Table Calculation - Hands On

Tableau Aggregate Function

Tableau is a powerful data visualization tool that allows users to perform complex calculations and functions. Here are some of the most commonly used numerical functions in Tableau:

Function	Description	Example
SUM()	Returns the sum of all the values in a particular field.	**SUM([Sales]) would return the total sales.**
AVG()	Returns the average of all the values in a field.	**AVG([Sales]) would return the average sales.**
MIN()	Returns the minimum value in a field.	**MIN([Sales]) would return the smallest sales value.**
MAX()	Returns the maximum value in a field.	**MAX([Sales]) would return the largest sales value.**
COUNT()	Returns the number of rows in a field.	**COUNT([Customer Name]) would give you the number of customers.**
COUNTD()	Returns the number of distinct values in a field.	**COUNTD([Customer Name]) would give you the number of unique customers.**
MEDIAN()	Returns the median value of a field.	**MEDIAN([Sales]) would give the middle sales value when all sales are listed in numerical order.**
ROUND()	Rounds a number to a specified number of decimal places.	**ROUND([Sales], 2) would round the sales values to two decimal places.**
ABS()	Returns the absolute value of a number.	**ABS([-5]) would return 5.**
SQRT()	Finds the square root of a number.	**SQRT([Sales]) would return the square root of the sales values.**

Remember to wrap your field names in square brackets [] when using these functions. These functions can be used in creating calculated fields, setting conditions, or in creating more complex queries.

Date Functions

MTD,YTD, QTD

In Tableau, the terms MTD, QTD, and YTD stand for Month-To-Date, Quarter-To-Date, and Year-To-Date, respectively. These are all types of cumulative calculations, where the aim is to show the aggregated measure from the start of the period (month, quarter, or year) up to the current or latest available date.

Here are some of the most frequently used date functions in Tableau:

Term	Description	Tableau Calculation
MTD (Month-To-Date)	MTD is a cumulative calculation that shows the aggregated measure from the start of the month up to the current or latest available date.	**IF DATETRUNC('month', [Order Date]) = DATETRUNC('month', TODAY()) THEN [Sales] ELSE NULL END**
QTD (Quarter-To-Date)	QTD is a cumulative calculation that shows the aggregated measure from the start of the quarter up to the current or latest available date.	**IF DATETRUNC('quarter', [Order Date]) = DATETRUNC('quarter', TODAY()) THEN [Sales] ELSE NULL END**
YTD (Year-To-Date)	YTD is a cumulative calculation that shows the aggregated measure from the start of the year up to the current or latest available date.	**IF DATETRUNC('year', [Order Date]) = DATETRUNC('year', TODAY()) THEN [Sales] ELSE NULL END**

In these formulas, replace [Sales] with the field you are interested in and [Order Date] with the appropriate date field in your dataset. The calculations provide the sales (or other metric) for the specified period up to the current date. If you want the full month, quarter, or year (rather than up to the current date), you will need to adjust your calculation accordingly.

YOY, MOM, QOQ

Year-over-year (YOY), Month-over-month (MOM), and Quarter-over-quarter (QOQ) are often used in financial and business analysis to compare performance across different time periods.

In Tableau, you can create these calculations using a combination of date functions and basic arithmetic operations. Here's how you can set up these calculations:

These calculations give you the relative growth or change in your chosen measure (like sales) from one period to the next. The "-1" at the end of each formula is used to convert the ratio to a percentage change.

Calculation	Description	Tableau Calculation
YOY (Year-over-Year)	YOY growth compares a measure in the current year to the same measure in the previous year.	**(SUM(IF YEAR([Order Date]) = YEAR(TODAY()) THEN [Sales] END) / SUM(IF YEAR([Order Date]) = YEAR(DATEADD('year', -1, TODAY())) THEN [Sales] END)) – 1**

MOM (Month-over-Month)	MOM calculation compares a measure in the current month to the same measure in the previous month.	**(SUM(IF DATETRUNC('month', [Order Date]) = DATETRUNC('month', TODAY()) THEN [Sales] END) / SUM(IF DATETRUNC('month', [Order Date]) = DATETRUNC('month', DATEADD('month', -1, TODAY())) THEN [Sales] END)) - 1**
QOQ (Quarter-over-Quarter)	QOQ growth compares a measure in the current quarter to the same measure in the previous quarter.	**(SUM(IF DATETRUNC('quarter', [Order Date]) = DATETRUNC('quarter', TODAY()) THEN [Sales] END) / SUM(IF DATETRUNC('quarter', [Order Date]) = DATETRUNC('quarter', DATEADD('quarter', -1, TODAY())) THEN [Sales] END)) - 1**

Datadiff

In Tableau, the DATEDIFF function is used to calculate the difference between two dates. The basic syntax of the function is as follows:

DATEDIFF(date_part, date1, date2, [start_of_week])

The date_part parameter specifies the part of the date to focus on, such as 'year', 'quarter', 'month', 'day', etc.

The date1 and date2 parameters are the dates you're comparing,

the start_of_week parameter is optional and lets you specify which day is to be considered the first day of the week.

Here are some examples of the DATEDIFF function:

Calculation	Description	Example	Use Case
DATEDIFF('day', date1, date2)	Calculates the number of days between two dates.	**DATEDIFF('day', [Order Date], [Ship Date])**	This will return the number of days between the Order Date and Ship Date.
DATEDIFF('month', date1, date2)	Calculates the number of months between two dates.	**DATEDIFF('month', [Order Date], [Ship Date])**	This will return the number of months between the Order Date and Ship Date.
DATEDIFF('week', date1, date2, 'start_of_week')	Calculates the number of weeks between two dates, with an optional parameter to specify the first day of the week (default is Sunday).	**DATEDIFF('week', [Start Date], [End Date], 'monday')**	This will return the number of weeks between Start Date and End Date, considering Monday as the first day of the week.

Table Calculations:

Tableau's window functions are a subset of table calculations that perform a calculation across a set of table rows that are related to the current row. These are commonly used for running calculations and comparisons between different rows in a partition of the data.

Here are some of the most frequently used window functions in Tableau:

Function	Description	Example
WINDOW_SUM(expression, [start, end])	Computes the sum of the expression across all rows in the window.	**WINDOW_SUM(SUM([Sales]))**
WINDOW_AVG(expression, [start, end])	Calculates the average of the expression across all rows in the window.	**WINDOW_AVG(AVG([Profit]))**
WINDOW_MAX(expression, [start, end])	Returns the maximum of the expression across all rows in the window.	**WINDOW_MAX(MAX([Sales]))**
WINDOW_MIN(expression, [start, end])	Returns the minimum of the expression across all rows in the window.	**WINDOW_MIN(MIN([Sales]))**
WINDOW_COUNT(expression, [start, end])	Counts the number of rows in the window.	**WINDOW_COUNT(COUNT([Order ID]))**
WINDOW_MEDIAN(expression, [start, end])	Returns the median of the expression across all rows in the window.	**WINDOW_MEDIAN(MEDIAN([Profit]))**
FIRST()	Returns the number of rows from the current row to the first row in the partition.	**FIRST()**
LAST()	Returns the number of rows from the current row to the last row in the partition.	**LAST()**
INDEX()	Returns the index of the current row in the partition.	**INDEX()**
RANK() and RANK_DENSE()	Assign a unique rank to each row in a partition, with the option to rank either ascending or descending.	**RANK(), RANK_DENSE()**
RUNNING_SUM(expression)	Returns the running sum of the expression across all rows in the window, starting at the first row and ending at the current row.	**RUNNING_SUM(SUM([Sales]))**

RUNNING_AVG(expression)	Returns the running average of the expression across all rows in the window, starting at the first row and ending at the current row.	**RUNNING_AVG(AVG([Profit]))**
LOOKUP(expression, [offset])	Fetches the value of the expression from a certain row, defined by the offset relative to the current row.	**LOOKUP(SUM([Sales]), -1)**

Window functions in Tableau can be incredibly powerful, but they require careful handling. You often need to specify the partitioning and ordering of your data using the 'Compute Using' options in the Table Calculation dialog.

String Calculation:

In Tableau, the Sample Superstore dataset includes a variety of string fields, such as 'Category', 'Sub-Category', 'Region', 'State', and 'Product Name'. Here are some commonly used string functions applied to this dataset:

Function	**Description**	**Example**
LEFT(string, num_chars)	Returns the first (left-most) 'num_chars' characters from 'string'.	**LEFT([Product Name], 5)**
RIGHT(string, num_chars)	Returns the last (right-most) 'num_chars' characters from 'string'.	**RIGHT([Product Name], 5)**
MID(string, start, [length])	Returns a substring from 'string', starting at position 'start' and of 'length' characters.	**MID([Product Name], 2, 5)**
LEN(string)	Returns the length of 'string'.	**LEN([Product Name])**
UPPER(string)	Converts 'string' to upper case.	**UPPER([Region])**
LOWER(string)	Converts 'string' to lower case.	**LOWER([Region])**
TRIM(string)	Removes leading and trailing spaces from 'string'.	**TRIM([Product Name])**
SPLIT(string, delimiter, token_number)	Splits 'string' into parts based on a 'delimiter' and returns the part at 'token_number'.	**SPLIT([Product Name], ' ', 1)**
CONTAINS(string, substring)	Returns TRUE if 'substring' is found in 'string', and FALSE otherwise.	**CONTAINS([Product Name], 'Table')**

Type Conversion

In Tableau, you may often need to convert data from one type to another. Here are some commonly used conversion functions with examples:

Function	Description	Example
INT(expression)	Converts the result of the expression to an integer.	**INT([Sales])**
FLOAT(expression)	Converts the result of the expression to a float.	**FLOAT([Profit])**
STR(expression)	Converts the result of the expression to a string.	**STR([Order ID])**
DATE(expression)	Converts the result of the expression to a date. The expression must result in a string in the format 'yyyy-mm-dd' or 'yyyy-mm-dd hh:mm:ss'.	**DATE([Order Date String])**
DATETIME(expression)	Converts the result of the expression to a datetime. The expression must result in a string in the format 'yyyy-mm-dd hh:mm:ss'.	**DATETIME([Order DateTime String])**

Logical Calculations:

Tableau allow you to perform a test and return a result based on whether the test is true or false. These calculations are especially useful for segmenting data, creating groups or categories, and preparing data for further analysis.

Here are some of the most common logical functions and operations used in Tableau:

Function	Description	Example
IF THEN ELSE	Allows you to perform a test and return different results based on whether the test is true or false.	**IF [Sales] > 1000 THEN 'High Sales' ELSE 'Low Sales' END**
AND	Allows you to test whether two conditions are both true.	**IF [Sales] > 1000 AND [Profit] > 100 THEN 'High Performing' ELSE 'Low Performing' END**
OR	Allows you to test whether at least one of two conditions is true.	**IF [Sales] > 1000 OR [Profit] > 100 THEN 'High Performing' ELSE 'Low Performing' END**
NOT	Reverses the result of a test.	**IF NOT([Sales] > 1000) THEN 'Low Sales' ELSE 'High Sales' END**
CASE	Allows you to perform a series of tests and return different results based on the first test that is true.	**CASE [Region] WHEN 'West' THEN 'W' WHEN 'East' THEN 'E' WHEN 'Central' THEN 'C' WHEN 'South' THEN 'S' ELSE 'Other' END**
ZN(expression)	Used to replace null values with zero.	**ZN([Profit])**
IIF(test_expression, true_value, false_value, [null_value])	A logical function used to perform conditional tests. Returns a value if the condition is true, and another value if the	**IIF([Profit] > 0, "Profit", "Loss", "Unknown")**

	condition is false. Also includes an option to return a value if the test expression is null.	

Note: IF when you need to check multiple conditions sequentially, and use IIF when you need to check a single condition and handle NULL values specifically.