

Function	Description
<b>Text Functions</b>	
<b>concatenate()</b>	The concatenate() function is used to combine multiple text values into one. You can reference field names or text inside quotes. For example, to combine First Name, a space and Last Name into a Full Name calculated text field, you can use the following formula:
concatenate({Field A}, " ", {Field B}, " ", {Field C})	concatenate({First Name}, " ", {Last Name})
<b>contains()</b>	The contains() function will return a 1 (one) if the "text" is found in Field A and a 0 (zero) if it does not contain the "text". For example, to determine if an email address was part of a phrase: contains("@", "You can reach them at example@trackvia.com.") would return 1 because the 'at' symbol is within the phrase.
contains("text", {Field A})	The contains() function can also be used in conjunction with the if() function to evaluate if a statement is true. For example, the formula if(contains("urgent", {Message}), "yes", "no") will return the word "yes" if the Message field contains the word urgent, and "no" if does not.
<b>mid()</b>	The mid() function gives you the ability to return a specific set of characters from another field. The parameters require stating what number of character to start at (1 is the beginning) and how many characters you want to return. This is commonly used when creating a unique value (or key). For example, with this function you can create an ID that contains text from another field that appends to an auto-counter field (unique number). In other words, you can combine a portion of text from a single line field along with a number field using the mid() function within the concatenate() function (see above).
mid({Field A}, Number of start character, Number of characters to return)	So if you have a Single Line field labeled "Field A" and want to grab the first 3 letters, the formula would look like:  mid({Field A}, 1, 3)  Then to combine that with an Auto-counter field labeled "ID Number", you can use the following:  concatenate(mid({Field A}, 1, 3), {ID Number})
<b>snip()</b>	The snip() function is useful for extracting a section of text out of longer strings of text that have a predictable format. This is done by specifying a character to start at within the text and then choosing a character to end the snip. For example, if you have a field that contains a person's full name and you need a field that only displays the person's first name, you can snip the first name into a Calculated Text field. Here is an example:
snip({Field A}, "start text", "end text")	snip({Full Name}, "", " ")  To start at the beginning of a field's value, you will need to utilize a pair of quotes with nothing in between. In this scenario, the end character is a space (the space between a person's first and last name). <i>*Note: this formula will only return the first word (two word first names will not bring over both words). There currently is not a way to specify the end of a string of text - an actual character is required.</i>
<b>substitute()</b>	The substitute() function allows you to identify a character (or multiple) in a value and substitute it with another value. For example, if you have a paragraph field that contains the "&" symbol numerous times and you would like to change this to "and", you can use the following formula to solve for this:
substitute({Field A}, "text1", "text2")	substitute({Field A}, "&", "and")
<b>left()</b>	The left() function is similar to the mid() function, except you do not need to specify a starting point. You can return a specific number of characters from the beginning of a text string (the left side). This requires entering the field name and the number of characters to return in a calculated field. One example would be to return the first 5 digits of a long number. The formula would look like:
left({Field A}, number of characters to return)	left({Long Number}, 5)
<b>right()</b>	Similar to the left() function, the right() function allows you to return a specific number of characters from the beginning of a text string (the right side). For example, if you need to return the last 4 digits of a large number, you can use the following formula:
right({Field A}, number of characters to return)	right({Large Number}, 4)
<b>upper()</b>	The upper() function is used to display text values in all uppercase letters in a calculated text field. For example, if you have field labeled "Country" and someone enters "usa" in this field, you can use the following formula to return "USA":
upper({Field A})	upper({Country})
<b>lower()</b>	The lower() functions allows you to display text in all lowercase letters. This is great for making text uniform across all records or when concatenating into a 'code' that requires all lowercase letters. Here is an example with the lower function being used in conjunction with another field that contains a number:
lower({Field A})	lower({Code Prefix}) & {ID Number}
<b>char_length()</b>	The char_length() function is used to determine the number of characters in a field. For example, if you have a text field with the string "Working with formulas!", you can use the following formula:
char_length({Field A})	char_length({Text Field})
<b>length()</b>	Similar to char_length(), the length() function the displays the length of a field, measured in bytes, using the following formula:
length({Field A})	length({Text Field})  In this case, if the value of the text field contains "Español", using the length() function would return 8. However, if you were to use char_length(), the result would be 7. While standard English characters typically only take up one byte, non-English characters can take up multiple bytes.
<b>rpad()</b>	The rpad() function is used to ensure all results have the same number of characters, by adding or removing characters to the right of a field's value, where N specifies the number of characters to return. Below is an example of using rpad() on a text field in which the value is "TrackVia":
rpad({Field A}, N, "Character for padding")	Sample Formulas: rpad({Text Field}, 10, "**") returns <i>TrackVia**</i> rpad({Text Field}, 5, "**") returns <i>ckVia</i>
<b>lpad()</b>	Similar to rpad(), lpad() will add or remove characters to the left of a field's value. Using lpad() on a field containing the value of "TrackVia" would look like the following:
lpad({Field A}, N, "Character for padding")	lpad({Text Field}, 10, "**") returns <i>**TrackVia</i> lpad({Text Field}, 5, "**") returns <i>Track</i>
<b>reverse()</b>	Reverses the characters from a specified field. For a field with the value of "TrackVia", the formula would look like the following:
reverse({Field A})	reverse({Text Field}) returns <i>aIVkarT</i>
<b>trim()</b>	Removes any leading or trailing spaces from the value of a field.
trim({Field A})	trim({Text Field})

Number Functions	
<b>abs()</b>	Absolute value of a number
abs(number)	abs(-1) returns 1 abs(1) returns 1
<b>exp()</b>	Exponential of x (e to the power of x)
exp(x)	exp(1) returns 2.718...
<b>ln()</b>	Natural logarithm of x (base e)
ln(x)	ln(2) returns 0.693...
<b>log()</b>	Logarithm of x (specified base)
log(base or x, x)	log(3, 2) returns 1.584...
<b>log10()</b>	Common logarithm of x (base 10)
log10(x)	log10(2) returns 0.301...
<b>mod()</b>	Returns the remainder after dividing number by modulus
mod(number, modulus)	mod(7, 2) returns 1 mod(6, 2) returns 0
<b>power()</b>	x to the power of y
power(x, y)	power(2, 2) returns 4
<b>rand()</b>	Dynamically returns a random number between 0 and 1
rand()	
<b>round()</b>	Rounds a number to given digits after decimal point
round((Number))	round(4/3, 3) returns 1.333 round(1.6772, 2) returns 1.68
<b>sign()</b>	Returns 0 if x is blank or zero, -1 if x is negative, and 1 if x is positive
sign(x)	sign(1) returns "positive" sign(0) returns "0" sign(-1) returns "negative"
<b>sqrt()</b>	Square root of x
sqrt(x)	sqrt(64) returns 8 sqrt(12) returns 3.464...
<b>sin()</b>	Sine of x
sin(x)	sin(2) returns 0.909...
<b>cos()</b>	Cosine of x
cos(x)	cos(2) returns -0.416...
<b>tan()</b>	Tangent of x
tan(x)	tan(2) returns -2.185...
<b>asin()</b>	Inverse Sine of a number
asin(number)	asin(1) returns 1.570...
<b>acos()</b>	Inverse Cosine of a number
acos(number)	acos(1) returns 0
<b>atan()</b>	Inverse Tangent of a number
atan(number)	atan(1) returns 0.785...
<b>atan2()</b>	Arc tangent of the two variables x and y
atan2(x, y)	atan2(1, 2) returns 0.463...
<b>degrees()</b>	Number of degrees in an angle of x radians
degrees(radians)	degrees( $\pi/2$ ) returns 90
<b>radians()</b>	Number of radians in an angle of x degrees
radians(degrees)	radians(90) returns $\pi/2$ (1.570...)
<b>pi()</b>	The constant pi, 3.14159... (takes no inputs)
pi()	
<b>ceiling()</b>	Rounds up to the next integer.
ceiling(x)	ceiling(6.1) returns 7.
<b>floor()</b>	Rounds down to the closest integer.
floor(x)	floor(6.1) returns 6.

Date Functions	
<p><b>now()</b></p> <p><b>year()</b></p> <p><b>month()</b></p> <p><b>day()</b></p> <p><b>weekday()</b></p>	<p>The now() function can be entered on its own in a Calculated Date (or Text) field to show the current date (and time in Text) or it can be used to reference the current date in a formula. In a Calculated Date field, it will display "Today".</p> <p>You can return the year of a date by placing the name of a date field (or a specific date in quotes) within the year() function. This will return a 4 digit number. For example, to return someone's birth year, you can write year({Birthdate}) in a Calculated Number field.</p> <p>The month() and day() functions can be used in a similar manner. They will return numbers, i.e. "1" for "January" and "9" for the 9th day of the month.</p> <p>The weekday () function will return a number between 1 through 7 where Sunday is "1" and Saturday is a "7".</p> <p><i>Note: You cannot use these functions on their own in a Calculated Date field. The result is a number, which is not accepted on its own in a date field type.</i></p>
<p><b>dateadd()</b></p> <p><b>datesub()</b></p>	<p>You can use these functions in a Calculated Date field to add or subtract days, hours, etc. from another date in your table.</p> <p>Both functions require the syntax -- datesub({DateField}, n, "unit") -- where</p> <ul style="list-style-type: none"> <li>- DateField is a field of type Date or Date and Time</li> <li>- n is the number of units to add or subtract</li> <li>- unit is one of the following:</li> </ul> <p>"yyyy": Year  "q": Quarter  "m": Month  "d": Day  "ww": Week  "h": Hour  "n": Minute  "s": Second</p> <p>One example use case would be to determine an expected ship date for an order if you generally ship two days after the order date.  <i>Example formula</i> : dateadd({Order Date}, 2, "d")</p>
<p><b>datedif()</b></p>	<p>This function can be used to determine the number of months, days, hours, etc between two dates or date and times. You can reference date fields or static dates (i.e. "2013-01-01") using the syntax:</p> <p>datedif({End Date}, {Start Date}, "unit")</p> <p>Please see the section above for the available units.</p> <p><i>Note: If using this function alone, you would insert the function into a calculated text or calculated number field type. Because it returns a whole number, it will not be accepted into a date field type.</i></p>
<p><b>convert_tz()</b></p> <p>convert_tz({Date and Time}, "Desired Time Zone")</p>	<p>This function will convert a date and time field from one time zone to another. Please reference our knowledge base for a complete list of available time zones.</p> <p>convert_tz({Start Time}, "America/Chicago")</p> <p><i>Note: If using this function alone, you would insert the function into a calculated text or calculated number field type. Because it returns a whole number, it will not be accepted into a date field type.</i></p>
<p><b>dayname()</b></p> <p>dayname({Date})</p>	<p>This function will display the day of the week (Sunday, Monday, etc) for a specified date field.</p>
<p><b>dayofweek()</b></p> <p>dayofweek({Date})</p>	<p>This function will show the numeric value (1-7) of the day of the week from a date field type. Sunday is assigned the default value of 1.</p>
<p><b>dayofyear()</b></p> <p>dayofyear({Date})</p>	<p>Expanding on dayofweek(), the dayofyear function outputs the numeric value of the date entered. The result will be between 1-366.</p>
<p><b>last_day()</b></p> <p>last_day({Date})</p>	<p>Returns the last of the month of a given date field.</p> <p>When the Date selected is June 6, 2015, last_day({Date}) returns June 30th, 2015.</p>
<p><b>hour()</b></p> <p>hour({Date and Time})</p>	<p>When applied date and time field type, this function displays the hour of the day as a numeric value between 0 and 23. This similar to how military time works with 0 being equal to 12AM.</p>
<p><b>monthname()</b></p> <p>monthname({Date})</p>	<p>Displays the name of the month from a date field.</p> <p>When the Date selected is June 6, 2015, monthname({Date}) returns June.</p>
<p><b>quarter()</b></p> <p>quarter({Date})</p>	<p>Returns the numeric value (1-4) of a date field.</p> <p>When the Date selected is June 6, 2015, quarter({Date}) returns 2</p>
<p><b>weekdays()</b></p> <p>weekdays({Start Date} , {End Date})</p>	<p>Returns the number of weekdays between a start and end date.</p> <p>weekdays(06/01/2015, 06/27/2015) returns 20.</p> <p><i>Note: If using this function alone, you would insert the function into a calculated text or calculated number field type. Because it returns a whole number, it will not be accepted into a date field type.</i></p>
<p><b>week()</b></p> <p>week({Date})</p> <p>week({Date}, [mode])</p>	<p>This function returns the week number of a date. You can reference date fields or static dates (i.e. "2013-01-01") using the syntax:</p> <p>week({DateField})</p> <p>This function can also specify the start date or day for the 1st week of the year. You can add a 2nd part to this function called the "mode"; i.e. week(date, [mode]). Please refer to our knowledge base for more information.</p> <p><i>Note: If using this function alone, you would insert the function into a calculated text or calculated number field type. Because it returns a whole number, it will not be accepted into a date field type.</i></p>
<p>The date a record is created or updated can be used in a formula by referencing the field name in all uppercase letters. Example:  dateadd({CREATED}, 1, "m") will add a month to the Created field that is included with every record to display the date and time the record was added to a table.</p>	

Child Functions	
<p><b>childconcatenate()</b></p> <p>childconcatenate({child field reference}, "delimiter")</p>	<p>The childconcatenate() function is used to combine multiple values from the linked records' field in a child table. After selecting the childconcatenate() function in the list of options presented in the Result Formula box, you will see there are two parts that need to be entered within the function: the reference to the child name (which you can also find in the list) and the delimiter (the character you would like to use to separate each child record's value), which needs to be entered within quotation marks.</p> <p>One example would be if you would like to see a comma-separated list of products that one of your clients has purchased where the Clients table is a parent of the Orders table. You can use the following formula to display this information in a calculated text field in the Clients table:</p> <p>childconcatenate({Orders}.Customer Link).Product, ", "</p>
<p><b>childconcatenatedistinct()</b></p> <p>childconcatenatedistinct({child field reference}, "delimiter")</p>	<p>This function works the same as the function above, except it will only return the unique values from the child table's field. For example, if the results from the formula above return "Dell, Mac, Samsung, Dell, HP", the childconcatenatedistinct() function will only return "Dell, Mac, Samsung, HP".</p>
<p><b>count()</b></p> <p>count({child field reference})</p> <p><b>countdistinct()</b></p> <p>countdistinct({child field reference})</p>	<p>The count() function requires choosing a reference to a field in the child table. For each record in the parent table, it will return the number of records linked where the field chosen is not blank. In order to return a count for every child record linked to a parent record, you can utilize the field that is linking the two tables together. Here is an example:</p> <p>count({Orders}.Customer Link).Customer Link)</p> <p>This function works the same as the function above, except it will only return the unique values from the child table's field.</p>
<p><b>sum()</b></p> <p>sum({child field reference})</p> <p><b>average()</b></p> <p>average({child field reference})</p>	<p>The sum() function allows you to add values from linked child records where each value in the field referenced is a number, currency or percent. For example, if you would like to display the total amount a customer has spent from multiple orders, you can use the following formula:</p> <p>sum({Orders}.Customer Link).Total Amount)</p> <p>Similar to the sum() function, the average() function is used to return the average value from linked child records where each value in the field referenced is a number, currency or percent. It sums the values and divides it by the number of linked child records. The following formula can be used to return the average amount a customer spends per order:</p> <p>average({Orders}.Customer Link).Total Amount)</p>
<p><b>max()</b></p> <p>max({child field reference})</p> <p><b>min()</b></p> <p>min({child field reference})</p> <p><b>stdev()</b></p> <p>stdev({child field reference})</p> <p><b>stdevp()</b></p> <p>stdevp({child field reference})</p>	<p>Returns the maximum value.</p> <p>max({Parent Table}.Link to Parent).Child Field)</p> <p>Returns the minimum value.</p> <p>min({Parent Table}.Link to Parent).Child Field)</p> <p>Returns the sample standard deviation.</p> <p>stdev({Parent Table}.Link to Parent).Number)</p> <p>Returns the population standard deviation.</p> <p>stdevp({Parent Table}.Link to Parent).Number)</p>
Logic Functions	
<p><b>if()</b></p> <p>if(condition, true result, false result)</p>	<p>Enter a condition, what should return in the field if that condition is true, followed by what should return if the condition is false.</p> <p>Example: if({Price} &gt; 50, "Expensive", "Inexpensive")</p> <p>If there are multiple conditions, you can nest if() functions within each other. Here's the syntax when nesting 3 if() functions (4 different results):</p> <p>if(condition, true result, if(condition, true result, if(condition, true result, false result)))</p> <p><i>Note: Do not forget the final false result needed when nesting if() functions. Also, a closed parenthesis is required at the end of the formula for each if() function used.</i></p>
<p><b>or()</b></p> <p>or(condition A, condition B, condition C)</p> <p><b>and()</b></p> <p>and(condition A, condition B)</p>	<p>When writing a formula that requires "or" logic (only one or more condition needs to be met), you can enter each within the or() function.</p> <p>Example: or({Price}=10, {Price}=20, {Price}=30)</p> <p>When writing a formula that requires multiple conditions to be met, you can enter each within the and() function.</p> <p>Example: and({City}="Denver", {State}="CO")</p>
<p><b>isblank()</b></p> <p>isblank({Field Name})</p>	<p>This function can be used in a logic formula when you need to determine if a field does not have a value (is blank or null).</p> <p>Example in an if() function: if(isblank({City}), "You need to enter a City", "")</p> <p><i>Note: An empty pair of quotation marks indicates to leave the field blank.</i></p>