

Variables, Operators & Functions

Last Modified on 05/27/2021 10:52 am MDT

A formula uses numeric data from a [select list](#), [numeric](#) or [date](#) field, another formula, or workflow state to perform a calculation. This code is comprised of the following components:

- **Variable:** A value on which the calculations are performed. Variables represent the values from a select list, numeric field, another formula, or workflow state. Because formulas use data that can change, all variables must be assigned a name (e.g. LIKELIHOOD for the Likelihood field) that allows the calculations to remain valid, even if the values change. If you're creating a variable using a field, formula, or state from relationship or reference object types, you must select a **variable sub-type** to specify how the data from multiple objects is compiled, calculated, and displayed.
- **Operator:** A character that represents an action. Operators are typically used to perform arithmetic actions (e.g. + for add or * for multiply), but they can also be used to group or compare variables. Operators can be used in formulas that contain a function and multiple operators can be added to a single formula.
- **Function:** A block of code that performs calculations on multiple values. For example, SUM(LOSS) calculates the total values from the Loss numeric field across multiple objects. Because functions require multiple values to complete the calculation, **Relationship** and **Reference** variable types are typically used with functions, however, it's possible to perform functions using fields or formulas added directly to the object type. Functions can be used in formulas that contain operators and multiple functions can be added to a single formula.



A formula with two variables and an operator.

Average Impact

MEAN(IMPACT)

- Variable
- Function

A formula with a variable and a function.

% of Samples Failed

(SUM(FAILED))/(SUM(TESTED))*100

- Variable
- Operator
- Function

A formula with two variables, four operators (grouping, division, and multiplication), and two functions.

The formulas, operators, and functions in this article are not an exhaustive list of available options. If your organization requires a more complex formula, contact [Resolver Support](#) for assistance.

Field & Formula Variables

Field and formula variables represent the values pulled from numeric fields, date fields, formulas, or select lists, which is the data on which the formula calculations are performed. When using fields or formulas in a **Relationship** or **Reference** variable, a variable **sub-type** must be selected to determine how the data from multiple objects will be compiled, calculated, and displayed. For information on workflow state variable sub-types, see the **Workflow State Variables** section below.

VARIABLES

Variable Type

Relationship ▾

Relationship

Issues ▾

Available Components

Due Date ▾

Sub Type

Array ▾

Array

Sum

Count

Average

Every

Max

Min

Name

DUEDATE

Description

Variable sub-types for a Relationship variable. The Array subtype is selected by default.

Note that the examples in the table below use VARIABLEA with a sample data set of 100, 9, 13, 4, 1. This data set represents the numeric values that would be collected from multiple objects.

SUB-TYPE NAME	DESCRIPTION	EXAMPLE
Array	Creates a set of values from the variable. To display these values in a formula, they must first be converted into a single value using a function (i.e. sum, min, max, or mean). This is the default subtype.	Selecting the Array sub-type for VARIABLEA will return the values from the variable on every object in the relationship/reference (e.g. 100, 9, 13, 4, 1). Before the data from VARIABLEA can be displayed or used within a larger formula, it must be aggregated into a single value using a function (e.g. SUM(VARIABLEA)).
Sum	Calculates a total from the variable's set of values and returns a single number. This subtype is not available for select list variables.	Selecting the Sum sub-type for VARIABLEA will return the sum of all values pulled from VARIABLEA on every object in the relationship/reference (e.g. 127).

Count	Returns the number of times a variable has been added to an object, thereby counting the number of objects in the relationship/reference.	Selecting the Count sub-type for VARIABLEA will return the number of times VARIABLEA appears on the objects in the relationship/reference (e.g. 5).
Average	Calculates an average number from the variable's set of values. This subtype is not available for select list variables.	Selecting the Average sub-type for VARIABLEA will return the sum of all numeric values in VARIABLEA, divided by the number of objects in the relationship/reference (e.g. $(100 + 9 + 13 + 4 + 1) / 5 = 25$).
Every	Checks if the variable contains a value on the objects in the relationship/reference. If all objects contain <i>any</i> value, a 1 (representing a true result) is returned. If some or all of the objects are missing a value, a 0 (representing a false result) is returned. This subtype does not check for specific values and is typically used in a workflow condition to control if an object can move into the next state.	Selecting the Every sub-type for VARIABLEA will check if VARIABLEA contains a value on all the objects in the relationship/reference. If all the objects contain a value, the result is 1 (true). If some or all of the objects are missing a value, the result is 0 (false).
Max	Calculates the highest number from the variable's set of values. This subtype is not available for select list variables.	Selecting the Max sub-type for VARIABLEA will return the highest number from the set of values (e.g. 100).
Min	Calculates the lowest number from the variable's set of values. This subtype is not available for select list variables.	Selecting the Min sub-type for VARIABLEA will return the lowest number from the set of values (e.g. 1).

Workflow State Variables

Workflow state variables represent the current state of one or more objects in a **Relationship** or **Reference** variable type using states from the object type's workflow or any related assessment workflows. These variables require a **sub-type** to determine how the state data will be compiled, calculated, and displayed. Both library and assessment workflow states can be selected in these variable types.

VARIABLES

Variable Type **Relationship**

Relationship Issues

Available Components

Closed

Sub Type

All

Name

CLOSEDISSUE

Description

A workflow state variable type. The All sub-type is selected by default.

Note that the examples in the table below use CLOSEDISSUE as the variable, which represents the Closed workflow state of objects in the Issues relationship.

SUB-TYPE NAME	DESCRIPTION	EXAMPLE
All	Checks if all objects in the relationship or reference are in the state selected in the Available Components dropdown menu. If all the objects are in the specified state, a 1 (representing a true result) is returned. If some or all of the objects aren't in the specified state, a 0 (representing a false result) is returned. This is the default sub-type for workflow state variables.	Selecting the All sub-type for the CLOSEDISSUE variable will check if all the objects in the Issues relationship are in the Closed state. If all objects are in the Closed state, the result is 1 (true). If some or all of the objects are not in the Closed state, the result is 0 (false).

Any	Checks if any of the objects in the relationship or reference are in the state selected in the Available Components dropdown menu. If some or all of the objects are in the specified state, a 1 (representing a true result) is returned. If none of the objects are in the specified state, a 0 (representing a false result) is returned.	Selecting the Any sub-type for the CLOSEDISSUE variable will check if any of the objects in the Issues relationship are in the Closed state. If some or all of the objects are in the Closed state, the result is 1 (true). If none of the objects are in the Closed state, the result is 0 (false).
Count	Returns the number of objects in the relationship or reference that are currently in the state selected in the Available Components dropdown menu.	Selecting the Count sub-type for the CLOSEDISSUE variable will return the number of objects in the Issues relationship that are currently in the Closed state (e.g. 12).

Operators

Operators are used to perform basic calculations, compare, assign, or group data, or compare data to return a true or false (Boolean) result.

Examples in the table below that include an ellipsis (...) indicate that the operator and its components cannot be used as a standalone formula and must be part of a larger formula.

OPERATOR	NAME	DESCRIPTION	EXAMPLE	RESULT
()	Grouping	Groups variables and operators within a formula to create a single value.	2 * (3 + 4) ...	14
+	Add		4 + 5	9
-	Subtract		7 - 3	4
*	Multiply		2 * 3	6
/	Divide		6 / 2	3
		Checks if the variable's		

==	Equal to	value matches the value entered following the == symbol. If so, the result is true .	X == 4 ...	If X has a value of 4, the result is true .
!=	Unequal	Checks if the values of two variables are different. If they're different, the result is true .	2 != 3	True
?:	Conditional expression	Checks if the value of a variable is true or false . If true , the result is the value entered after the ? symbol. If false , the result is the value entered after the : symbol.	15 > 100 ? 1 : -1	-1
<	Less than	Checks if the value of the left variable is less than the value of the right variable. If yes, the result is true .	2 < 3	True
>	Greater than	Checks if the value of the left variable is greater than the value of the right variable. If yes, the result is true .	2 > 3	False
<=	Less than or equal to	Checks if the value of the left variable is less than or equal to the value of the right variable. If yes, the result is true .	4 <= 3	False
>=	Greater than or equal to	Checks if the value of the left variable is greater than or equal to the value of the right variable. If yes, the result is true .	2 + 4 >= 6	True

and	Logical <i>and</i>	If both variables are true , the result is true . If either or both variables are false , the result is false .	X == 5 AND Y == 5 ...	If both X and Y have a value of 5, the result is true . If either X or Y has a value other than 5, the result is false .
or	Logical <i>or</i>	If either or both of the variables are true , the result is true . If one or both of the variables is false , the result is false .	X == 4 OR Y == 3 ...	If Y has a value of 3, the result is true . If X is false , but Y is 3, the result is true . Otherwise, the result is false .

Functions

Functions perform a task or calculate a value. Because functions require multiple values to complete the calculation, **Relationship** and **Reference** variable types are typically used with functions; however, it's possible to perform functions using fields or formulas added directly to the object type.

NAME	DESCRIPTION	EXAMPLE	RESULT
Max	Calculates the maximum value from a variable.	max(A)	The highest numeric value of the variable A (e.g. 8).
Mean	Calculates the mean value from a variable.	mean(A)	The sum of all numeric values in variable A, divided by the amount of numbers in the set (e.g. $5 + 8 + 4 + 4 / 4 = 5.25$).
Min	Calculates the minimum value from a variable.	min(A)	The lowest numeric value of variable A (e.g. 1).
Sum	Calculates the total value	sum(A)	The total of all numeric values in variable A (e.g.

	from a variable.		15).
timeDiff	Compares the days, months, or seconds between variables. This function requires date-related data from two Date & Time fields or a Date & Time field and the today() function. See the Time Functions article for more information.	timeDiff(A,B,"days")	The difference in days between variables A and B (e.g. 5 days).
timeOffset	Offsets a variable by number of seconds, days, or months. This function requires a Date & Time field variable. This function returns the offset date in Unix timestamp which will need to be converted manually. See the Time Functions article for more information.	timeOffset(A,1,"days")	Offsets the date in variable A by one day.
today	Returns the current date. This function does not accept any parameters (variables inside the parentheses), however, it can be used as parameters inside the timeDiff function.	timeDiff(A,today(),"days")	The difference in days between variable A and today's date.

Examples

For a list of common formulas, see the [Formula Examples](#) article.