at the data in the customers table in the [sample database](https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/).

In the customers table, many customers do not have state data in the state column therefore when we select customers, the state column displays NULL values, which is not meaningful for the reporting purpose. See the following query:





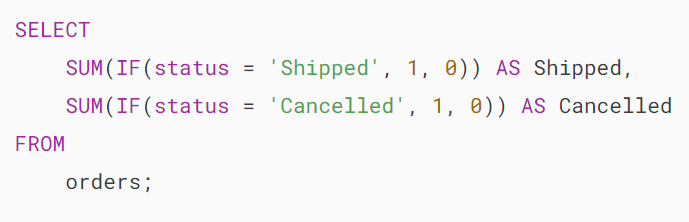
We can improve the output by using the IF function to return N/A if the state is NULL as the following query:



## **MySQL IF function with aggregate functions**

### **MySQL SUM IF – Combining the IF function with the SUM function**

The IF function is useful when it combines with an aggregate function. Suppose if you want to know how many orders have been shipped and cancelled, you can use the IF function with the [SUM aggregate function](https://www.mysqltutorial.org/mysql-aggregate-functions/) as the following query:

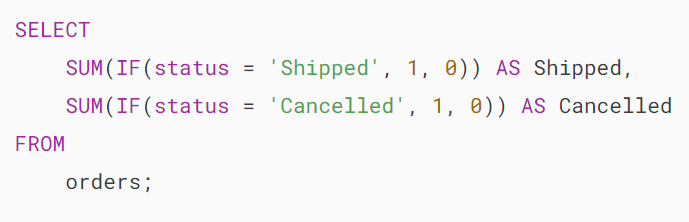


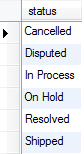
MySQL SUM IF

In the query above, if the order status is shipped or cancelled , the IF function returns 1 otherwise it returns 0. The SUM function calculates the total number of shipped and cancelled orders based on the returned value of the IF function.

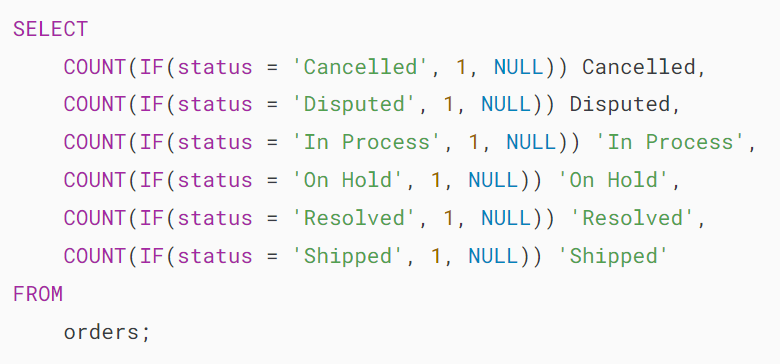
### **MySQL COUNT IF – Combining the IF function with the COUNT function**

First, we select [distinct](https://www.mysqltutorial.org/mysql-basics/mysql-distinct/) order’s status in the orders table using the following query:



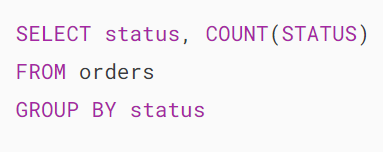


Second, we can get the number of orders in each status by combining the IF function with the COUNT function. Because the COUNT function does not count NULL values, the IF function returns NULL if the status is not in the selected status, otherwise it returns 1. See the following query:



MySQL COUNT IF - orders' status

Of course, you can achieve the similar result using the [GROUP BY clause](https://www.mysqltutorial.org/mysql-basics/mysql-group-by/) and the COUNT function without using the IF function as the following query:



In this tutorial, we have introduced you to the MySQL IF function which helps you write a query with condition embedded in the SELECT clause.

## **Reference**

* <http://dev.mysql.com/doc/refman/5.7/en/control-flow-functions.html#function_if> – MySQL IF function

# **MySQL IFNULL**

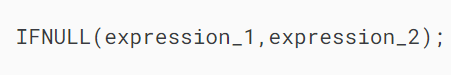
**Summary**: in this tutorial, you will learn about the **MySQL IFNULL** function, which is a very handy control flow function to handle NULL values.

## **Introduction to MySQL IFNULL function**

MySQL IFNULL function is one of the MySQL control flow functions that accepts two arguments and returns the first argument if it is not NULL. Otherwise, the IFNULL function returns the second argument.

The two arguments can be literal values or expressions.

The following illustrates the syntax of the IFNULL function:



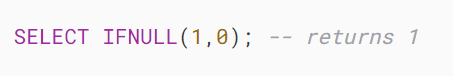
The IFNULL function returns expression\_1 if expression\_1 is not NULL ; otherwise, it returns expression\_2. The

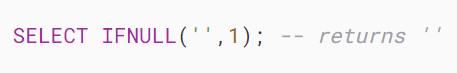
The IFNULL function returns a string or a numeric based on the context where it is used.

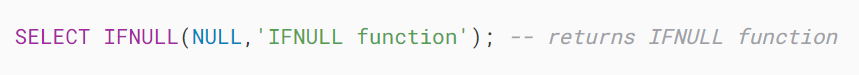
If you want to return a value based on TRUE or FALSE condition other than NULL, you should use the [IF function](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-if-function/).

## **MySQL IFNULL function examples**

See the following IFNULL function examples:





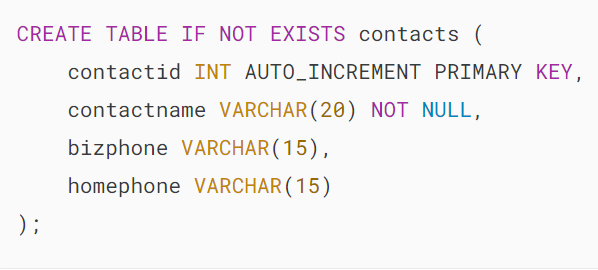


How it works.

* IFNULL(1,0) returns 1 because 1 is not NULL .
* IFNULL(' ',1) returns ' ' because the ' ' string is not NULL .
* IFNULL(NULL,'IFNULL function') returns IFNULL function string because the first argument is NULL.

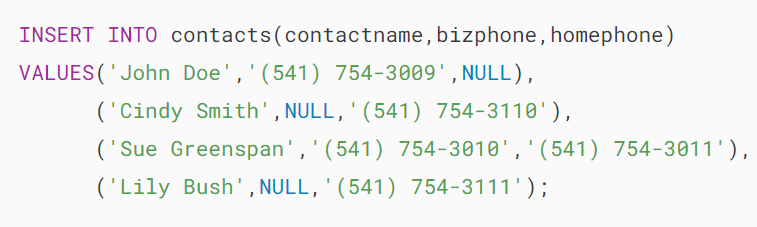
Let’s take a practical example of using the IFNULL function.

First, [create a new table](https://www.mysqltutorial.org/mysql-basics/mysql-create-table/) named contacts using the following statement:

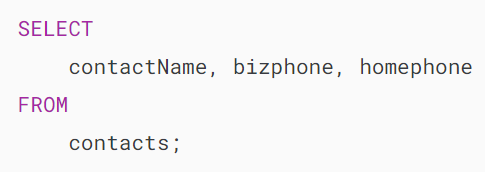


Each contact has a name, business phone and home phone.

Second, [insert data](https://www.mysqltutorial.org/mysql-basics/mysql-insert/) into the contacts table:



Some contacts have only home phone or business phone. To get all the contact name and phone from the contacts table, you use the following query:

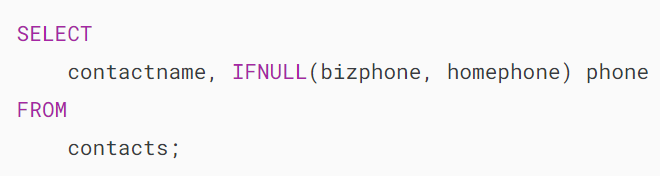


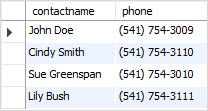


It would be nice if we can get the contact’s home phone if the contact’s business phone is not available. This is where the

This is where the IFNULL function comes to play. The IFNULL function returns the home phone if the business phone is NULL.

Third, use the following query to get the names and phones of all the contacts:





Notice that you should avoid using the IFNULL function in the [WHERE clause](https://www.mysqltutorial.org/mysql-basics/mysql-where/), because it degrades the performance of the query. If you want to check if a value is

If you want to check if a value is NULL or not, you can use IS NULL or IS NOT NULL in the WHERE clause.

In this tutorial, we have introduced you to MySQL IFNULL function and shown you how to use the IFNULL function in the queries.

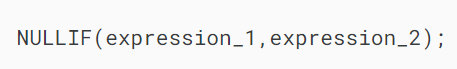
# **MySQL NULLIF**

**Summary**: in this tutorial, you will learn about the **MySQL NULLIF** function and how to use it to prevent the division by zero error in a query.

## **Introduction to MySQL NULLIF function**

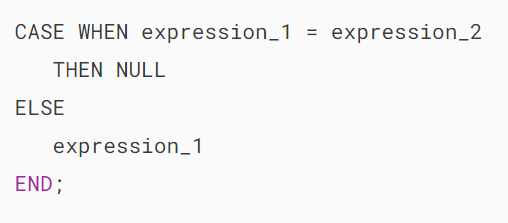
The NULLIF function is one of the control flow functions that accepts 2 arguments. The NULLIF function returns NULL if the first argument is equal to the second argument, otherwise it returns the first argument.

The syntax of the NULLIF function is as follows:



The NULLIF function returns NULL if expression\_1 = expression\_2 is true , otherwise it returns expression\_1

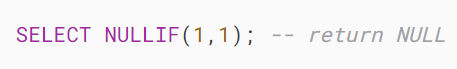
Notice that the NULLIF function is similar to the following expression that uses the [CASE expression](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-case-function/):

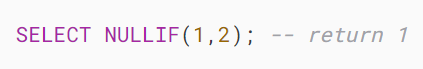
\

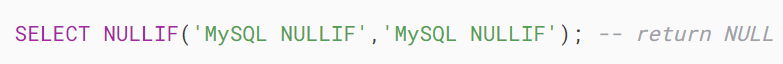
Note that you should not confuse the NULLIF function a similar function called [IFNULL function](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-ifnull/).

## **MySQL NULLIF examples**

Let’s take a look at some examples of using the NULLIF function to understand how it works.













How the statements work.

* NULIF(1,1) returns NULL because 1 is equal 1.
* NULLIF(1,2) returns 1, which is the first argument, because 1 is not equal 2.
* NULLIF('MySQL NULLIF','MySQL NULLIF') returns NULL because two arguments are the same string
* NULLIF('MySQL NULLIF','MySQL NULLIF') returns MySQL NULLIF because two strings are not equal.
* NULLIF(1,NULL) returns 1 because 1 is not equal to NULL
* NULLIF(NULL,1) return the first argument i.e., NULL , because NULL is not equal to 1.

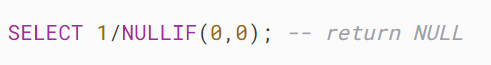
## **Using NULLIF function to prevent division by zero error**

We often use the NULLIF function to prevent the division by zero error in a query. If the MySQL server has ERROR\_FOR\_DIVISION\_BY\_ZERO mode enabled, it will issue an error when a division by zero occurred.

See the following statement:

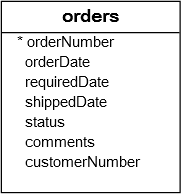


In this case, you can use the NULLIF function to prevent the division by zero as follows:

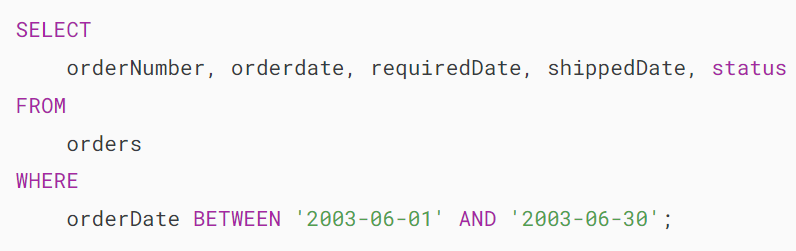


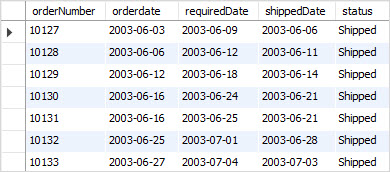
Because zero is equal to zero, the expression NULLIF(0,0) returns NULL. As the result, the statement returns NULL.

Let’s take a look at the orders table in the [sample database](https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/).

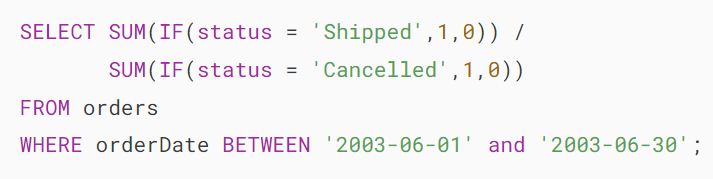


First, to get all orders created in June 2003, you use the following query:



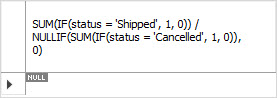
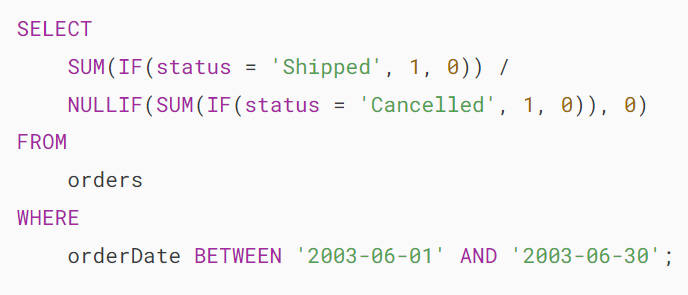


Second, calculate the number of shipped orders / the number of cancelled orders in June 2003, you use the [SUM](https://www.mysqltutorial.org/mysql-aggregate-functions/mysql-sum/) and [IF](https://www.mysqltutorial.org/mysql-control-flow-functions/mysql-if-function/) functions.



MySQL issues an error because in June 2003 there was no cancelled order created. It means that expression SUM(IF(status = 'Cancelled',1,0)) returns zero.

Third, to prevent the division by zero error, you use the NULLIF function as the following query:



Because there was no cancelled order created in June 2003, the SUM(IF(status = 'Cancelled',1,0) expression returns zero, which also makes the NULLIF(SUM(IF(status = 'Cancelled',1,0),0) expression returns a NULL value.

In this tutorial, we have introduced you to NULLIF function, which is very handy in some cases such as preventing division by zero error in queries.