

ARRAYELEMENTAT Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *array_ref*
 - *int_index_ref*
- *Examples*
 - *Example - Student progress across tests*

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.

Basic Usage

Array literal reference example:

```
derive type:single value:ARRAYELEMENTAT([A,B,C,D],2)
```

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

Column reference example:

```
derive type:single value:ARRAYELEMENTAT(myArrays,9) as:'10th_myArrays'
```

Output: Generates the new `10th_myArrays` column containing the tenth element of the arrays listed in the `myArrays` column.

Array function example:

```
derive type:single value:ARRAYELEMENTAT(CONCAT([colA,colB]),3) as:'strArrayElementAtIndex'
```

Output: Generates the new `strArrayElementAtIndex` column containing the fourth element of the concatenated array.

Syntax and Arguments

```
derive type:single value: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

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 must be 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.
- References to columns of integer data type are not supported.

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*.

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 change in scores between first and fourth tests.

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]

Transform:

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

```
derive type:single value:IF(ARRAYLEN(Scores) < 4,"incomplete","") as:'Error'
```

This test flags Cameron Charles only.

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

```
derive type:single value:ARRAYELEMENTAT(Scores,0) as:'Scores1'
```

```
derive type:single value:ARRAYELEMENTAT(Scores,3) as:'Scores4'
```

 **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:

```
derive type:single value:SUBTRACT(Scores4,Scores1) as:'Scores_change'
```

Results:

LastName	FirstName	Scores	Error	Scores1	Scores4	Scores_change
Allen	Amanda	[79, 83,87,81]		79	81	2
Bell	Bobby	[85, 92, 94, 98]		85	98	13
Charles	Cameron	[88,81,85]	incomplete	88		
Dudley	Danny	[82,88,81,77]		82	77	-5
Ellis	Evan	[91,93,87,93]		91	93	2