SQL percentage calculation examples

Here, you will see the different ways to calculate SQL percentage between multiple columns and rows. You will also see how to calculate SQL percentages for numeric columns, grouped by categorical columns. You will use subqueries, the OVER clause, and the common table expressions (CTE) to find SQL percentages.

So, let’s begin without any ado.

## **Finding Percentage using Two Variables**

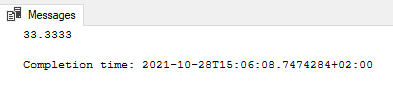
There is no built-in operator that calculates percentages in SQL Server. You have to rely on basic arithmetic operations i.e. (number1/number2 x 100) to find percentages in SQL Server.

Before finding the SQL percentages across rows and columns, let’s first see how you can find percentages using two basic variables in SQL Server.

The script below defines three float variables @num1, @num2 and @perc. Next, the @num2 variable is divided by the @num1 variable and the result is multiplied by 100 which is stored in the @perc variable and is printed on the console.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | DECLARE @num1 as FLOAT  DECLARE @num2 as FLOAT    DECLARE @perc as FLOAT  SET @num1 = 150  SET @num2 = 50    SET @perc = @num2/@num1 \* 100    PRINT @perc |

**Output:**



## **Finding Percentages Between Two Columns**

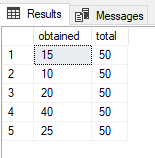
Finding Percentages between two columns is straightforward. You can simply use the column names and the division operator “/” to divide values in one column by another. The result is a list of values that correspond to the result of the division of all the values in the two columns.

Let’s see an example.

The script below, creates a table Result with two float type columns “obtained”, and “total”. The script also inserts five dummy rows in the Result table. The SELECT query then selects all the records in the Result table. Here is an example:

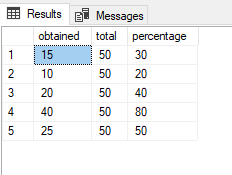
|  |  |
| --- | --- |
| 1  2  3  4  5 | CREATE TABLE Result(obtained float, total float)  INSERT INTO Result values(15,50),(10,50),(20,50),(40,50),(25,50)    SELECT \*  FROM Result |

**Output:**



Let’s try to find percentages for each row as a result of the division between the values in the “obtained” and “total” columns as shown below where a new column is added for percentages.

|  |  |
| --- | --- |
| 1  2 | SELECT obtained, total, obtained/total \* 100 as 'percentage'  FROM Result |



## **Finding Percentages via Subqueries**

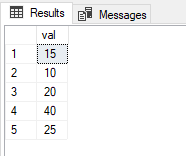
Finding SQL percentages between two columns is straightforward. However, the process is not as straightforward for finding percentages across rows for different scenarios.

Let’s first discuss a very simple scenario where you have to find what is the percentage of a value in a column among all the rows in the column.

The following script creates a table Scores with one column.

|  |  |
| --- | --- |
| 1  2  3  4  5 | CREATE TABLE Scores(val float)  INSERT Scores(val) values(15),(10),(20),(40),(25);    SELECT \*  FROM Scores |

**Output:**



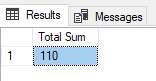
Now if you want to find what percent of the sum of total values in the “val” column does each value constitutes, you can use subqueries.

In this regard, the outer query will multiply all the values in the “val” column by 100 which will be divided by the result of the subquery which finds the sum of all the values in the “val” column.

Let’s first see how our subquery looks that calculate the sum of the values in the “val” column.

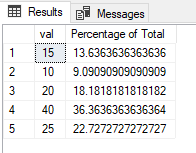
|  |  |
| --- | --- |
| 1  2 | SELECT SUM(val) as 'Total Sum'  FROM Scores |

**Output:**



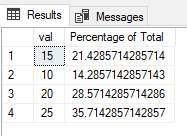
The following script returns the percentage of the total for each value in the “val ” column.

|  |  |
| --- | --- |
| 1  2  3 | SELECT val,  val \* 100/(SELECT SUM(val) FROM Scores) as 'Percentage of Total'  From Scores |



If you do not want to exclude any value while calculating the percentage of the total, you can do so with the WHERE clause as shown in the script below where the value 40 is not included.

|  |  |
| --- | --- |
| 1  2 | SELECT val,  val \* 100/(SELECT SUM(val) FROM Scores WHERE val < 40) as 'Percentage of Total' From Scores WHERE val < 40 |

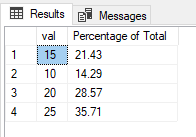


You can see from the above output that the values now have a larger percentage share of total values since the value 40 is removed.

Finally, as a side note, you can round off the percentages returned using the “round” function as shown below. The script below rounds off the percentage values to 2 decimal places.

|  |  |
| --- | --- |
| 1  2 | SELECT val,  round(val \* 100/(SELECT SUM(val) FROM Scores WHERE val < 40), 2) as 'Percentage of Total' From Scores WHERE val < 40 |

**Output:**



Let’s now see a real-world example of how you can calculate SQL percentage. You will be using the Northwind sample database which you can download and install from [this link](https://github.com/Microsoft/sql-server-samples/tree/master/samples/databases/northwind-pubs).

Run the following script to see the columns in the Products table.

|  |  |
| --- | --- |
| 1  2 | USE Northwind  SELECT \* FROM Products |

**Output:**



The Products table contains columns that contain Supplier ID, Category ID, Unit Price and other information about the products in the Northwind database.

Consider a scenario where you have to find the percentage of the products supplied by each supplier. To find such percentage values, you need two values:

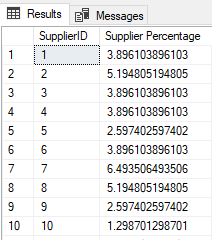
1. The total number of all the products which you can get via the COUNT function
2. The total number of products supplied for each supplier, which you can get using the GROUP BY function.

You can then multiply the 2nd value (count of products grouped by supplier ids) by 100 and then divide the result by the 1st value (total count of products).

Here is how you can use subqueries to find these such SQL percentages.

|  |  |
| --- | --- |
| 1  2  3  4 | USE Northwind  SELECT SupplierID, count(\*) \* 100.0 / (SELECT count(\*) from Products) as 'Supplier Percentage'  FROM Products  GROUP BY SupplierID |

**Output:**



The above results show that the percentage of products supplied by each supplier. For instance, you can see that the supplier with id 1 supplied 3.89% of the total products.

## **USING Over Clause**

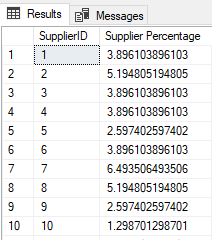
The Over clause is an extremely useful Window function that calculates values over a range of values. You can use the Over clause to calculate SQL percentages as well. With the Over clause, you can avoid the use of subqueries for calculating percentages.

Let’s see an example. The script below finds the percentage of products supplied by every supplier. You can see that the following script is very similar to what you saw in the previous section. However, in this case, instead of using a subquery that returns the count of all products, we use the Over clause which returns the sum of all the products.

|  |  |
| --- | --- |
| 1  2  3  4 | USE Northwind  SELECT SupplierID, count(\*) \* 100.0 / sum(count(\*)) Over() as 'Supplier Percentage'  FROM Products  GROUP BY SupplierID |

The output below is similar to what you achieved using a subquery.

**Output:**



Let’s now see a more complex example of finding SQL percentages. Consider a scenario where for each supplier you want to calculate the percentage of unit price for all the products. In other words, you want to find out that what percentage of the sum of unit prices for all the products is being paid to a different supplier. You can do so with the help of the OVER clause as follows.

In the script below, the supplier ids are displayed along with the sum of unit prices paid to all suppliers, and the percentages of unit prices paid to all suppliers.

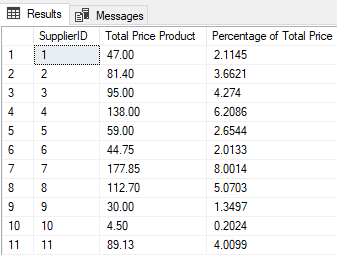
To calculate the sum of unit prices, the formula is simple, you can use the SUM function and pass it to the column containing the unit prices.

To calculate the percentages for unit prices, you need to multiply the sum of unit prices for each supplier by 100 and then divide the result with the total sum of unit prices for all the suppliers. In the script below, in the denominator, the SUM function is called twice. Since we are using the OVER operator, the first SUM function only adds the prices for each supplier. To find the sum of unit prices paid to all suppliers, another SUM function is called.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | USE Northwind    SELECT SupplierID, SUM(UnitPrice) as 'Total Price Product',  (SUM(UnitPrice) \* 100 )/SUM(SUM(UnitPrice)) OVER () as 'Percentage of Total Price'    FROM Products  GROUP BY SupplierID |

In the output below, you can see the supplier ids, the sum of unit prices paid to each supplier and the percentage of unit prices for each supplier.

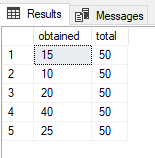
**Output:**



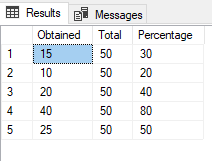
## **Using Common Table Expressions**

Finally, you can also use common table expressions (CTE) to calculate percentages. Let’s first see how you can use CTEs to find percentages between values in two columns.

The script below finds the percentages by dividing the values in the “obtained” column by the values in the “total” column.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | WITH ResultCTE(Obtained, Total, Percentage)  AS  (  SELECT obtained,  total,  (obtained/total) \* 100 Percentage  FROM Result  )  SELECT \* FROM ResultCTE |

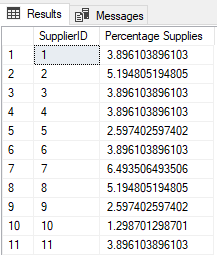


You can use CTE expressions to find more complex SQL percentages just as you did with the OVER clause. For example, the script below uses the CTE to calculate the percentage of products supplied by each supplier.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | USE Northwind;  with ProductCTE(SupplierID, Supplier\_Count)  as  (  select SupplierID, count(\*)  from Products  group by SupplierID  )  SELECT SupplierID, Supplier\_Count \* 100.0/(select sum(Supplier\_Count) from ProductCTE) as 'Percentage Supplies'  from ProductCTE; |

The result is similar to what you achieved via the OVER clause.

**Output:**



Finally, you can also use the OVER clause in combination with the CTE to calculate percentages. For example, the script below uses the CTE to find the percentage of products supplied by each supplier, along with the percentage of unit prices paid to all the suppliers for all the products.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | USE Northwind;  with ProductCTE(SupplierID, Supplier\_Count, Price\_Percentage)  as  (  select SupplierID, SUM(UnitPrice) as 'Total Price Product',  (SUM(UnitPrice) \* 100 )/SUM(SUM(UnitPrice)) OVER () as 'Percentage of Total Price'  from Products  group by SupplierID  )  SELECT \*  from ProductCTE; |

