String Functions

Tableau Desktop Reference Guide

String functions manipulate string data. String functions are useful for harvesting meaningful information found within strings and for improving the presentation of your data.

Parse out portions of a string using functions such as **FIND** and **FINDNTH** to find the position of delimiters or spaces and then using the **LEFT**, **RIGHT**, or **MID** functions to return or replace parts of the given string. Alternatively, use the **SPLIT** function when possible.

Some string functions use an optional starting position argument so the action is performed starting at the position given. The first character in the string is position 1. Other string functions use an optional length argument. If the length is not given, the action is performed for the entirety of the string.

Pattern Matching

Functions such as a **REGEXP_REPLACE** work with a regular expression, also called a rational expression, and can be used for pattern matching, similar to a "find and replace" command. This pattern matching syntax conforms to the International Components for Unicode (ICU) standard, and ranges from precise equality to general similarity using wildcard characters.

These functions are useful when dealing with strings that follow a particular pattern, such as phone numbers, email names, or part numbers. You can use these functions to isolate, replace, or test the validity of parts of these strings.

Examples

These are some examples of common string functions:

Function Syntax	Purpose	Example
LEFT(string, number) **RIGHT** and **MID** are similar	Returns the leftmost number of characters in a string.	**LEFT**(**[Postal Code]**, 2) returns "98" if the **[Postal Code]** is "98103"
SPLIT(string, delimiter, token number)	Returns a substring from a string, using a delimiter to divide the string into substrings and then using the token to determine which substring to return.	**SPLIT**(**[Phone]**, "-", 2) returns "633" if the value of **[Phone]** is "206-633-3400" since "633" is the 2nd token substring delimited by a hyphen.
CONTAINS(string, substring) **ENDSWITH** and **STARTSWITH** are similar	Returns true if the given string contains the specified substring, otherwise returns false.	**CONTAINS**(**[Product Name]**, "Phone") returns true if **[Product Name]** is "Apple iPhone" and false if **[Product Name]** is "Apple iPad"
FIND(string, substring, [start]) **FINDNTH** is similar	Returns the position of the substring within the given string, or 0 if the substring is not found.	**FIND**(**[Customer Name]**, ",") returns 4 if the **[Customer Name]** is "Doe, Jane" since that is the position of the comma delimiter.

MIN(expression) or **MIN** (expr1, expr2) **MAX** is similar	Returns the minimum of a single expression across all records, or the minimum of the two expressions for each record. For strings, the result is based on alphabetical order.	**MIN**(**[Vendor Name]**) returns "AAA Towing" if that is the first alphabetical value for **[Vendor Name]**.
REGEXP_REPLACE(string, pattern, replacement string) **REGEXP_EXTRACT**, **REGEXP_EXTRACTNTH**, and **REGEXP_MATCH** are similar.	Returns a string where the matching pattern is substituted with the replacement string.	**REGEXP_REPLACE**(**[Part ID]**, "\s", "-") returns "abc-123" if the **[Part ID]** is "abc 123" since the "\s" matches a white space character, which will be replaced by a hyphen, "-".

Many other common string functions are available and include:

Function	Purpose
REPLACE, **SPACE**	Return or replace parts of the given string.
ASCII, **CHAR**	Convert between ASCII codes and characters.
LOWER, **UPPER**	Update the string to upper or lowercase characters.
ISDATE	Return true if the string is a valid date.