POWER-BI SCENARIO BASED QUESTIONS

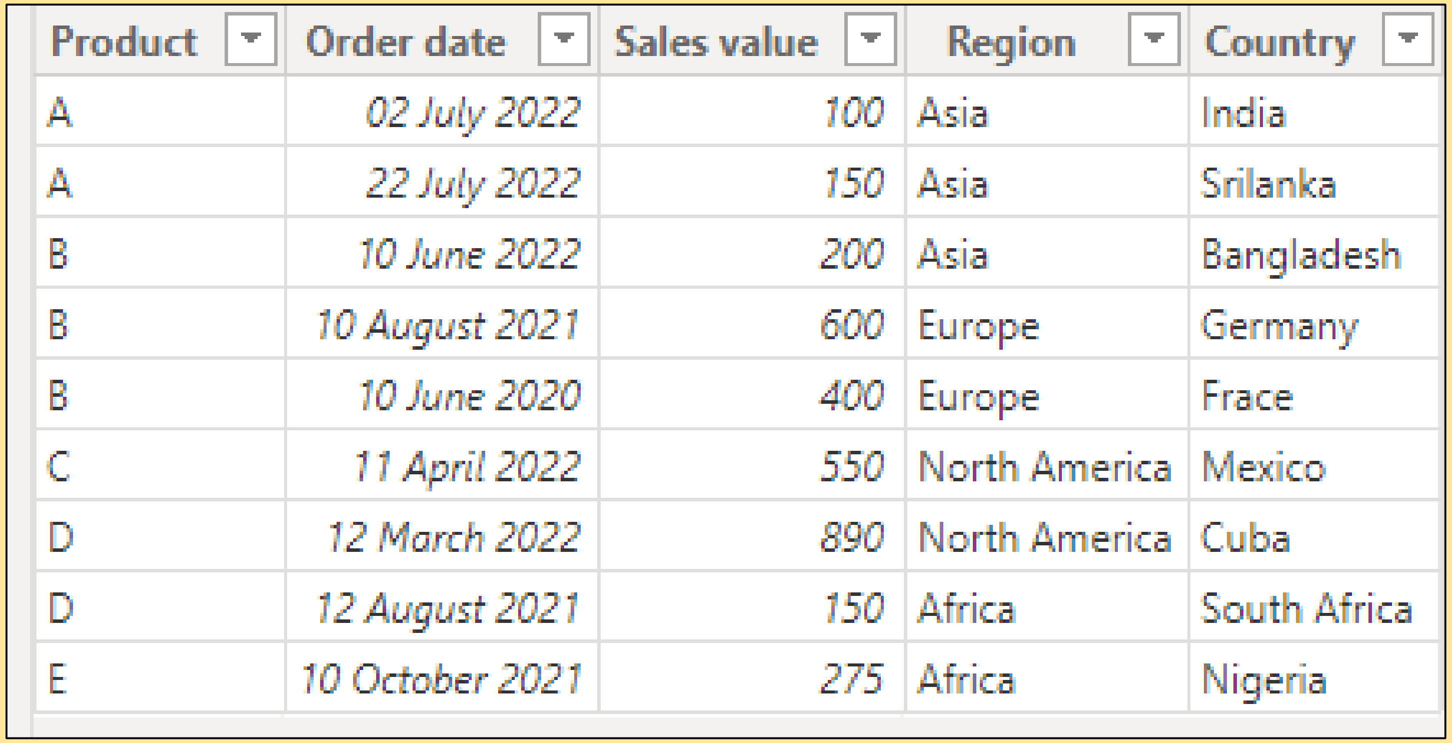
QUESTION 15 & 16

- DATESYTD() vs TOTALYTD()
- Calculate Running Total

- MAYURI .D.

QUESTION 15

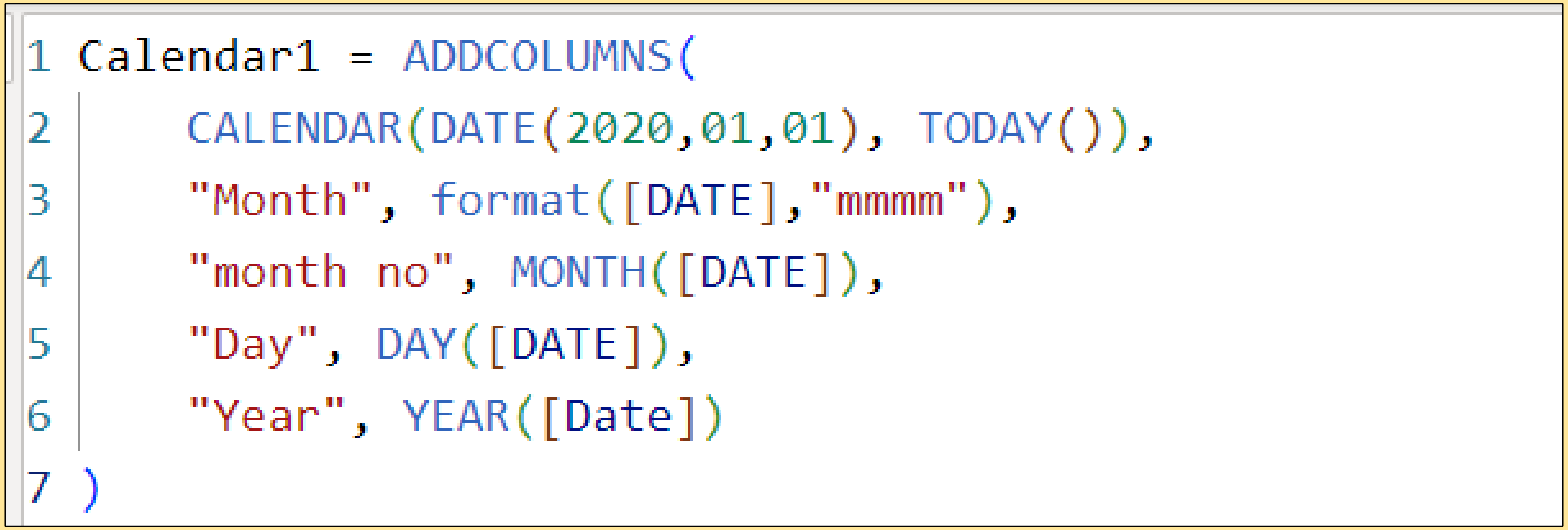
SAMPLE TABLES



Product	Order date	Sales value	Region	Country
A	02 July 2022	100	Asia	India
A	22 July 2022	150	Asia	Srilanka
B	10 June 2022	200	Asia	Bangladesh
B	10 August 2021	600	Europe	Germany
B	10 June 2020	400	Europe	Frace
C	11 April 2022	550	North America	Mexico
D	12 March 2022	890	North America	Cuba
D	12 August 2021	150	Africa	South Africa
E	10 October 2021	275	Africa	Nigeria

Sales3 table

Calendar table



```
Calendar1 = ADDCOLUMNS(
        CALENDAR(DATE(2020,01,01), TODAY()),
        "Month", format([DATE],"mmmm"),
        "month no", MONTH([DATE]),
        "Day", DAY([DATE]),
        "Year", YEAR([Date])
)
```

QUESTION 15(1)

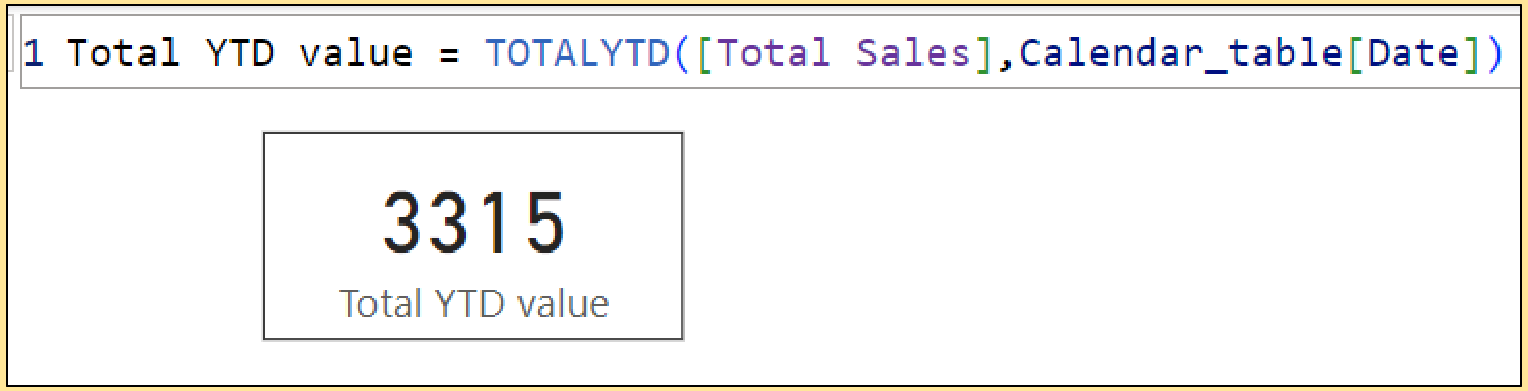
Differentiate between TOTALYTD() and DATESYTD()

- TOTALYTD() evaluates the value of given expression till the current date whereas DATESYTD() returns table of date column till current date

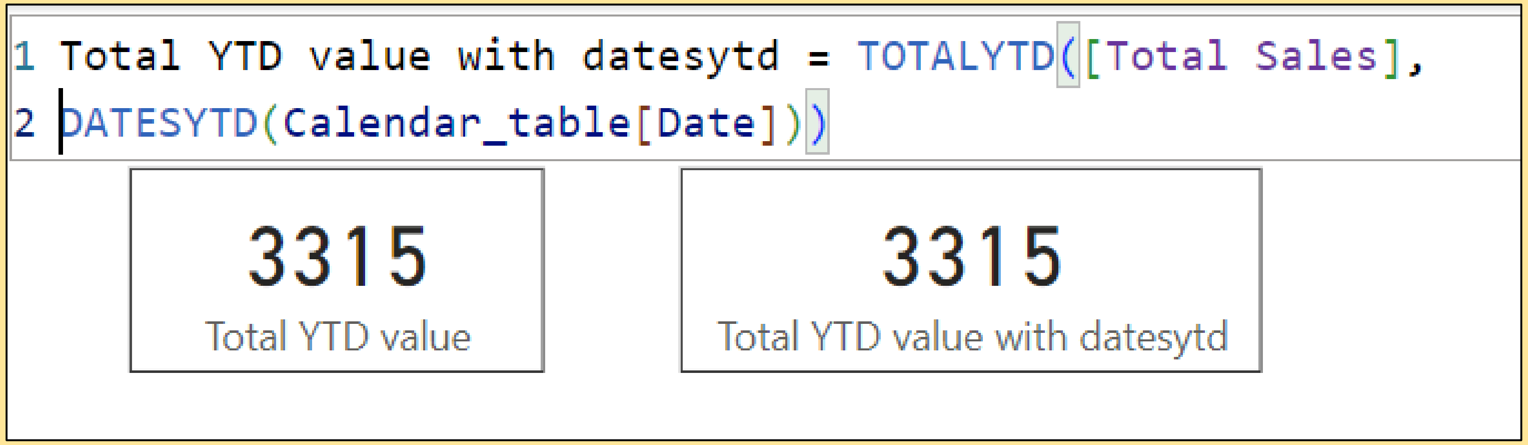
- TOTALYTD() takes compulsory parameters likes expression and date whereas DATESYTD() takes only one compulsory parameter like date

QUESTION 15(2)

Can DATESYTD() be used within TOTALYTD() ?



```
Total YTD value = TOTALYTD([Total Sales],Calendar_table[Date])
```



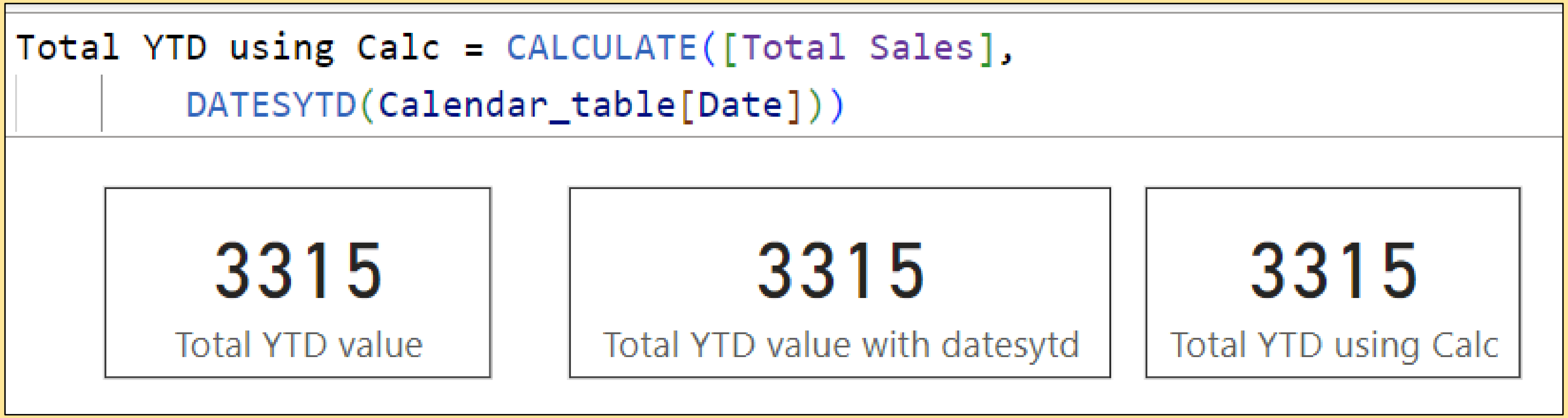
```
Total YTD value with datesytd = TOTALYTD([Total Sales],
DATESYTD(Calendar_table[Date]))
```

1st measure is simply to calculate YTD by simply giving date column.
2nd measure is to calculate YTD by using DATESYTD() within TOTALYTD().

From both measure, it is clearly visible that they return same results,
So yes DATESYTD() can be used within TOTALYTD().

QUESTION 15(3)

Calculate YTD using DATESYTD() without using TOTALYTD()



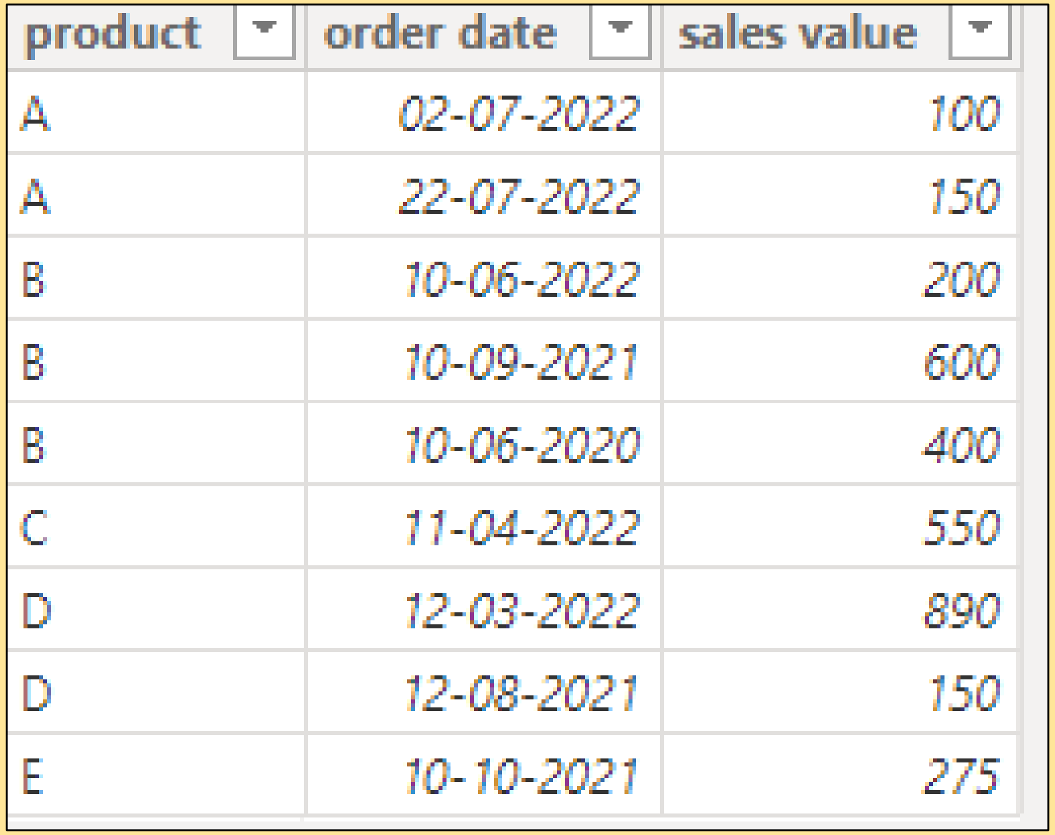
```
Total YTD using Calc = CALCULATE([Total Sales],
        DATESYTD(Calendar_table[Date]))
```

Without using TOTALYTD() same result is obtained like other two measures. So it is possible to calculate TOTALYTD by using just DATESYTD() without using TOTALYTD().

QUESTION 16

Calculate Running Total

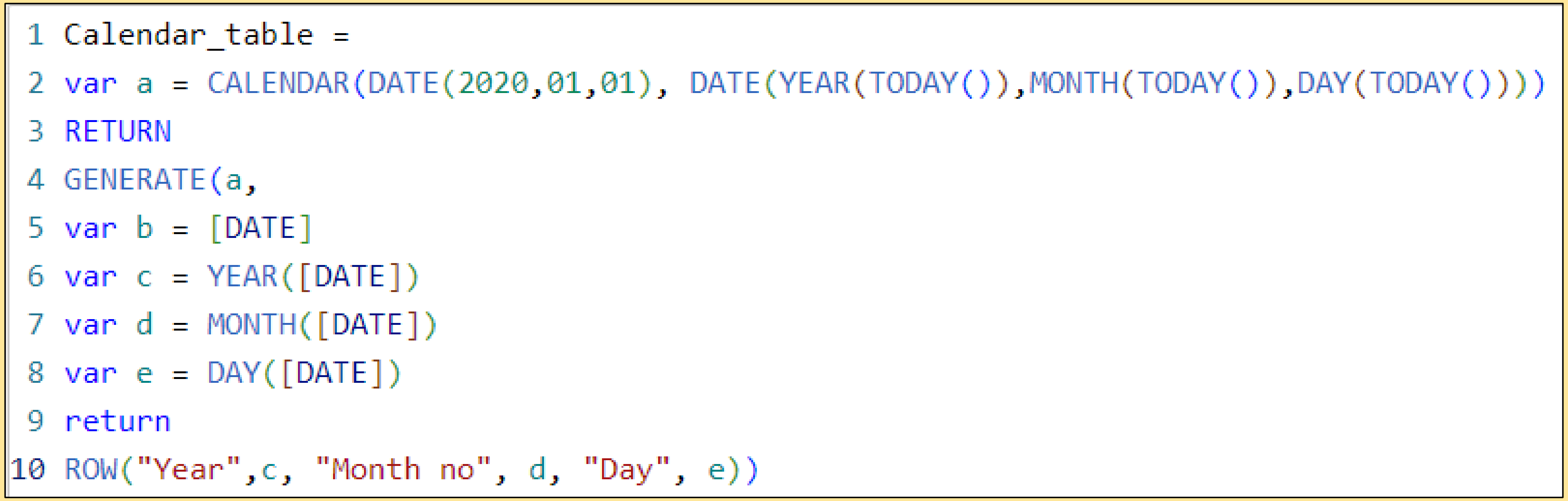
SAMPLE TABLE



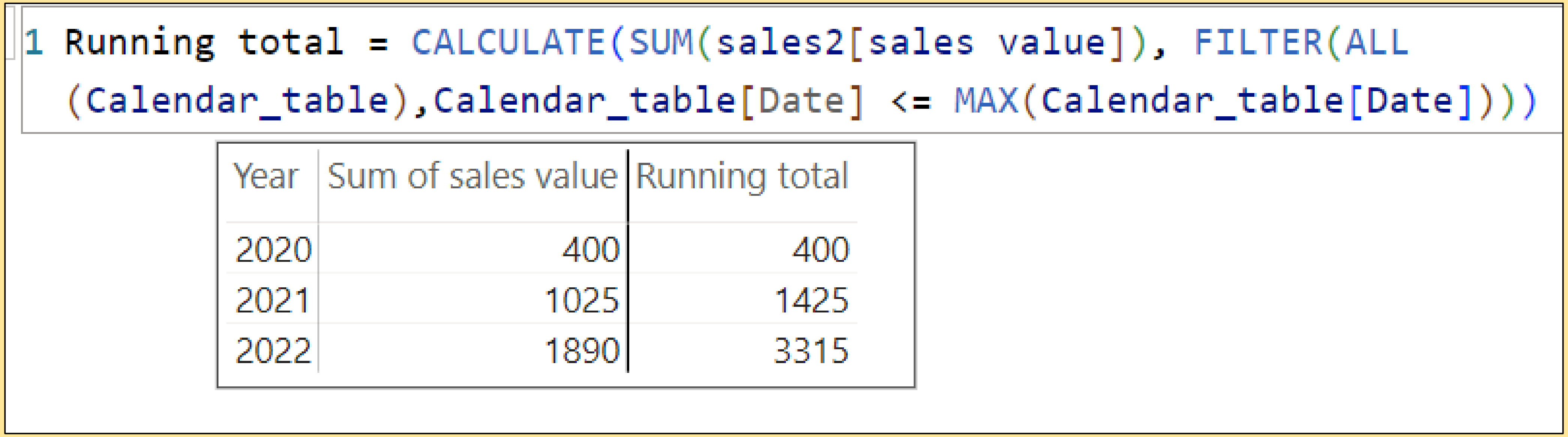
product	order date	sales value
A	02-07-2022	100
A	22-07-2022	150
B	10-06-2022	200
B	10-09-2021	600
B	10-06-2020	400
C	11-04-2022	550
D	12-03-2022	890
D	12-08-2021	150
E	10-10-2021	275

Sales2 table

Calendar table



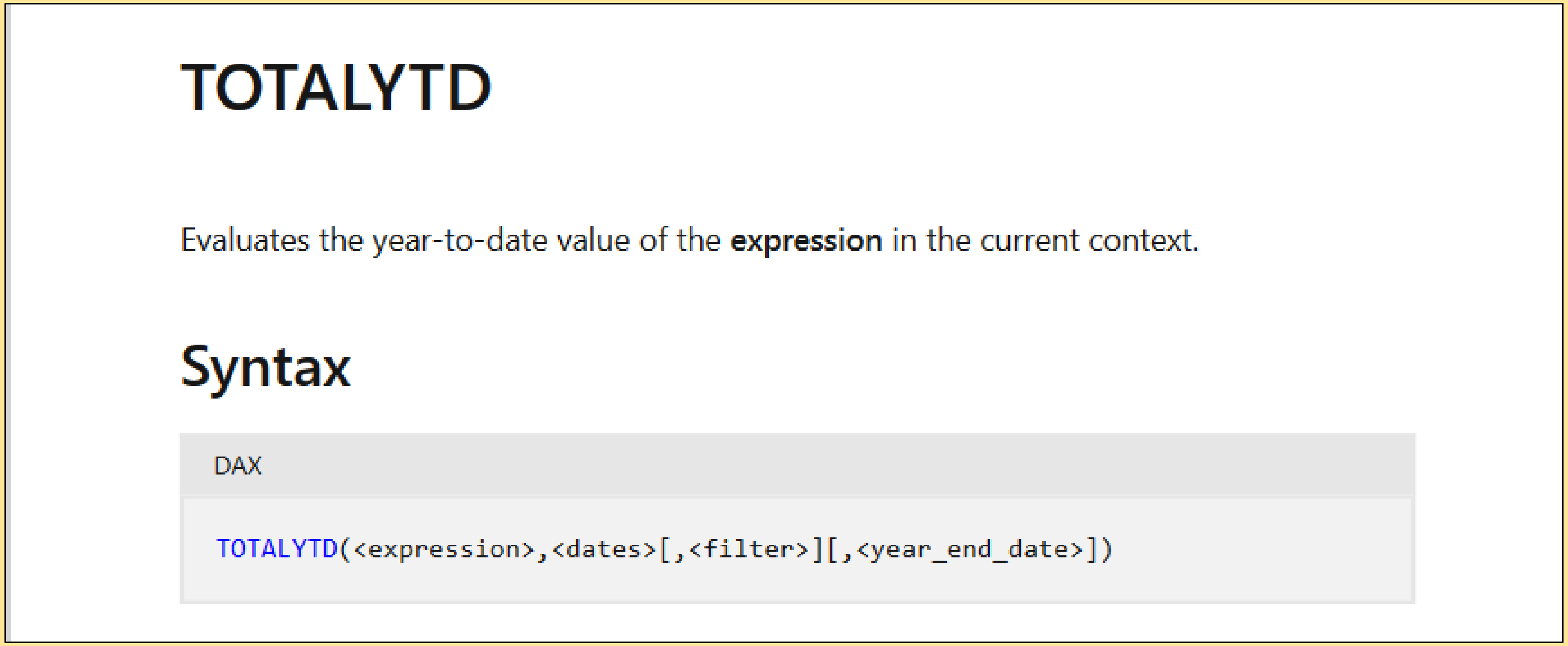
```
Calendar_table =
var a = CALENDAR(DATE(2020,01,01), DATE(YEAR(TODAY()),MONTH(TODAY()),DAY(TODAY())))
RETURN
GENERATE(a,
var b = [DATE]
var c = YEAR([DATE])
var d = MONTH([DATE])
var e = DAY([DATE])
return
ROW("Year",c, "Month no", d, "Day", e))
```



```
Running total = CALCULATE(SUM(sales2[sales value]), FILTER(ALL
  (Calendar_table),Calendar_table[Date] <= MAX(Calendar_table[Date])))
```

Year	Sum of sales value	Running total
2020	400	400
2021	1025	1425
2022	1890	3315

Add up the sales for all the days from the start up to the current date, creating a running total that grows as you go through each day.

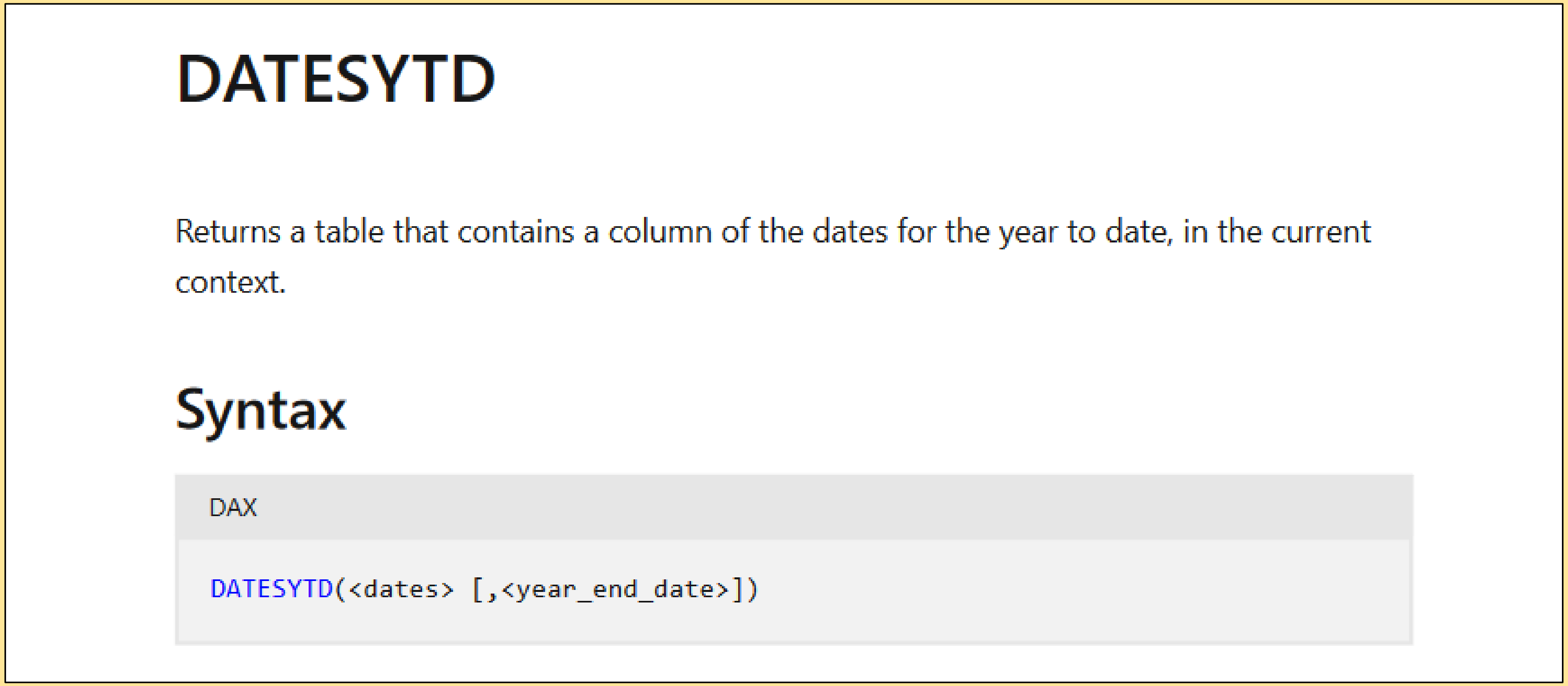

TOTALYTD

Evaluates the year-to-date value of the **expression** in the current context.

Syntax

DAX

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```



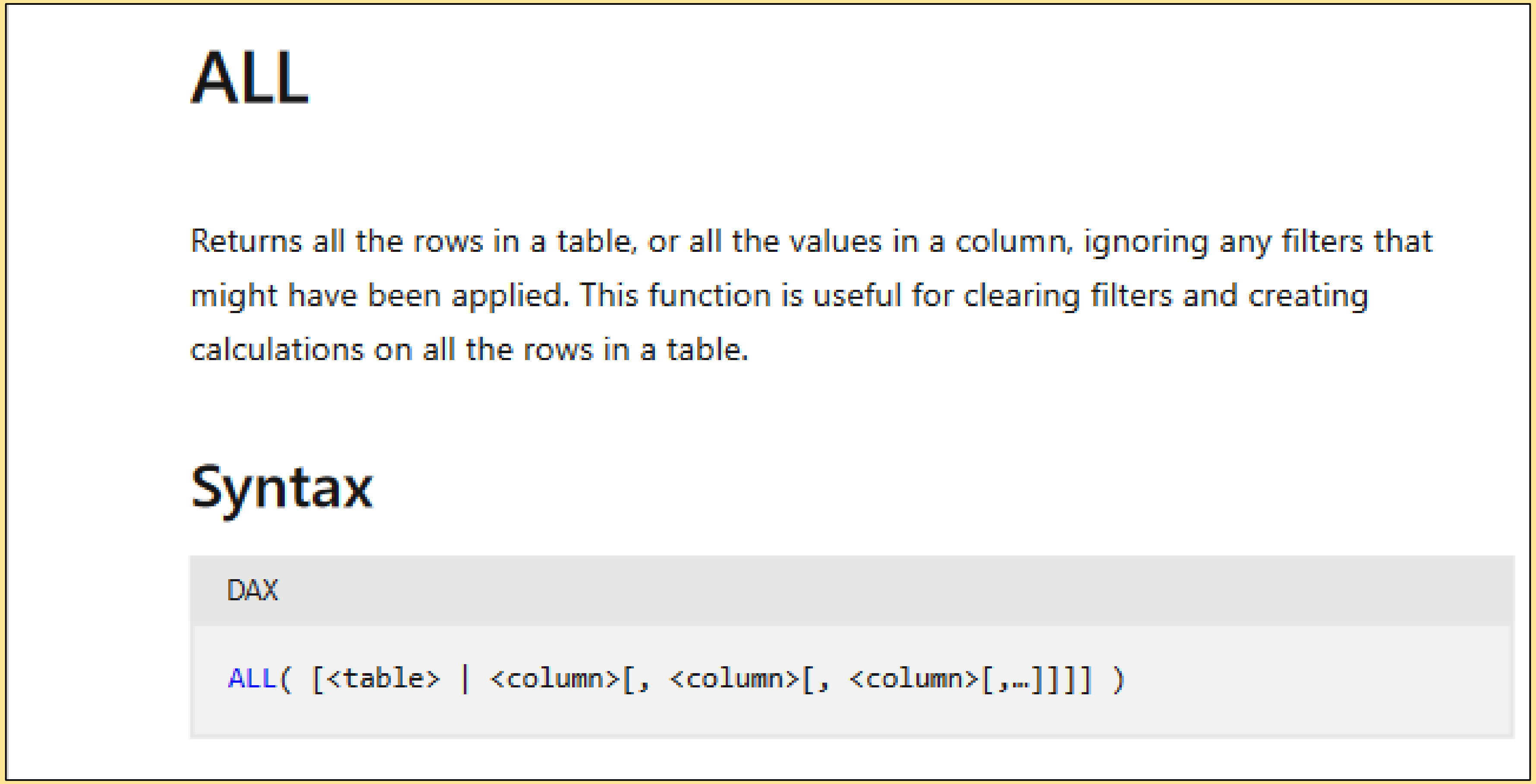
DATESYTD

Returns a table that contains a column of the dates for the year to date, in the current context.

Syntax

DAX

```
DATESYTD(<dates> [,<year_end_date>])
```



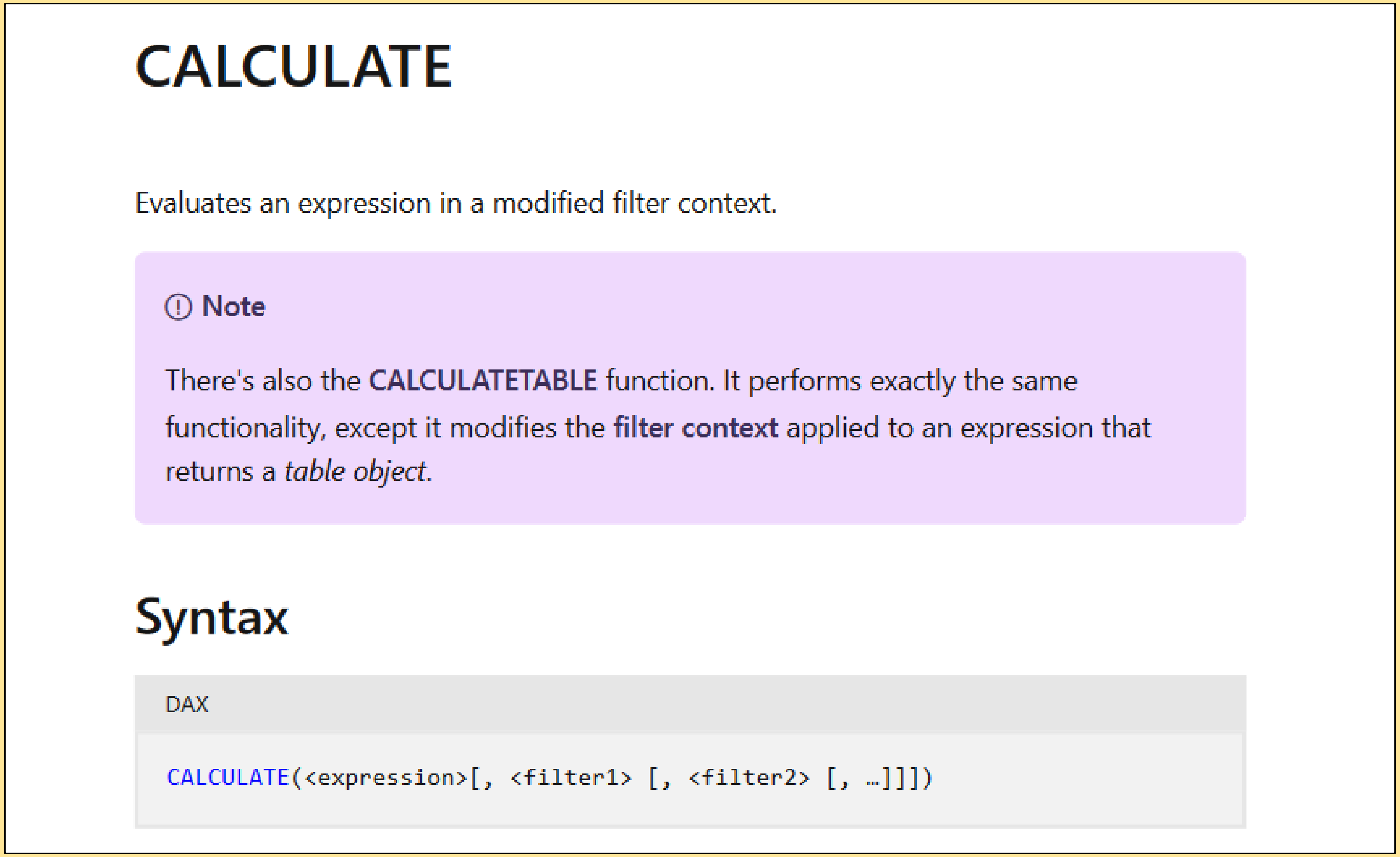
ALL

Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. This function is useful for clearing filters and creating calculations on all the rows in a table.

Syntax

DAX

```
ALL( [<table> | <column>[, <column>[, <column>[,…]]]] )
```



CALCULATE

Evaluates an expression in a modified filter context.

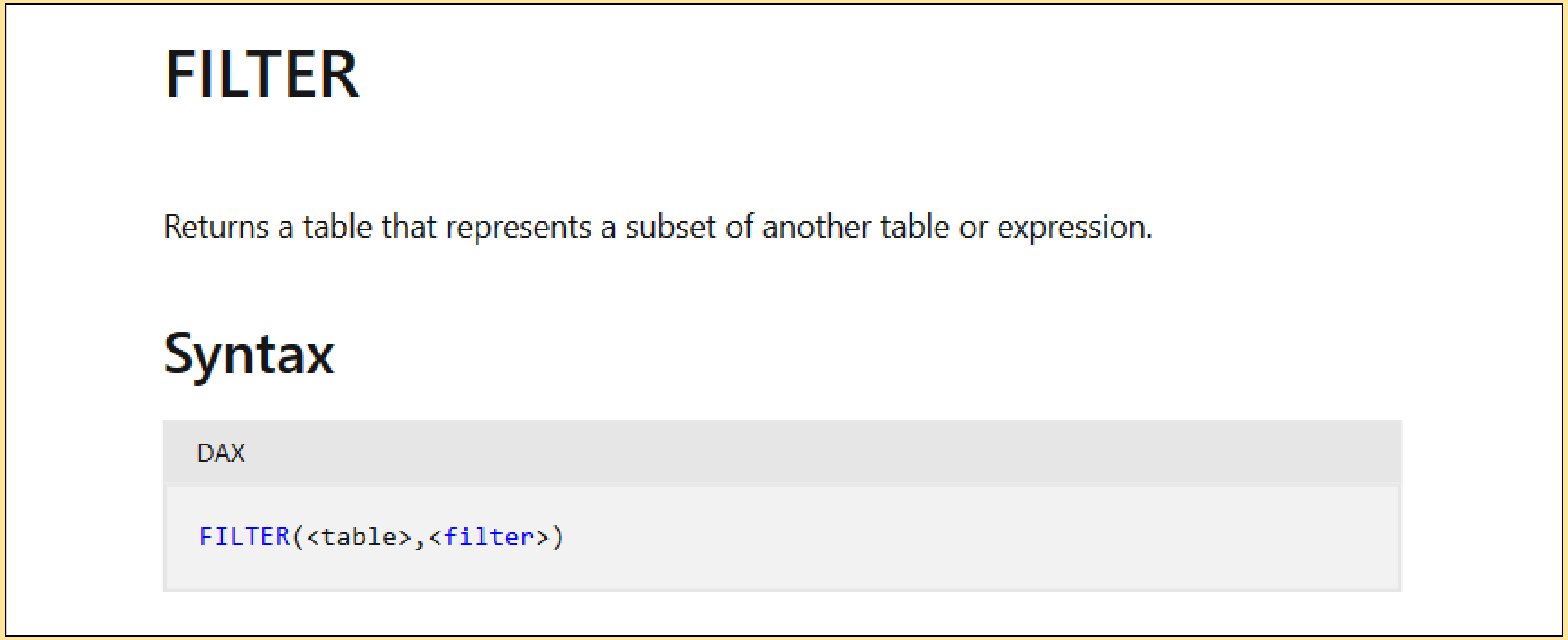
ⓘ **Note**

There's also the **CALCULATETABLE** function. It performs exactly the same functionality, except it modifies the **filter context** applied to an expression that returns a *table object*.

Syntax

DAX

```
CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])
```



FILTER

Returns a table that represents a subset of another table or expression.

Syntax

DAX

```
FILTER(<table>,<filter>)
```

THANK YOU

- MAYURI .D.