

# EXACT Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *string\_ref1, string\_ref2*
  - *ignore\_case*
- *Examples*
  - *Example - Simple string comparisons*

---

Returns `true` if the second string evaluates to be an exact match of the first string. Source values can be string literals, column references, or expressions that evaluate to strings.

**Wrangle vs. SQL:** This function is part of Wrangle, a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

## Basic Usage

### String literal reference example:

```
exact('a', 'a')
```

**Output:** Returns `true`, since the values are identical.

### String literal reference example:

```
exact('a', 'A')
```

**Output:** Returns `false`, since the capitalization is different between the two strings.

### Column reference example:

```
exact(string1, string2)
```

**Output:** Returns the evaluation of `string1` column values being exact matches with the corresponding `string2` column values.

## Syntax and Arguments

```
exact(string_ref1, string_ref2 [, ignore_case])
```

Argument	Required?	Data Type	Description
<code>string_ref1</code>	Y	string	Name of first column or first string literal to apply to the function
<code>string_ref2</code>	Y	string	Name of second column or second string literal to apply to the function
<code>ignore_case</code>	N	string	When <code>true</code> , matching is case-insensitive. Default is <code>false</code> .

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

## ignore\_case

When `true`, matches are case-insensitive. Default is `false`.

**NOTE:** This argument is not required. By default, matches are case-sensitive.

Required?	Data Type	Example Value
No	String literal evaluating to a Boolean	'true'

## 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.

**Transformation:**

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

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	STRINGGREATERTHAN( <code>stringA</code> , <code>stringB</code> )
<b>Parameter: New column name</b>	'greaterThan'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	STRINGGREATERTHANEQUAL( <code>stringA</code> , <code>stringB</code> )
<b>Parameter: New column name</b>	'greaterThanEqual'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	STRINGLESSTHAN( <code>stringA</code> , <code>stringB</code> )
<b>Parameter: New column name</b>	'lessThan'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	STRINGLESSTHANEQUAL( <code>stringA</code> , <code>stringB</code> )
<b>Parameter: New column name</b>	'lessThanEqual'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	EXACT( <code>stringA</code> , <code>stringB</code> )

Parameter: New column name	'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 STRINGLESSTHAN 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-left: 20px;">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.