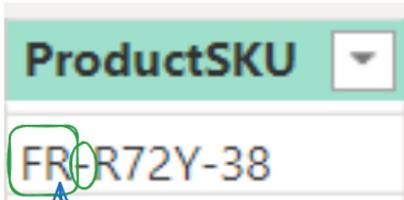


Power BI - DAX - p3

```
Month Name(Short) =
    UPPER(
        LEFT('Calendar Lookup'[Month Name] , 3)
    )
```



Left(ProductSKU, 2)

Search("-") -- index it return -1

dynamic

RLT-1233-K2

RLT-1233-K2

GLTK-1233-K2

5

ASSIGNMENT: TEXT FUNCTIONS

1. Update the Month Short column in the Calendar Lookup table to extract and capitalize the first 3 characters of the month name.

```
Month Short (DAX) =
    UPPER(
        LEFT(
            'Calendar Lookup'[Month Name] ,
            3))
    )
```

2. Create a new column in the Product Lookup table named SKU Category, to return any number of characters before the first hyphen in the ProductSKU column.

SKU Category =

LEFT(

'Product Lookup'[ProductSKU],

SEARCH(

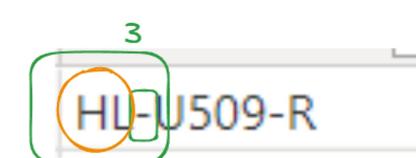
"-",

'Product Lookup'[ProductSKU]

)

-1

3



→ HL [SKU Category]

No. of character

LEFT(Text, [NumberOfCharacters])
Returns the specified number of characters from the start of a text string.
LEFT

3 index for the first "-"

	Day Name	Start of Week	Start of Month	Month Name	Day of Week
• Valid	100%	Valid	100%	Valid	100%
• Error	0%	Error	0%	Error	0%
• Empty	0%	Empty	0%	Empty	0%
	7 distinct, 0 unique	131 distinct, 0 unique	30 distinct, 0 unique	12 distinct, 0 unique	7 distinct, 0 unique
0 Wednesday		29-12-2019	01-01-2020	January	3
0 Thursday		29-12-2019	01-01-2020	January	4
0 Friday		29-12-2019	01-01-2020	January	5
0 Saturday		29-12-2019	01-01-2020	January	6
0 Sunday		05-01-2020	01-01-2020	January	0
0 Monday		05-01-2020	01-01-2020	January	1

Weekend = IF('Calendar Lookup'[Day of Week] IN {6,0} , "Weekend" , "Weekday")

Weekend = if('Calendar Lookup'[Day of Week] IN {6,0}, "Weekend", "Weekday")

BASIC DATE & TIME FUNCTIONS

TODAY/NOW

Returns the current date or exact time

=TODAY/NOW()

DAY/MONTH /YEAR

Returns the day of the month (1-31), month of the year (1-12), or year of a given date

=DAY/MONTH/YEAR(Date)

HOUR/MINUTE /SECOND

Returns the hour (0-23), minute (0-59), or second (0-59) of a given datetime value

=HOUR/MINUTE/SECOND(Datetime)

WEEKDAY/
WEEKNUM

Returns a weekday number from 1 (Sunday) to 7 (Saturday), or the week # of the year

=WEEKDAY/WEEKNUM(Date,
[ReturnType])

EOMONTH

Returns the date of the last day of the month, +/- a specified number of months

=EOMONTH(StartDate,
Months)

DATEDIFF

Returns the difference between two dates, based on a given interval (day, hour, year, etc.)

=DATEDIFF(Date1,
Date2, Interval)

2.

```
Weekend =  
    IF(  
        'Calendar Lookup'[Week Of Day] IN {6,7},  
        "Weekend",  
        "Weekday")
```

1

```
Week Of Day =  
    WEEKDAY(  
        'Calendar Lookup'[Date],  
        2)
```

ASSIGNMENT: DATE & TIME

1. Create a new column in the Customer Lookup table named Birth Year, to extract only the year from the BirthDate column

```
Birth Year =  
YEAR(  
    'Customer Lookup'[BirthDate]  
)
```

Merged Tables

Data > Search

- > Measure Table (DAX)
- > Calendar Lookup
- > Customer Lookup
- > Product Categories Lookup
- > Product Lookup
- > Product Subcategories Lookup
- > Returns Data
- > Sales Data 2020
- > Sales Data 2021
- > Sales Data 2022
- > Sales Record 2020-2022
 - CustomerKey
 - Σ Index
 - OrderDate
 - Σ OrderLineItem
 - OrderNumber
 - Σ OrderQuantity
 - Σ Product Lookup.ProductCost
 - Σ Product Lookup.ProductPrice
 - ProductKey
 - Quantity Type
 - StockDate
 - TerritoryKey

• Related Function : Calculated Column.

Create Retail Cost in Sales table.

Structure Formatting Properties

1 Retail Cost = RELATED('Product Lookup'[ProductCost])

	OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Quantity Type	Retail Cost
176	01-07-2020	05-03-2020	SO46680	371	21193	9	1	1	Single Item	\$1,320.68
184	03-07-2020	16-03-2020	SO46697	371	21173	9	1	1	Single Item	\$1,320.68
287	14-07-2020	10-05-2020	SO46797	371	21197	9	1	1	Single Item	\$1,320.68
348	22-07-2020	08-05-2020	SO46860	371	21180	9	1	1	Single Item	\$1,320.68
407	30-07-2020	10-05-2020	SO46917	371	21202	9	1	1	Single Item	\$1,320.68

Create Retail Price in Sales table.

Structure Formatting Properties

1 Retail Price = RELATED('Product Lookup'[ProductPrice])

Index	OrderDate	StockDate	OrderNumber	ProductKey	CustomerKey	TerritoryKey	OrderLineItem	OrderQuantity	Quantity Type	Retail Cost	Retail Price
1176	01-07-2020	05-03-2020	SO46680	371	21193	9	1	1	Single Item	\$1,320.68	\$2,181.5
1184	03-07-2020	16-03-2020	SO46697	371	21173	9	1	1	Single Item	\$1,320.68	\$2,181.5
1287	14-07-2020	10-05-2020	SO46797	371	21197	9	1	1	Single Item	\$1,320.68	\$2,181.5
1348	22-07-2020	08-05-2020	SO46860	371	21180	9	1	1	Single Item	\$1,320.68	\$2,181.5
1407	30-07-2020	10-05-2020	SO46917	371	21202	9	1	1	Single Item	\$1,320.68	\$2,181.5
1466	07-08-2020	20-04-2020	SO47127	371	21258	9	1	1	Single Item	\$1,320.68	\$2,181.5
1489	09-08-2020	20-06-2020	SO47141	371	21225	9	1	1	Single Item	\$1,320.68	\$2,181.5
1503	12-08-2020	15-07-2020	SO47162	371	21418	9	1	1	Single Item	\$1,320.68	\$2,181.5
1575	19-08-2020	28-04-2020	SO47230	371	21256	9	1	1	Single Item	\$1,320.68	\$2,181.5

Revenue = 'Sales Record 2020-2022'[Retail Price] * 'Sales Record 2020-2022'[OrderQuantity]

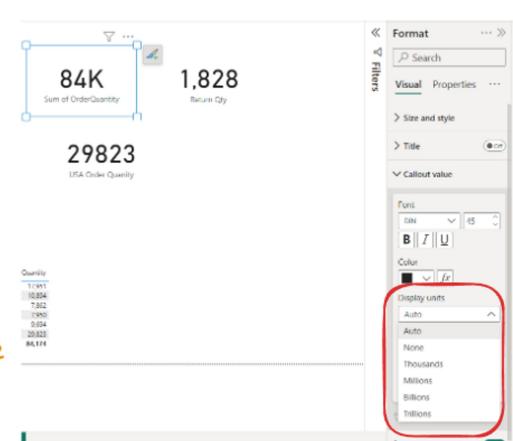
ne Help Table tools Column tools

Structure Formatting Properties Sort

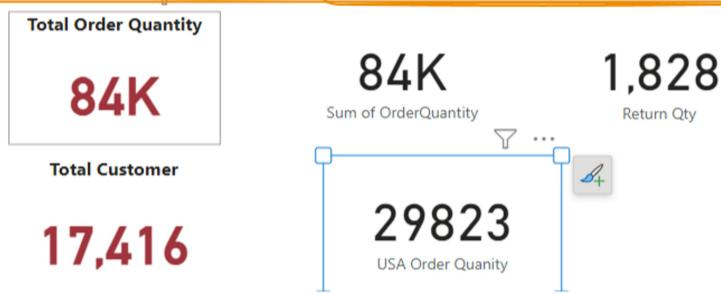
1 Revenue = 'Sales Record 2020-2022'[OrderQuantity] * RELATED('Product Lookup'[ProductPrice])

Country	Total Quantity
Australia	17,951
Canada	10,894
France	7,862
Germany	7,950
United Kingdom	9,694
United States	29,823
Total	84,174

Implicit Measure



```
USA Order Quantity = CALCULATE([Total Quantity], 'Territory Lookup'[Country] = "United States")
```



Calculate Function will play around existing measure and applying filter to produce complex Measure.

RELATED

RELATED() :-

Returns related values in each row of a table based on relationships with other tables

=RELATED(Column Name)

The column from a related table containing the values you want to retrieve

Examples:

- 'Product Lookup'[Product Name]
- 'Territory Lookup'[Country]

HEY THIS IS IMPORTANT!

- RELATED works like a VLOOKUP function in Excel – it uses the relationship between tables (defined by primary and foreign keys) to pull values from one table into a new column of another.
- Since this function requires row context, it can only be used as a calculated column or as part of an iterator function that cycles through all rows in a table (FILTER, SUMX, MAXX, etc.)

PRO TIP:

Instead of using RELATED to create extra columns (which increases file size), nest it within measures like FILTER or SUMX

```
Retail Price =  
    RELATED()  
        'Product Lookup'[ProductPrice])
```

```
Revenue =  
    'Sales Data'[Retail Price] * 'Sales Data'[OrderQuantity]
```

CALCULATE

CALCULATE()

Evaluates an expression in a context that is modified by filters

```
USA Revenue = CALCULATE([Revenue], 'Territory Lookup'[Country] = "United States")
```

=CALCULATE(Expression, [Filter1], [Filter2],...)

Name of an existing measure or a DAX formula
for a valid measure

Examples:

- [Total Orders]
- SUM('Returns Data'[Return Quantity])

A Boolean (True/False) expression or a table
expression that defines a filter.

Note: these require fixed values or aggregation
functions that return a scalar value (you cannot
create filters based on measures)

Examples:

- 'Territory Lookup'[Country] = "USA"
- Calendar[Year] <> MAX(Calendar[Year])

PRO TIP:

Think of CALCULATE as a filter modifier; it allows you to
overrule existing report filters and "force" new filter context

EXAMPLE: CALCULATE

X ✓ 1 Red Sales = `CALCULATE([Quantity Sold], 'Product Lookup'[Product Color] = "Red")`

- Here we've defined a new measure named Red Sales, which evaluates the Quantity Sold measure under a filter context where the product color is "Red"

Product Color	Quantity Sold	Red Sales
Black	10,590	4,011
Multi	5,756	4,011
Red	4,011	4,011
Silver	3,257	4,011
Total	23,614	4,011

Note how we see the same repeated values for each product color, and even the total!

HEY THIS IS IMPORTANT!

- The CALCULATE function modifies and overrules any competing filter context!
- In this matrix, the "Black" row has competing filter context: Product Color = Black (from the row label) and Product Color= "Red" (from the CALCULATE function)
- Both can't be true at the same time, so the "Red" filter from CALCULATE takes priority

CALCULATE

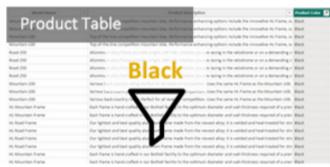
Filters are modified by CALCULATE
[Product Color] = "Red"

STEP 1

Filter context is detected
& applied

Product Color	Quantity Sold	Red Sales
Black	10,590	4,011
Red	4,011	4,011
Silver	3,257	4,011

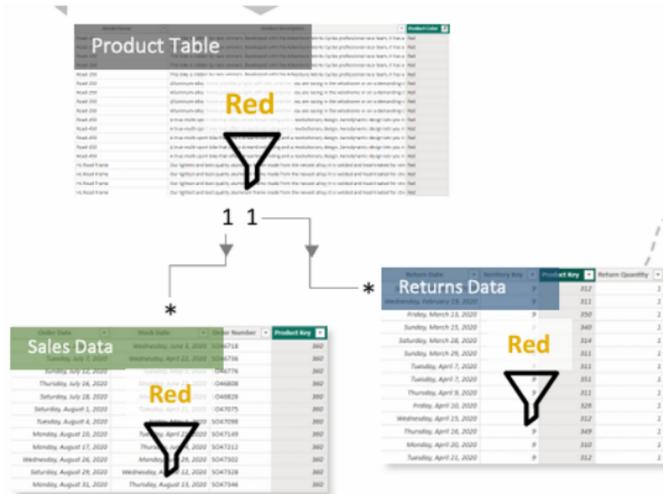
'Product Lookup'[Product Color] = "Black"



If the measure being evaluated contains a CALCULATE function, filter context is overwritten between Step 1 & Step 2

STEP 2

Filters flow "downstream"
to related tables



STEP 3

Measure evaluates against the filtered table

```
1 Quantity Sold =  
2 SUM( 'Sales Data'[Order Quantity] )
```

- Sum of the Order Quantity column in the Sales Data table, filtered to rows where the product color is "Red"

$$= 4,011$$

```

1 Bulk Orders =
2   CALCULATE(
3     [Total Orders],
4     'Sales Data'[OrderQuantity] > 1
5 )

```



```

Weekend Orders =
  CALCULATE(
    [Total Orders],
    'Calendar Lookup'[Weekend] = "Weekend")

```



ProductColor	Total Quantity
Black	10,590
Blue	3,779
Multi	5,756
NA	51,080
Red	4,011
Silver	3,257
White	1,063
Yellow	4,638
Total	84,174

Weekend	Total Quantity
Weekday	59,887
Weekend	24,287
Total	84,174

Total Quantity

X ✓ 1 Red Sales = CALCULATE([Quantity Sold], 'Product Lookup'[Product Color] = "Red")

Weekend Orders = CALCULATE([Total Quantity] , 'Calendar Lookup'[Weekend] = "Weekend")

Bulk Orders = CALCULATE([Total Quantity] , 'Sales Record 2020-2022'[OrderQuantity] > 1)