**Table of Contents**

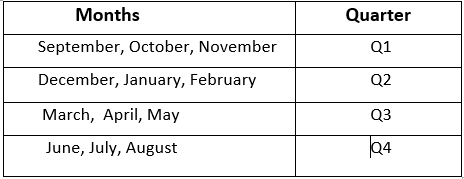
[1. Filter data based on Quarters *(Start – 8:33)* ………….2](#_Toc107479601)

[2. Adding Year to Date (YTD) and Year to Go (YTG) (8:33 – End) …….5](#_Toc107479602)

`

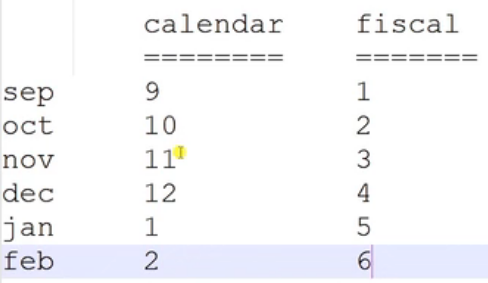
# 1. Filter data based on Quarters *(Start – 8:33)*

Any business would like to filter and compare their data by quarter. For a calendar year, Q1 will be from Jan to March… Q4 will be from Oct to December. Basically, a quarter is 1/4th which is 3 months since there are 12 months in a year. For Atliq, the fiscal year starts in **September** and the quarters are divided as follows:



**Steps for getting the Quarters:**

1. Go to the Data view and click on the **dim\_date** table.
2. Now, here you need to know the difference between the calendar year and the fiscal year
   1. A calendar year is the one where January is the starting month.
   2. The fiscal year depends from company to company and for Atliq it starts in **September**.



1. Now, you need to create a new column with the name **fy\_month\_num** which shows the fiscal month number for a given date.
   1. **What is a fy\_month\_num?** Take an example - September can also be termed as month no. 9 as per the calendar year. However, as per fiscal year, it will be no. 1 as it is the start of fiscal year.
2. As the values in the date column of the **dim\_date** table are represented in calendar year form, you need to add an extra 4 months to the date to get the fiscal month.

i.e fiscal\_month = calendar\_month + 4

**Why add an extra 4 months to get the fiscal year month?**

* For Atliq, the financial year starts in September, and generally, the calendar year starts in January.
* So, Atliq’s starting month is ahead of 4 months to the calendar year, and this changes from company to company based on their business needs.
* So, the calendar month number for September is 9 and the fiscal month number is 1. In order to make this 9-month number to 1, we need to add an extra 4 to the extracted month and the DATE() function will convert it as the month ‘January’ and give month number 1.
* Same goes for the October month, where the calendar year month is 10, and when we add 4, it then becomes 14 and the DATE() function converts into the February month and gives the month number 2.

1. So, you will extract the year using YEAR() and month using MONTH() from the date column of dim\_date and add an extra 4 to the extracted month, and finally create a new date using the DATE() function and from that, you will extract the fiscal month. The formula goes as follows:

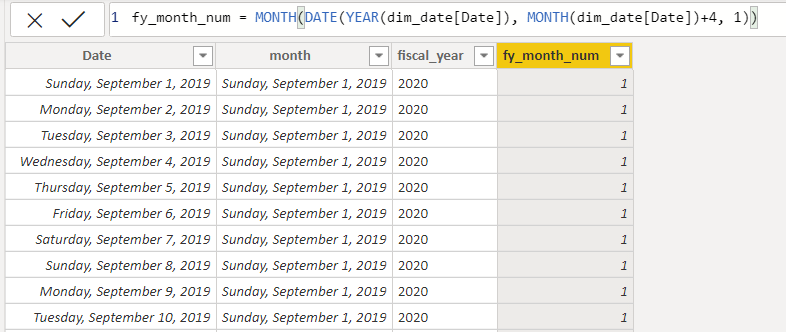
**fy\_month\_num** = MONTH( //getting the month from newly created date

DATE( //creating the new date

YEAR(dim\_date[date], //getting the year

MONTH(dim\_date[date])+4, //getting the month and adding 4 to get fiscal\_month

1)) //day of the month



1. Getting the Quarter from the new column **fy\_month\_num**:

**quarters** = “Q” //Adding Prefix “Q”

& // & is useful for concatenating

ROUNDUP( //rounding to the upper digit

dim\_date[fy\_month\_num] / 3, //divide by 3 to get quarter number

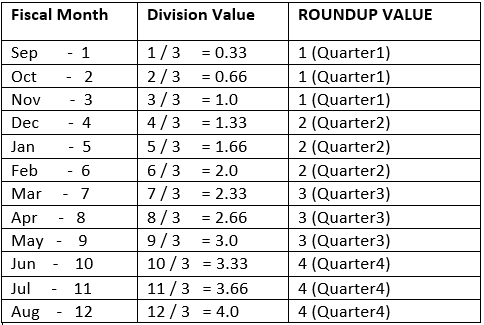
0 //Number of decimal we want

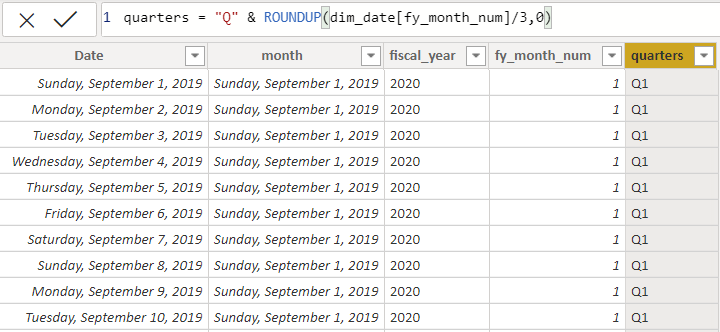
)

**Why divide fy\_month\_num by 3 and do the roundup?**

* We can better understand this by taking examples and decoding them.
* Suppose the fiscal month is 1[September], then **1 divided by 3** gives **0.33** and the roundup function rounds the result to the **upper decimal number** and gives the value as 1 which means Quarter1.
* Let’s take another fiscal month 3[November], then 3 divided by 3 gives 1 and roundup returns the value 1 which is again the Quarter1.
* For January, the fiscal month is 5, then 5 divided by 3 gives 1.66 and roundup returns value 2 which means Quarter2.

Complete Process:





1. Finally, you will see it in the reporting:



# 2. Adding Year to Date (YTD) and Year to Go (YTG) (8:33 – End)

By general definition, YTD means the value accumulated from the beginning of the year until today and YTG means the rest of the year.  
  
If you go with that definition, in the course, Sep 1st to the Current date will be YTD, and the rest of the days until 31st August should be YTG.   
  
Now, sit back for a second and ask the question:  
  
**Why would Atliq or any company need YTD and YTG?**

Sales (also called Actuals) YTD will help Atliq understand how much they have sold until yesterday from the beginning of the fiscal year. Now they compare it with last year’s sales YTD or Target YTD to see if they are aligned with the vision.  
  
While you cannot have something like Sales YTG (because sales data is something you have sold already and YTG is future) you can have Forecast YTG or Target YTG to understand how much we need to sell in the rest of the year to meet the target or meet our forecast plan.  
  
**In other words, Sep 1st to the last sales date will be YTD, and the rest of the days until 31st August should be YTG.**Now, let’s back to the dataset  
  
Atliq’s dataset for this course was created in Jan 2022. Hence, the last sales date would be in the month of December 2021. Therefore, September to December will be **Year to Date,** and the rest of the days till year-end [August] is **Year to Go**.

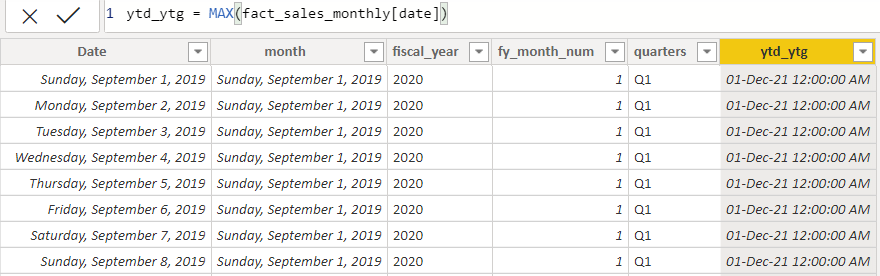
**Steps for getting YTD and YTG:**

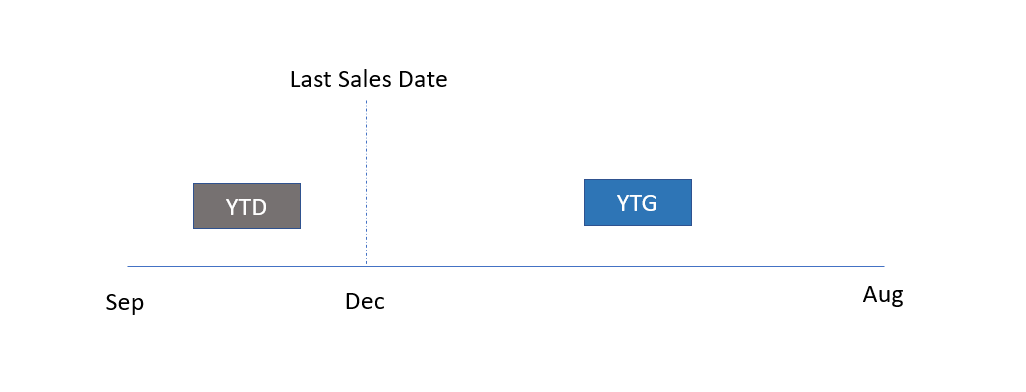
1. You can check the last sale date from the sales table using MAX()

i.e MAX(fact\_sales\_monthly[date])

Note: Even though Atliq’s dataset is not updating every day you need to think about creating a dynamic formula as if the last sales date gets updated every day. By using Max() you will ensure that always the last sales date is captured.

Ytd\_ytg measure below represents the last sales date.



1. Let’s go ahead and find if each of the values in the date column is YTD or YTG.   
     
    **Now, think like an analyst!  
     
   Use this thought process to build this column dynamically for all fiscal years.**(Remember, this is just one way to do it – you can come up with an even better idea!)  
   As you could have realized already – the ‘last sales date’ is the boundary between YTD and YTG.  
     
   Hence, the logic for generating YTD and YTG is when a date in the date column is greater than the last sales then it is YTG. otherwise, it is YTD.   
     
     
     
   However, if you compare the date column with the last sales date you will get the YTD and YTG only for the current fiscal year which is 2022. Since you need it for all fiscal years – a column that works for all fiscal years is needed i.e. a column that is not attributed to the year but just the months.  
     
   And we have the winner, it’s **fy\_month\_num**.  
     
   **Why?** -> You can see that it has the fiscal year month number which is the same for all fiscal years. i.e fy\_month\_num for Dec 21 and Dec 20 is the same as it takes the month into the account and not the year.  
     
   So, you will compare the fy\_month\_num of the date column vs fy\_month\_num of the last sales date to find if a given date is YTD or YTG.

We have the fy\_month\_num already, let’s find the fy\_month\_num of the last sales date now.

1. First, get the last sale date in fact\_sales\_monthly[date]) and store it in a variable LASTSALESDATE.

var LASTSALESDATE = MAX(LastSalesMonth[LastSalesMonth]) //getting the last sale date

1. Find the fiscal year month number of ‘last sales date’ using the same formula we used in section 1.

var FYMONTHNUM = MONTH( //getting the month from newly created date

DATE( //creating the new date

YEAR(LASTSALESDATE), //getting the year

MONTH(LASTSALESDATE)+4, //getting the month and adding 4 to get fiscal\_year

1. //day of the month

)

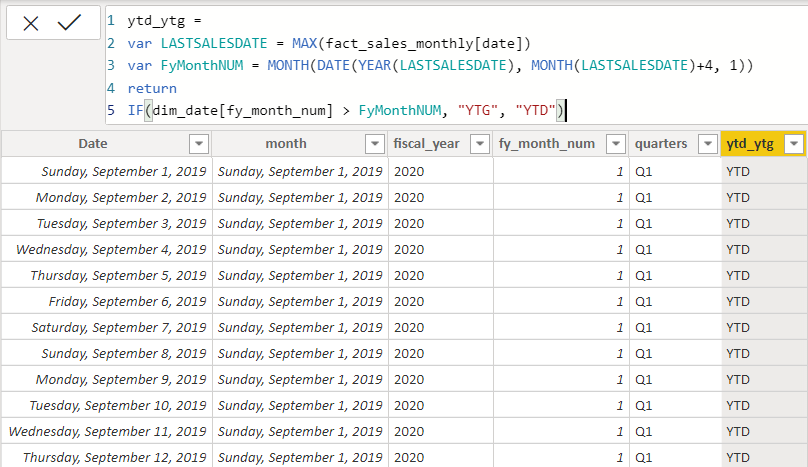
5. Now you need to use the if condition to check whether the fy\_month\_num of a date is greater than FYMONTHNUM of last sales date in order to assign it as YTG or else YTD.

IF(dim\_date[fy\_month\_num] > FyMonthNUM, //if fiscal month > last date fiscal month

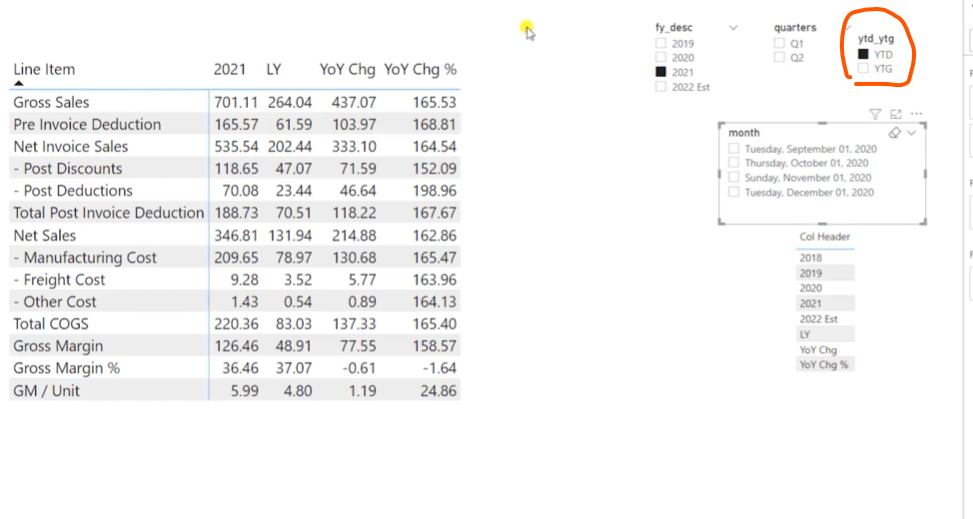
"YTG", //if the condition is True, Year to Go

"YTD") //if the condition is False, Year to Date

6. Final Formula and Output:



As you can see you have got YTD or YTG definition for each of the date values and this is another reason to have this created as a column than a measure. Since it is a stored column value now, it can be used as a page slicer.



Happy Learning 😊