

## Power BI - DAX - p3

### 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]  
)
```

Age = DateDiff(BirthYear, Year(Today))

Birth Year	Age
1964	60
1964	60
1963	61
1974	50
1965	59
1963	61
1975	49
1976	48
1976	48
1974	50
1963	61
1963	61
1964	60
1964	60
1964	60

Age = DATEDIFF('Customer Lookup'[BirthDate], TODAY(), YEAR)

Age1 = YEAR(TODAY()) - 'Customer Lookup'[Birth Year]

## RELATED

RELATED() :-

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

=RELATED(Column Name)

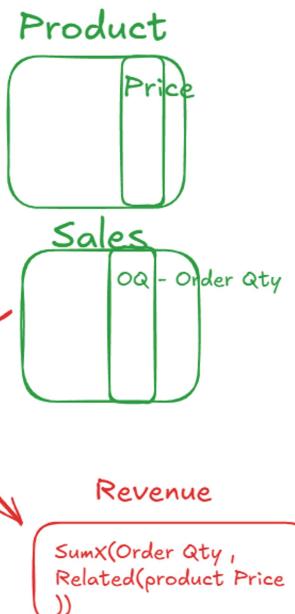
The column from a related table containing the values you want to retrieve

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

### Table - Sales Record 2020 - 2022

| Product Price = RELATED('Product Lookup'[ProductPrice])

Sales = 'Sales Record 2020-2022'[Product Price] \* 'Sales Record 2020-2022'[OrderQuantity]

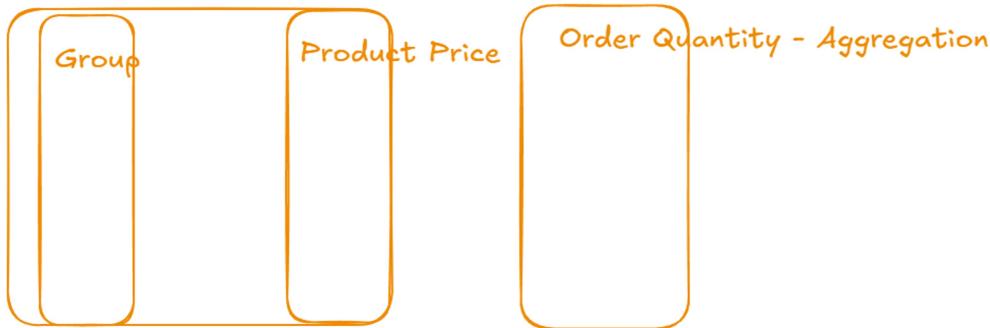
Revenue = SUM('Sales Record 2020-2022'[Sales])

But it takes lot of space.....

### Table Name

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

### Product Lookup



(o.q1 \* price1) + (o.q2 \* price2) + (... \* ....) + (... \* ...) + .....

Bulk Orders 1 = CALCULATE([Quantity Sold], 'Sales Record 2020-2022'[Quantity Type] = "Multiple Items")

## 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
<b>Total</b>	<b>23,614</b>	<b>4,011</b>

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"

Product Table
Black

Red

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

Product Table
Red

Sales Data	Returns Data
Red	Red

### 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 )
```

Filter

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

## DAX MEASURE TOTALS

Measure totals may seem incorrect or inconsistent depending on how they are calculated, because they don't simply add up the visible values in the report

```
1 Total Orders =  
2 DISTINCTCOUNT(  
3     'Sales Data'[Order Number]  
4 )
```

[Total Orders] counts distinct orders  
in the Sales Data table



Total Returns look right, but  
shouldn't Total Orders be 37,888??

Category Name	Total Returns	Total Orders
Accessories	1,115	16,983
Bikes	427	13,929
Clothing	267	6,976
<b>Total</b>	<b>1,809</b>	<b>25,164</b>

Order Date	Stock Date	Order Number	Product Key
Thursday, June 30, 2022	Thursday, April 07, 2022	S074140	568
Thursday, June 30, 2022	Friday, March 04, 2022	S074140	477
Thursday, June 30, 2022	Monday, May 30, 2022	S074140	223
Thursday, June 30, 2022	Friday, April 29, 2022	S074141	604
Thursday, June 30, 2022	Wednesday, May 04, 2022	S074141	471
Thursday, June 30, 2022	Monday, May 30, 2022	S074142	383
Thursday, June 30, 2022	Friday, March 18, 2022	S074142	490
Thursday, June 30, 2022	Tuesday, March 15, 2022	S074143	479
Thursday, June 30, 2022	Friday, April 08, 2022	S074143	606
Thursday, June 30, 2022	Tuesday, March 22, 2022	S074143	477
Thursday, June 30, 2022	Thursday, June 02, 2022	S074143	462
Thursday, June 30, 2022	Monday, April 25, 2022	S074144	574
Thursday, June 30, 2022	Sunday, April 24, 2022	S074144	220
Thursday, June 30, 2022	Monday, March 14, 2022	S074145	561
Thursday, June 30, 2022	Tuesday, June 14, 2022	S074146	584
Thursday, June 30, 2022	Friday, March 18, 2022	S074147	605
Thursday, June 30, 2022	Sunday, May 29, 2022	S074147	538
Thursday, June 30, 2022	Thursday, March 24, 2022	S074147	490

Order S074144 included two products: a bike and a helmet. That counts as 1 distinct order for the Total and 1 distinct order for BOTH Accessories & Bikes

With no filter context, there are 25,164 total distinct orders

Table: Sales Data (56,046 rows) Column: Order Numbe (25,164 distinct values)

#### PRO TIP:

Understand EXACTLY how your measures calculate and what they are designed to measure

## ASSIGNMENT: CALCULATE

1. Create a new measure named Bike Returns to calculate the total quantity of bikes returned.
2. Create a matrix to show Bike Returns (values) by Start of Month (rows). What do you notice about the volume of bike returns over time?
3. Create a new measure named Bike Sales to calculate the total quantity of bikes sold, and add it to the matrix. What do you notice?
4. Create a new measure named Bike Return Rate using either CALCULATE or DIVIDE, and add it to the matrix
5. How would you respond to the Product VP's concerns about rising bike returns?

```
Bike Returns = CALCULATE([Return Qty] , 'Product Categories Lookup'[CategoryName] = "Bikes")
```

```
Bikes Sold = CALCULATE([Quantity Sold] , 'Product Categories Lookup'[CategoryName] = "Bikes")
```

```
Bikes Return Rate = CALCULATE([Return Rate] , 'Product Categories Lookup'[CategoryName] = "Bikes")
```

```
Return Rate of Bikes = DIVIDE([Bike Returns] , [Bikes Sold])
```

Year	Bike Returns	Bikes Sold	Bikes Return Rate	Return Rate of Bikes
2020	86	2630	3.27%	3.27%
January	4	184	2.17%	2.17%
February	4	165	2.42%	2.42%
March	9	198	4.55%	4.55%
April	14	204	6.86%	6.86%
May	11	206	5.34%	5.34%
June	4	212	1.89%	1.89%
July	3	247	1.21%	1.21%
August	6	278	2.16%	2.16%
September	2	196	1.02%	1.02%
October	11	223	4.93%	4.93%
November	5	191	2.62%	2.62%
December	13	326	3.99%	3.99%
2021	172	5610	3.07%	3.07%
January	8	242	3.31%	3.31%
February	8	267	3.00%	3.00%
March	8	266	3.01%	3.01%
April	5	290	1.72%	1.72%
May	10	329	3.04%	3.04%
June	8	312	2.56%	2.56%
July	12	506	2.37%	2.37%
August	14	485	2.89%	2.89%
September	22	575	3.83%	3.83%
October	26	612	4.25%	4.25%
November	25	688	3.63%	3.63%
December	26	1038	2.50%	2.50%
2022	171	5689	3.01%	3.01%
January	14	766	1.83%	1.83%
February	22	806	2.73%	2.73%
March	27	888	3.04%	3.04%
April	38	956	3.97%	3.97%
May	36	1116	3.23%	3.23%
June	34	1157	2.94%	2.94%
Total	429	13929	3.08%	3.08%

Country	Total Orders	All Orders
Australia	6060	25164
Canada	3024	25164
France	2315	25164
Germany	2294	25164
United Kingdom	2771	25164
United States	8700	25164
<b>Total</b>	<b>25164</b>	<b>25164</b>

Country	Total Orders	All Orders	% of All Orders
Australia	6060	25164	24.08%
Canada	3024	25164	12.02%
France	2315	25164	9.20%
Germany	2294	25164	9.12%
United Kingdom	2771	25164	11.01%
United States	8700	25164	34.57%
<b>Total</b>	<b>25164</b>	<b>25164</b>	<b>100.00%</b>

Year	Total Orders	All Orders
2020	2630	25164
2021	10695	25164
2022	11839	25164
<b>Total</b>	<b>25164</b>	<b>25164</b>

Year	Total Orders	All Orders	% of All Orders
2020	2630	25164	10.45%
2021	10695	25164	42.50%
2022	11839	25164	47.05%
<b>Total</b>	<b>25164</b>	<b>25164</b>	<b>100.00%</b>

All Orders = `CALCULATE([Total Orders], ALL('Sales Record 2020-2022'))`

% of All Orders = `DIVIDE([Total Orders], [All Orders])`

num              den

## ALL

`ALL()` :-

Returns all rows in a table, or all values in a column, ignoring any filters that have been applied.

=`ALL(Table or Column, [Column2], [Column3],...)`

The table or column that you want to clear filters on

Examples:

- Transactions
- Products[Category]

Additional columns that you want to clear filters on (optional)

- Cannot specify columns if your first parameter is a table
- All columns must include the table name and come from the same table

Examples:

- 'Customer Lookup'[City], 'Customer Lookup'[Country]
- Products[Product Name]

### PRO TIP:

Instead of adding filter context, the `ALL` function removes it. This is often used in "% of Total" calculations, when the denominator needs to remain fixed regardless of filter context.

```

All Orders =
CALCULATE(
    [Total Orders],
    ALL(
        'Sales Data'))

```

```

% of All Orders =
DIVIDE(
    [Total Orders],
    [All Orders])

```

Overall Average Price =

```

CALCULATE(
    [Average Retail Price],
    ALL(
        'Product Lookup'))

```

Avg(Retail Price / Product Price)

CategoryName	Weekend Orders	Total Orders	All Orders	% of All Orders	Average Retail Price	Overall Average Price
Accessories	4,913	16,983	25,165	67.49%	\$34.26	\$714.44
Bikes	3,995	13,929	25,165	55.35%	\$1,541.38	\$714.44
Clothing	1,962	6,976	25,165	27.72%	\$50.68	\$714.44
Components			25,165		\$432.19	\$714.44
Total	7,214	25,165	25,165	100.00%	\$714.44	\$714.44

## ASSIGNMENT: CALCULATE & ALL

---

1. Create a new measure named All Returns to calculate the total number of returns, regardless of filter context
2. Create a new measure named % of All Returns that divides Total Returns by All Returns
3. Create a matrix to show % of All Returns (values) by product Category Name (rows). Which category accounts for the largest percentage of returns? The smallest?