

# ARRAYINTERSECT Function

Generates an array containing all elements that appear in multiple input arrays, referenced as column names or array literals.

**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

### Array literal reference example:

```
arrayintersect(["A", "B", "C"], ["A", "D", "E"])
```

**Output:** Returns a single array with a single element:

```
["A"]
```

### Column reference example:

```
arrayintersect([array1, array2])
```

**Output:** Returns a single array listing all of the elements that appear in both `array1` and `array2` in order.

## Syntax and Arguments

```
arrayintersect(array_ref1, array_ref2)
```

Argument	Required?	Data Type	Description
array_ref1	Y	string or array	Name of first column or first array literal to apply to the function
array_ref2	Y	string or array	Name of second column or second array literal to apply to the function

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

### array\_ref1, array\_ref2

Array literal or name of the array column whose intersection you want to derive. You can intersect two or more array columns together.

### Usage Notes:

Required?	Data Type	Example Value
Yes	Array literal or column reference	myArray1, myArray2

## Examples

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

## Example - Simple intersection example

This simple example illustrates how the following functions operate on nested data.

- ARRAYCONCAT - Concatenate multiple arrays together. See *ARRAYCONCAT Function*.
- ARRAYINTERSECT - Find the intersection of elements between multiple arrays. See *ARRAYINTERSECT Function*.
- ARRAYCROSS - Compute the cross product of multiple arrays. See *ARRAYCROSS Function*.
- ARRAYUNIQUE - Generate unique values across multiple arrays. See *ARRAYUNIQUE Function*.

### Source:

Code formatting has been applied to improve legibility.

Item	ArrayA	ArrayB
Item1	[ "A" , "B" , "C" ]	[ "1" , "2" , "3" ]
Item2	[ "A" , "B" ]	[ "A" , "B" , "C" ]
Item3	[ "D" , "E" , "F" ]	[ "4" , "5" , "6" ]

### Transformation:

You can apply the following transforms in the following order. Note that the column names must be different from the transform name, which is a reserved word.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	ARRAYCONCAT([Letters, Numerals])
<b>Parameter: New column name</b>	'concat2'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	ARRAYINTERSECT([Letters, Numerals])
<b>Parameter: New column name</b>	'intersection2'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	ARRAYCROSS([Letters, Numerals])
<b>Parameter: New column name</b>	'cross2'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	ARRAYUNIQUE([Letters, Numerals])

<b>Parameter: New column name</b>	'unique2'
-----------------------------------	-----------

**Results:**

For display purposes, the results table has been broken down into three separate sets of columns.

Column set 1:

Item	ArrayA	ArrayB	concat2	intersection2
Item1	["A", "B", "C"]	["1", "2", "3"]	["A", "B", "C", "1", "2", "3"]	[ ]
Item2	["A", "B"]	["A", "B", "C"]	["A", "B", "A", "B", "C"]	["A", "B"]
Item3	["D", "E", "F"]	["4", "5", "6"]	["D", "E", "F", "4", "5", "6"]	[ ]

Column set 2:

Item	cross2
Item1	[["A", "1"], ["A", "2"], ["A", "3"], ["B", "1"], ["B", "2"], ["B", "3"], ["C", "1"], ["C", "2"], ["C", "3"]]
Item2	[["A", "A"], ["A", "B"], ["A", "C"], ["B", "A"], ["B", "B"], ["B", "C"]]
Item3	[["D", "4"], ["D", "5"], ["D", "6"], ["E", "4"], ["E", "5"], ["E", "6"], ["F", "4"], ["F", "5"], ["F", "6"]]

Column set 3:

Item	unique2
Item1	["A", "B", "C", "1", "2", "3"]
Item2	["A", "B", "C"]
Item3	["D", "E", "F", "4", "5", "6"]