

LIST Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *function_col_ref*
 - *limit_int*
- *Examples*
 - *Example - Colors sold this month*

Extracts the set of values from a column into an array stored in a new column. This function is typically part of an aggregation.

Tip: To generate unique values for the list, apply the `ARRAYUNIQUE` function in the next step after this one. See *ARRAYUNIQUE Function*.

Input column can be of any type.

- By default, the list is limited to 1000 values. To change the maximum number of values, specify a value for the `limit` parameter.
- This function is intended to be used as part of an aggregation to return the distinct set of values by group. See *Pivot Transform*.

For a version of this function computed over a rolling window of rows, see *ROLLINGLIST Function*.

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

```
list(Name, 500)
```

Output: Returns an array of all values (up to a count of 500) from the `Name` column for each `Month` value.

Syntax and Arguments

```
list(function_col_ref, [limit_int]) [ group:group_col_ref] [limit:limit_count]
```

| Argument | Required? | Data Type | Description |
|-------------------------------|-----------|--------------------|--|
| <code>function_col_ref</code> | Y | string | Name of column to which to apply the function |
| <code>limit_int</code> | N | integer (positive) | Maximum number of values to extract into the list array. From 1 to 1000. |

For more information on the `group` and `limit` parameters, see *Pivot Transform*.

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

function_col_ref

Name of the column from which to extract the list of values based on the grouping.

- Literal values are not supported as inputs.
- Multiple columns and wildcards are not supported.

Usage Notes:

| Required? | Data Type | Example Value |
|-----------|---------------------------|---------------|
| Yes | String (column reference) | myValues |

limit_int

Non-negative integer that defines the maximum number of values to extract into the list array.

NOTE: If specified, this value must be between 1 and 1000, inclusive.

NOTE: Do not use the limiting argument in a `LIST` function call on a flat aggregate, in which all values in a column have been inserted into a single cell. In this case, you might be able to use the `limit` argument if you also specify a `group` parameter. Misuse of the `LIST` function can cause the application to crash.

Usage Notes:

| Required? | Data Type | Example Value |
|-----------|-----------|---------------|
| No | Integer | 50 |

Examples

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

Example - Colors sold this month

This example illustrates the following functions:

- `LIST` - Extracts up to 1000 values from one column into an array in a new column. See *LIST Function*.
- `UNIQUE` - Extracts up to 1000 unique values from one column into an array in a new column. See *UNIQUE Function*.

You have the following set of orders for two months, and you are interested in identifying the set of colors that have been sold for each product for each month and the total quantity of product sold for each month.

Source:

| OrderId | Date | Item | Qty | Color |
|---------|---------|-------|-----|-------|
| 1001 | 1/15/15 | Pants | 1 | red |

| | | | | |
|------|---------|-------|---|--------|
| 1002 | 1/15/15 | Shirt | 2 | green |
| 1003 | 1/15/15 | Hat | 3 | blue |
| 1004 | 1/16/15 | Shirt | 4 | yellow |
| 1005 | 1/16/15 | Hat | 5 | red |
| 1006 | 1/20/15 | Pants | 6 | green |
| 1007 | 1/15/15 | Hat | 7 | blue |
| 1008 | 4/15/15 | Shirt | 8 | yellow |
| 1009 | 4/15/15 | Shoes | 9 | brown |
| 1010 | 4/16/15 | Pants | 1 | red |
| 1011 | 4/16/15 | Hat | 2 | green |
| 1012 | 4/16/15 | Shirt | 3 | blue |
| 1013 | 4/20/15 | Shoes | 4 | black |
| 1014 | 4/20/15 | Hat | 5 | blue |
| 1015 | 4/20/15 | Pants | 6 | black |

Transformation:

To track by month, you need a column containing the month value extracted from the date:

| | |
|----------------------------|------------------------------|
| Transformation Name | Edit column with formula |
| Parameter: Columns | Date |
| Parameter: Formula | DATEFORMAT(Date, 'MMM yyyy') |

You can use the following transform to check the list of unique values among the colors:

| | |
|---|---------------------|
| Transformation Name | Pivot columns |
| Parameter: Row labels | Date |
| Parameter: Values | unique(Color, 1000) |
| Parameter: Max number of columns to create | 10 |

| Date | unique_Color |
|----------|---|
| Jan 2015 | ["green","blue","red","yellow"] |
| Apr 2015 | ["brown","blue","red","yellow","black","green"] |

Delete the above transform.

You can aggregate the data in your dataset, grouped by the reformatted Date values, and apply the LIST function to the Color column. In the same aggregation, you can include a summation function for the Qty column:

| | |
|------------------------------|----------------------------|
| Transformation Name | Pivot columns |
| Parameter: Row labels | Date |
| Parameter: Values | list(Color, 1000),sum(Qty) |

| | |
|---|----|
| Parameter: Max number of columns to create | 10 |
|---|----|

Results:

| Date | list_Color | sum_Qty |
|-------------|--|----------------|
| Jan 2015 | ["green","blue","blue","red","green","red","yellow"] | 28 |
| Apr 2015 | ["brown","blue","red","yellow","black","blue","black","green"] | 38 |