

TARGET DASHBOARD - Create and Share KPIs



TARGET
DASHBOARD

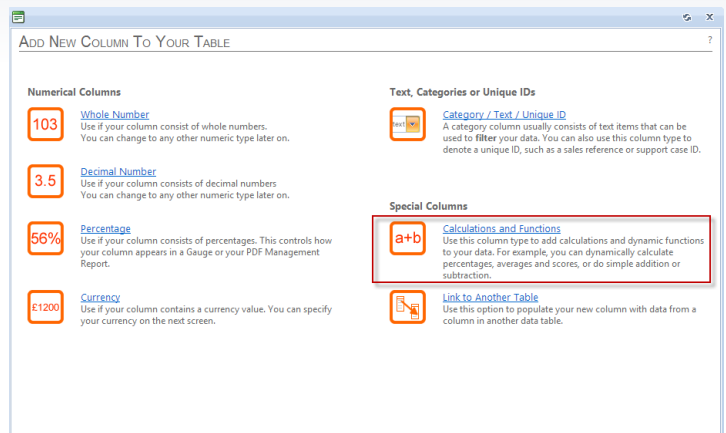
Target Dashboard Formulae & Dynamic Functions

Much like in Excel, you can use formulae in Target Dashboard to do calculations with your data. To do this, you would create a *Calculated Column* where your calculation will be carried out, displaying the results directly in your data table.

To create a calculated column in Target Dashboard you need to go to the *Table Designer* for the table where you want to add your column.

Click on *Add New Column*, type in a name, and select the calculated column from the drop down list.

When you've done that, you need to add your formula. You can find a list of all the formulae that are valid in Target Dashboard below.



Simple Formulae in Target Dashboard

These calculations are always 'horizontal' and are carried out for each row in the calculated column where there is data to calculate. They are carried out directly in your data table, and you will see the results in your table as soon as you save your calculated column.

*In Target Dashboard we use unique reference IDs for each column. These will need to be inserted into each individual formula that is created instead of the placeholders we have used for the purposes of this document. In Target Dashboard, these references can be found in the table on the right hand side of the formula builder, and they should always reside within [] when used in a formula. **The column references can be dragged and dropped from the table on the right hand side to the formula editor.***

List of Simple Formulae

[col1]+[col2]

This is simple addition. Data from the two columns will be added together, and the new value will be displayed in the calculated column.

[col1]-[col2]

[23456]-[65432]

This is simple subtraction. Data from the second column will be subtracted from the first, and the new value will be displayed in the calculated column.

[col1]*[col2]

[23456][65432]*

This is simple multiplication. Data from the two columns will be multiplied, and the new value will be displayed in the calculated column.

[col1]/[col2]

[23456]/[65432]

This is simple division. Data from the first column will be divided by the second and the new value will be displayed in the calculated column.

[col1]/[col2]*100

*[23456]/[65432]*100*

This will produce a percentage that the second column is of the first.

sum([col1],[col2],[col3]...)

sum([23456],[65432],[56789]....)

This formula will produce a sum of the specified columns. You can specify as many columns as you would like to sum from the same table.

sra([col1],no of rows)

sra([23456],12)

This formula will produce a simple rolling average across a specified number of rows. For example, if you have numerical data with 1 row per month, you can produce a rolling average for a set number of months. *Please note*, however, that this formula only works with purely numerical data, and will not produce an accurate rolling average when applied to a table that contains categorised data.

Continued on next page...

...Simple Formulae Continued

nbool([col1],value to convert)

nbool([23456],5)

This formula can be used to convert any numeric value into a 1 or a 0. It will convert the specified value into a 1, and anything else into a 0. This can be useful when creating a data table used to chart responses and calculating a score for a multiple choice survey (such as an NPS survey).

sbool([col1],"text to check")

sbool([23456],"yes")

This formula can be used to convert a text value into a 1 or a 0. It will convert the specified value into a 1, and anything else into a 0. This can be useful when creating a data table used to chart responses and calculating a score for a multiple choice survey (such as an NPS survey).

If([col1],"textToCheck",[col2],0)

If([234125],"blue",[543678],0)

This formula can be used to convert a text value (from [col1]) to any numeric value, either from another column ([col2]) or to a numeric value that you specify (replaces [col2]). It works the same way the *sbool()* formula does, but offers more customisation. It can be very useful if you're creating categorised stacked charts, for example.

Nif([col1],">",10,1)

Nif([123456],">",10,1)

This formula can be used to calculate if the value in the specified column is more, less or equal to a certain threshold. The operators that can be applied are: > (more than), < (less than), >= (more than or equal to), <= (less than or equal to), = (equal to).

The value after the operator is the threshold you wish to compare to, and the last value is what will be populated in the formula column if the conditions of the operator are met.

betbool([column Ref],lower value,upper value)

betbool([12356],24,49)

This is a formula for setting brackets in your data. It's a Boolean formula, which means when applied, it will create a column of ones and zeros based on the numeric criteria you specify within the formula.

In practice, perhaps you wish to create an age bracket in your data. You want to know issues that are between 24 and 48 hours old. The formula would then look something like the italicised example above.

Please note that the upper limit you pick is exclusive, so to include anything up to 48.99999..., you would need to put 49 as the upper value in your formula.

1

1

This simple formula puts a 1 on every row in the calculated column, and will continue to do so for every row that is added. It can be used to create a simple count of rows. This works very well with highly categorised data where you want to count how many of each category you have.

Dynamic Functions in Target Dashboard

Dynamic Functions in Target Dashboard are different from regular formulae because of how and when they are calculated. With a regular formula, the calculation is done directly in your data table, which is absolutely fine when you're calculating a value that accumulates across months, such as a sales value or your actual margin.

However, if you are calculating a percentage that can't be allowed to cumulate, or perhaps you have monthly data, but you need to divide the totals from two columns and display that value over a whole year. If you were to use a regular formula to do this, each month's value would cumulate and you would end up with an inaccurate figure. Because a dynamic function is just that – dynamic, it calculates the value *after* your data has been filtered and grouped in a chart on the dashboard, resulting in accurate values, no matter how you choose to display your data.

Another great use of dynamic functions is to calculate things like weighted averages, NPS (Net Promoter Score) and CSI (Customer Service Index) scores. Using these functions you can use Target Dashboard to chart the results of your customer satisfaction surveys, for example.

List of Dynamic Functions

Note that all of these dynamic functions have a dollar sign (\$) in front. This indicates that they are dynamic in Target Dashboard, and they will not work without it.

\$pct([col1],[col2])

\$pct([23456],[65432])

This function will calculate a dynamic percentage that will never accumulate, whether you choose to display your data in a daily, weekly, monthly or yearly format.

\$div([col1],[col2])

\$div([23456],[65432])

This function will dynamically divide one column by another and the result will work dynamically across filters and date groupings.

\$wqs([col1]:0,[col2]:5,[col3]:10)

\$wqs([23456]:0,[65432]:5,[56789]:10)

This function calculates a weighted average. In the example above, the value in column one is weighted as 0, column 2 as 5 and column 3 as 10. You can of course use more columns than three, and different weightings to the ones that are used for the purposes of this example. This would work well in, for example, calculating the results of a health and safety survey where the different answers (e.g bad, average, good) have different weighting.

In Target Dashboard, each answer option would have its own column, and if that answer was selected, the column would be populated with a 1. (This type of layout can easily be achieved with the *sbol()* formula described in the section about simple formulae if your data is text based) The formula then takes all the answers, applies the weighting and produces a weighted score that is dynamic across filters and date groupings.

This type of function could be combined with a custom feature in Target Dashboard, where we build the input form for your survey for you. You can then use it to collect responses on a tablet or send it out to customers and employees to fill in on the web. The results would be inserted directly into your Tar-

get Dashboard table, and your dashboards and presentations would update as the results come in.

\$nps([detractor]:[detractor],[neutral],[promoter]:[promoter])

\$nps([23456]:[65432],[56789],[98765]:[67890])

The NPS score is a measure of -100 to 100 of how likely someone is to recommend your company. You might have a question like: *How likely are you to recommend our service to your friends and family?* With answers ranging from *Not likely at all* to *Very likely*. Negative answers are called *detractors*, positive answers are *promoters* and in between are our neutral answers.

The data needs to be laid out in the same way as for the *\$wqs()* formula, and the score will be completely dynamic when the formula is put in place. Each detractor, promoter and neutral answer needs its own column in Target Dashboard, and in the bolded formula above, the text between the commas and colons would be replaced with column IDs.

To achieve the correct layout for charting an NPS survey, you can also use the NPS wizard that can be found in the *Table Designer* of any data table in Target Dashboard.

\$csi([col1]:0,[col2]:5,[col3]:10)

\$csi([23456]:0,[65432]:5,[56789]:10)

This formula calculates a CSI score, or a customer service index. It's used in surveys with questions like: *Rate your stay at our hotel from 1-5.*

The formula itself works the same way as the *\$wqs()* formula.

Continued on next page...

...Dynamic Functions Continued

\$fix(fix value)

\$fix(75)

This function fixes the value for each row of the calculated column to the value you put between the brackets. This value is static and will not accumulate. This is very useful as a target column if your target doesn't accumulate, e.g. if it's a percentage or if the column you are targeting is dynamic – an NPS column for example.

\$sra([col1], num_of_periods)

\$sra([23456],7)

This function creates a dynamic rolling average. As opposed to the regular rolling average this works well with categorised data as it dynamically adapts across chart filters and different date groupings. The formula is the same, apart from the \$ sign in front.

\$agedays(), \$agemonths(), \$ageyears()

These three formulae are very similar to each other, and are used just as they are above. *\$agedays* will count the number of full calendar days that have passed since the date of any row in your data. *\$agemonths* will count the number of completed months and *\$ageyears* will count the number of years.

\$var([column to check])

\$var([25674])

This formula calculates the variance between two consecutive values in the same column. For example, it can calculate the

difference in a value month on month, week on week or day on day depending on how your data is structured.

Since it is a dynamic calculation, it will still work and be accurate if your data is filtered or grouped.

We're here to Help

Should you have any questions about these formulae, dynamic functions or how they work at all, please let us know. The support team here at Target Dashboard are always here to help you, and will be happy to do so.

We can always be reached on support@targetdashboard.com.