

FUNCTION	SYNTAX	EXAMPLE	RETURN TYPE	DESCRIPTION
<b>Date</b>				
AS_OF_DATE	AS_OF_DATE()	DDIFF((First Payment Date),AS_OF_DATE())	Date	Returns the date selected in the As Of section in the Save Tape banner
DATE	DATE(text)	DATE("2018-02-28")	Date	Converts text to a date type
DAY	DAY(date)	DAY(DATE("2018-02-28"))	Numeric	Extracts the day of the date argument ranging from 1 to 31
DDIFF	DDIFF((date),(date))	DDIFF((First Payment Date),(Next Payment Date))	Date	Finds difference in days between the dates
DMINUS	DMINUS((date),1)	DMINUS((Paid Through Date),1)	Date	Subtracts the specified number of days to a date field
DPLUS	DPLUS((date),1)	DPLUS((Paid Through Date),1)	Date	Adds the specified number of days to a date field
MDIFF	MDIFF((date),(date))	MDIFF((First Payment Date),(Maturity Date))	Date	Finds difference in months between the dates
MMINUS	MMINUS((date),1)	MMINUS((Paid Through Date),1)	Date	Subtracts the specified number of months to a date field
MONTH	MONTH(date)	MONTH(DATE("2018-02-28"))	Numeric	Extracts the month of the date argument ranging from 1 to 12
MPLUS	MPLUS((date),1)	MPLUS((Paid Through Date),1)	Date	Adds the specified number of months to a date field
NULL_DATE	NULL_DATE	IF((Current Balance)>0,(Next Due Date),NULL_DATE)	Date	Returns a null value for a date field
YDIFF	YDIFF((date),(date))	YDIFF((First Payment Date),(Maturity Date))	Date	Finds difference in years between the dates
YEAR	YEAR(date)	YEAR(date)	Numeric	Extracts the year of the date argument
YMINUS	YMINUS((date),1)	YMINUS((Paid Through Date),1)	Date	Subtracts the specified number of years to a date field
YPLUS	YPLUS((date),1)	YPLUS((Paid Through Date),1)	Date	Adds the specified number of years to a date field
<b>Logical</b>				
AND	AND(logical_expression_1, logical_expression_2)	AND(EQUAL((DTI), 0), EQUAL((LTV),0))	Boolean	Tests a number of user-defined conditions and returns TRUE if ALL of the conditions evaluate to TRUE, or FALSE otherwise
BETWEEN	BETWEEN(value1, Boundary1, Boundary2)	BETWEEN((DTI), 0, 100)	Boolean	The formula returns TRUE if the values are between two boundaries, excluding the boundary values
CASE	CASE(WHEN(),WHEN(),ELSE())	CASE(WHEN((Borrower Original FICO) > 700, "Yes"),ELSE("No"))	Generic	Used for structuring operations with multiple logical branches. Must have one or more WHEN clauses and at most one ELSE clause.
WHEN	WHEN(conditional,result)	(WHEN((Borrower Original FICO) > 800, "Excellent")	Generic	Returns the result of the first satisfied WHEN clause, otherwise returns the result of the ELSE clause
ELSE	ELSE(result)	ELSE(NULL_DATE)	Generic	
EQUAL TO	EQUAL(value1, valueEqualTo)	EQUAL((DTI), 0)	Boolean	The formula returns TRUE if a value in cell A1 is equal to the values in cell B1; FALSE otherwise
GREATER THAN	GT(value1, valueGreaterThan)	GT((DTI), 100)	Boolean	The formula returns TRUE if a value in cell A1 is greater than a value in cell B1; otherwise it returns FALSE
GREATER THAN OR EQUAL TO	GTE(value1, valueGreaterThanOrEqualTo)	GTE((DTI), 100)	Boolean	The formula returns TRUE if a value in cell A1 is greater than or equal to the values in cell B1; FALSE otherwise
IF	IF(logical_expression, value_if_true, value_if_false)	IF(LT((DTI), 1), MULTIPLY((DTI), 100), (DTI))	Generic	Tests a user-defined condition and returns one result if the condition is TRUE, and another result if the condition is FALSE
IN	IN(value, collection_of_values)	IN((Property State),{"CT","MA","NH","VT"})	Boolean	Checks if first argument is equal to one of the values from the collection in the second argument
IS_EMPTY	IF(IS_EMPTY((Field)), (Replace with), (or else))	IF(IS_EMPTY((Modified Date)), "No", "Modified")	Boolean	Returns true if the field value is empty, returns false there is a value
LESS THAN	LT(value1, valueLessThan)	LT((DTI), 100)	Boolean	The formula returns TRUE if a value in cell A1 is less than in cell B1; FALSE otherwise
LESS THAN OR EQUAL TO	LTE(value1, valueLessThanOrEqualTo)	LTE((DTI), 100)	Boolean	The formula returns TRUE if a value in cell A1 is less than or equal to the values in cell B1; FALSE otherwise
NOT	NOT(logical_expression)	NOT(EQUAL((DTI), 0))	Boolean	Returns a logical value that is the opposite of a user supplied logical value or expression
OR	OR(logical_expression_1, logical_expression_2)	OR(EQUAL((DTI), 0), EQUAL((LTV),0))	Boolean	Tests a number of user-defined conditions and returns TRUE if ANY of the conditions evaluate to TRUE, or FALSE otherwise
<b>Math</b>				
ABS	ABS(value)	ABS(-3)	Numeric	Returns the absolute value of a number
ADD	value1+value2 or ADD(value1,value2)	{Original Balance}+100 or ADD((Original Balance),100)	Numeric	Returns the sum of two numbers
DIVIDE	value1/value2 or DIVIDE(value1,value2)	{Original Balance}/100 or DIVIDE((Original Balance),100)	Numeric	Returns one number divided by another
MODULO	MOD(value1,value2)	MOD((Rate Frequency),(Loan Age))	Numeric	Returns remainder from division
MULTIPLY	value1*value2 or MULTIPLY(value1,value2)	{Original Balance}*100 or MULTIPLY((Original Balance),100)	Numeric	Returns the product of two numbers
NULL_NUMBER	NULL_NUMBER	IF((Original FICO<300,NULL_NUMBER,(Original FICO))	Numeric	Returns a null value
POWER	POWER(value1,value2)	POWER((1+(Original Gross Rate)), (Original Term))-1	Numeric	Returns the first argument raised to the second argument
ROUND	ROUND(value,number_of_decimals)	ROUND((Original Balance),2)	Numeric	Round the value to the specified decimal place
SUBTRACT	value1-value2 or SUBTRACT(value1,value2)	{Original Balance}-100 or SUBTRACT((Original Balance),100)	Numeric	Returns the difference of two numbers
<b>Statistical</b>				
COUNTMATCH	COUNTMATCH(match, source)	COUNTMATCH("0",{12 Month Paystring})	Integer	Counts the number of occurrences with an argument
MAX	MAX({Integer}, {Integer})	MAX({Appraisal Value}, {Purchase Price})	Integer	Maximum value between two arguments
MIN	MIN({Integer}, {Integer})	MIN({Appraisal Value}, {Purchase Price})	Integer	Minimum value between two arguments
NZMAX	NZMAX({Integer}, {Integer})	NZMAX({Integer}, {Integer})	Integer	Non-Zero maximum value between arguments
NZMIN	NZMIN({Integer}, {Integer})	NZMIN({Integer}, {Integer})	Integer	Non-Zero minimum value between arguments
<b>Text</b>				
CONCAT	CONCAT(values, with)	CONCAT({Property Street}, {Property City})	Text	Joins together two or more text strings
CONTAINS	CONTAINS(string, "valueToContain")	CONTAINS({Property City}, "NEW")	Boolean	Returns if a column contains specific text
ENDS WITH	ENDS_WITH(string, "valueToEndWith")	ENDS_WITH({Property City}, "B")	Boolean	Returns if a column ends with specific text
LEFT	LEFT(string, numberOfCharacters)	LEFT({Property City}, 1)	Text	Returns a specified number of characters from the start of a supplied text string
LENGTH	LENGTH(string)	LENGTH({Property City})	Integer	Returns the length of a supplied text string
LOWERCASE	LOWERCASE(string)	LOWERCASE({Property City})	Text	Converts all letters in a text string to lowercase
REMOVE	REMOVE(string, "valueToRemove")	REMOVE({Property City}, "-")	Text	Removes "valueToRemove" if it is a match
REPLACE	REPLACE(string, "valueToReplace", "newValue")	REPLACE({Property City}, "-", "-")	Text	Replaces all or part of a text string with another string (from a user supplied position)
RIGHT	RIGHT(string, numberOfCharacters)	RIGHT({Property City}, 1)	Text	Returns a specified number of characters from the end of a supplied text string
STARTS WITH	STARTS_WITH(string, "valueToStartWith")	STARTS_WITH({Property City}, "A")	Boolean	Returns if a column starts with specific text
SUBSTRING	SUBSTRING(string, startIndex, endIndex)	SUBSTRING({Property City}, 0, 5)	Text	Returns the substring starting at start index up until end index (inclusive)
TEXT	TEXT(value)	TEXT({Property City})	Text	Converts a supplied value into text, using a user-specified format
TRIM	TRIM(string)	TRIM({Property City})	Text	Removes duplicate spaces, and spaces at the start and end of a text string
UPPERCASE	UPPERCASE(string)	UPPERCASE({Property City})	Text	Converts all letters in a text string to uppercase