ARRAYELEMENTAT Function

Contents:

Computes the 0-based index value for an array element in the specified column, array literal, or function that returns an array.

- This function calculates based on the outer layer of an array. If your array is nested, the count of inner elements is not factored.
- If a row contains a missing array, the returned value is `0`. If it contains a value that is not recognized as an array, the returned value is null.

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:

```
arrayelementat([A,B,C,D],2)
```

Output: Returns the third value in the array, which is `C`.

Column reference example:

```
arrayelementat(myArrays,9)
```

Output: Returns the tenth element of the arrays listed in the `myArrays` column.

Array function example:

```
arrayelementat(concat([colA,colB]),3)
```

Output: Returns the fourth element of the concatenated array.

Syntax and Arguments

```
arrayelementat(array_ref,int_index_ref)
```

Argument	Required?	Data Type	Description
array_ref	Y	string	Name of Array column, Array literal, or function returning an Array to apply to the function

int_index_ref	Y	integer (non-negative)	Index value for the array element to return. Value can be Integer literal, column containing Integer values, or function returning an Integer.

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

array_ref

Name of the array column, array literal, or function returning an array whose elements you want to return.

- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference or function) or array literal	`myArray1`

int_index_ref

Non-negative integer value representing the index value of the array element to return. Value can be Integer literal, column containing Integer values, or function returning an Integer.

- Value must a non-negative integer. If the value is `0`, then the first element of the array is returned.
- If this value is greater than the length of the string, then a null value is returned.

Usage Notes:

Required?	Data Type	Example Value
Yes	Integer (non-negative)	5

Examples

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

Example - Student progress across tests

This example covers the following functions:

- `ARRAYLEN` - Returns 1-based number of elements in an array. See *ARRAYLEN Function*.
- `ARRAYELEMENTAT` - Returns array element based on 0-based index parameter. See *ARRAYELEMENTAT Function*.
- `ARRAYSORT` - Returns array sorted in ascending or descending order. See *ARRAYSORT Function*.

Source:

Here are some student test scores. Individual scores are stored in the `Scores` column. You want to:

1. Flag the students who have not taken four tests.
2. Compute the range in scores for each student.

LastName	FirstName	Scores

Allen	Amanda	[79, 83,87,81]
Bell	Bobby	[85, 92, 94, 98]
Charles	Cameron	[88,81,85]
Dudley	Danny	[82,88,81,77]
Ellis	Evan	[91,93,87,93]

Transformation:

First, you want to flag the students who did not take all four tests:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`IF(ARRAYLEN(Scores) < 4,"incomplete","")`
Parameter: New column name	`'Error'`

This test flags Cameron Charles only.

The following transform sorts the array values in highest to lowest score:

Transformation Name	`Edit column with formula`
Parameter: Columns	`Scores`
Parameter: Formula	`ARRAYSORT(Scores, 'descending')`

The following transforms extracts the first (highest) and last (lowest) value in each student's test scores, provided that they took four tests:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ARRAYELEMENTAT(Scores,0)`
Parameter: New column name	`'highestScore'`

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`ARRAYELEMENTAT(Scores,3)`
Parameter: New column name	`'lowestScore'`

Tip: You could also generate the `Error` column when the `Scores4` column contains a null value. If no value exists in the array for the `ARRAYELEMENTAT` function, a null value is returned, which would indicate in this case an insufficient number of elements (test scores).

You can now track change in test scores:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`SUBTRACT(highestScore,lowestScore)`
Parameter: New column name	`'Score_range'`

Results:

LastName	FirstName	Scores	Error	lowestScore	highestScore	Score_range
Allen	Amanda	[87,83,81,79]		79	87	8
Bell	Bobby	[98,94,92,85]		85	98	13
Charles	Cameron	[88,85,81]	incomplete		88	
Dudley	Danny	[88,82,81,77]		77	88	11
Ellis	Evan	[93,93,91,87]		87	93	6