

STRINGLESSTHAN Function

Returns `true` if the first string evaluates to be less than the second string, based on a set of common collation rules.

Source values can be string literals, column references, or expressions that evaluate to strings.

Basic Usage

String literal reference example:

```
derive type:single value:STRINGLESSTHAN('a','b') as:'stringCompare'
```

Output: Generates `true` in the new column `stringCompare`, since `a` evaluates to be less than `b`.

String literal reference example:

```
derive type:single value:STRINGLESSTHAN('a','ab') as:'stringCompare'
```

Output: Generates `true` in the new column `stringCompare`, since the second string contains an additional letter.

String literal reference example:

```
derive type:single value:STRINGLESSTHAN('abc','x') as:'stringCompare'
```

Output: Generates `true` in the new column `stringCompare`, since the first letter of the first string is less than the first letter of the second string.

Column reference example:

```
derive type:single value:STRINGLESSTHAN(string1,string2) as:'stringCompare'
```

Output: Generates a new `stringCompare` column containing the evaluation of `string1` column values being greater than `string2` column values.

Collation Rules

Collation refers to the organizing of written content into a standardized order. String comparison functions utilize collation rules for Latin. A summary of the rules:

- Comparisons are case-sensitive.
 - Uppercase letters are greater than lowercase versions of the same letter.
 - However, lowercase letters that are later in the alphabet are greater than the uppercase version of the previous letter.
- Two strings are equal if they match identically.
 - If two strings are identical except that the second string contains one additional character at the end, the second string is greater.
- A **normalized version** of a letter is the unaccented, lowercase version of the letter. In string comparison, it is the lowest value of all of its variants.
 - `a` is less than `á`.
 - However, when compared to `b`, `a` = `á`.
 - The set of Latin normalized characters contains more than 26 characters.

This table illustrates some generalized rules of Latin collation.

Order	Description	Lesser Example	Greater Example
1	whitespace	(space)	(return)
2	Punctuation	'	@
3	Digits	1	2
4	Letters	a	A
5		A	b

Resources:

NOTE: In the following set of charts (linked below), the values at the top of the page are lower than the values listed lower on the page. Similarly, the charts listed in the left nav bar are listed in ascending order.

For more information on the applicable collation rules, see <http://www.unicode.org/charts/collation/>.

Syntax and Arguments

```
derive type:single value:STRINGLESSTHAN(string_ref1,string_ref2)
```

Argument	Required?	Data Type	Description
string_ref1	Y	string	Name of first column or first string literal to apply to the function
string_ref2	Y	string	Name of second column or second string literal to apply to the function

For more information on syntax standards, see *Language Documentation Syntax Notes*.

string_ref1, string_ref2

String literal, column reference, or expression whose elements you want to compare based on this function.

Usage Notes:

Required?	Data Type	Example Value
Yes	String literal, column reference, or expression evaluating to a string	myString1, myString2

Examples

Tip: For additional examples, see *Common Tasks*.

Example - Simple string comparisons

The following example demonstrates functions that can be used to compare two sets of strings. These functions include the following:

- **STRINGGREATERTHAN** - Evaluates to `true` if the first string is greater than the second string. See *STRINGGREATERTHAN Function*.

- `STRINGGREATERTHANEQUAL` - Evaluates to `true` if the first string is greater than or equal to the second string. See *STRINGGREATERTHANEQUAL Function*.
- `STRINGLESSTHAN` - Evaluates to `true` if the first string is less than the second string. See *STRINGLESSTHAN Function*.
- `STRINGLESSTHANEQUAL` - Evaluates to `true` if the first string is less than or equal to the second string. See *STRINGLESSTHANEQUAL Function*.
- `EXACT` - Evaluates to `true` if the first string is an exact match with the second string. See *EXACT Function*.

Source:

The following table contains some example strings to be compared.

rowId	stringA	stringB
1	a	a
2	a	A
3	a	b
4	a	1
5	a	;
6	;	1
7	a	a
8	a	aa
9	abc	x

Note that in row #6, `stringB` begins with a space character.

Transform:

For each set of strings, the following functions are applied to generate a new column containing the results of the comparison.

```
derive type:single value: STRINGGREATERTHAN(stringA,stringB) as: 'greaterThan'
```

```
derive type:single value: STRINGGREATERTHANEQUAL(stringA,stringB) as: 'greaterThanEqual'
```

```
derive type:single value: STRINGLESSTHAN(stringA,stringB) as: 'lessThan'
```

```
derive type:single value: STRINGLESSTHANEQUAL(stringA,stringB) as: 'lessThanEqual'
```

```
derive type:single value: EXACT(stringA,stringB) as: 'exactEqual'
```

Results:

In the following table, the `Notes` column has been added manually.

rowId	stringA	stringB	lessThanEqual	lessThan	greaterThanEqual	greaterThan	exactEqual	Notes
-------	---------	---------	---------------	----------	------------------	-------------	------------	-------

1	a	a	true	false	true	false	true	Evaluation of differences between STRINGLES, STHAN and STRINGGREATERTHAN and greater than versions.
2	a	A	true	true	false	false	false	Comparisons are case-sensitive. Uppercase letters are greater than lowercase letters.
3	a	b	true	true	false	false	false	Letters later in the alphabet (b) are greater than earlier letters (a).
4	a	1	false	false	true	true	false	Letters (a) are greater than digits (1).
5	a	;	false	false	true	true	false	Letters (a) are greater than non-alphanumerics (;).
6	;	1	true	true	false	false	false	Digits (1) are greater than non-alphanumerics (;). Therefore, the following characters are listed in order of evaluation: <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;">Aa1;</div>
7	a	a	false	false	true	true	false	Letters (and any non-breaking character) are greater than space values.
8	a	aa	true	true	false	false	false	The second string is greater, since it contains one additional string at the end.

9	abc	x	true	true	false	false	false	The second string is greater, since its first letter is greater than the first letter of the first string.
---	-----	---	------	------	-------	-------	-------	------------------------------------------------------------------------------------------------------------