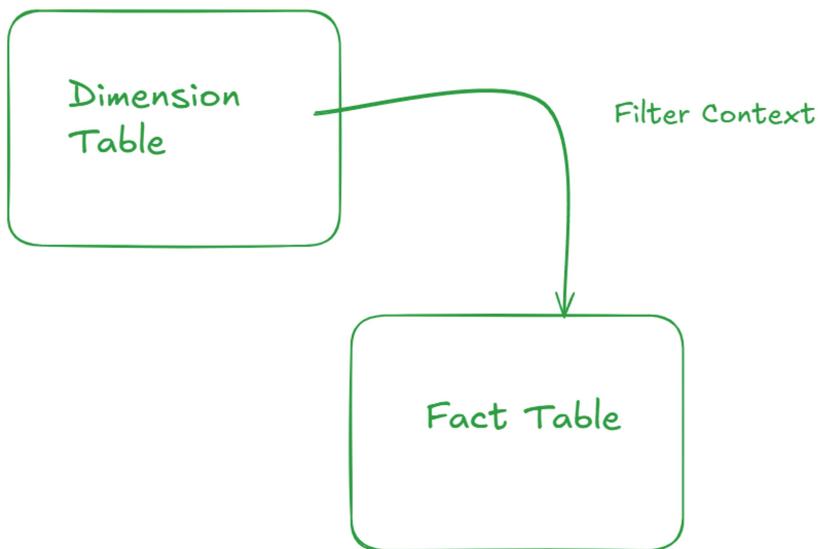


Advanced DAX Functions - p3



PRODUCTLINE	Total Revenue
Classic Cars	\$29,68,546.4
Motorcycles	\$9,71,086.29
Planes	\$8,77,942.21
Ships	\$6,77,940.4
Trains	\$2,03,804.26
Trucks and Buses	\$9,47,355.18
Vintage Cars	\$16,44,212.05
Total	\$82,90,886.79

I wanted to explore Function which ignores filter Context??

Just, Curious to know the contribution of each product in total Revenue??

ALL Function

→ To remove filter Context while calculating results. On simpler terms, it also means the value built on ALL Function would never split.

CALCULATE(Expression , Filter)



ALL('Something?') --> it will try to remove the filter

```
ALL([TableNameOrColumnName],
[ColumnName1], ...)
```

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

ALL Revenue \$% Format Currency

Measure Table (DA...) \$ % Auto

Structure Formatting

```

1 ALL Revenue =
2     CALCULATE(
3         [Total Revenue],
4         ALL('Vehicle Orders'))

```

Remove all the filters from Vehicle Orders Table.

PRODUCTLINE	Total Revenue	ALL Revenue
Classic Cars	\$29,68,546.4	\$82,90,886.79
Motorcycles	\$9,71,086.29	\$82,90,886.79
Planes	\$8,77,942.21	\$82,90,886.79
Ships	\$6,77,940.4	\$82,90,886.79
Trains	\$2,03,804.26	\$82,90,886.79
Trucks and Buses	\$9,47,355.18	\$82,90,886.79
Vintage Cars	\$16,44,212.05	\$82,90,886.79
Total	\$82,90,886.79	\$82,90,886.79

= \$2.9M / (2.9 + 0.97 + 0.87 + 0.67 + 2.03 + ...)

DIVIDE(Numerator, Denominator,
[AlternateResult])

Safe Divide function with ability to handle divide
by zero case.

Revenue Contribut... \$% Format Percentage

Measure Table (DA...) \$ % 2

Structure Formatting

```

1 Revenue Contribution % =
2     DIVIDE(
3         [Total Revenue],
4         [ALL Revenue],
5         "-")

```

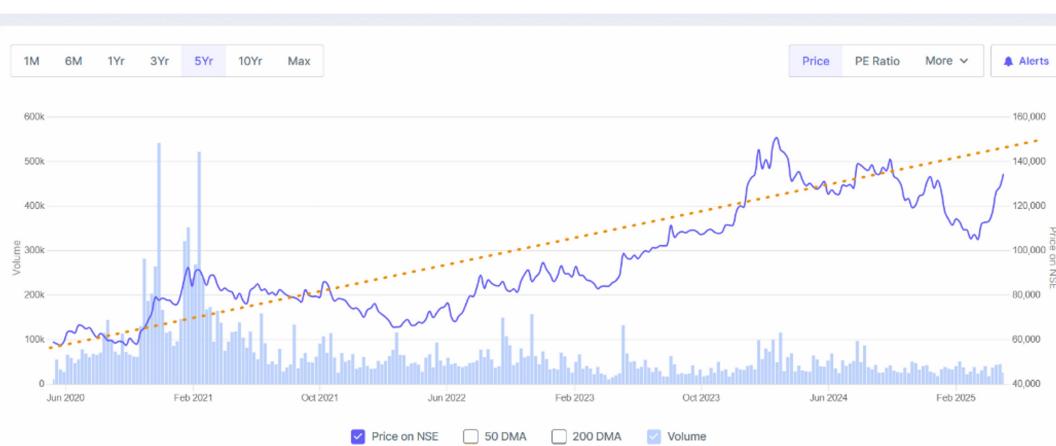
PRODUCTLINE	Total Revenue	ALL Revenue	Revenue Contribution %
Classic Cars	\$29,68,546.4	\$82,90,886.79	35.80%
Motorcycles	\$9,71,086.29	\$82,90,886.79	11.71%
Planes	\$8,77,942.21	\$82,90,886.79	10.59%
Ships	\$6,77,940.4	\$82,90,886.79	8.18%
Trains	\$2,03,804.26	\$82,90,886.79	2.46%
Trucks and Buses	\$9,47,355.18	\$82,90,886.79	11.43%
Vintage Cars	\$16,44,212.05	\$82,90,886.79	19.83%
Total	\$82,90,886.79	\$82,90,886.79	100.00%

PRODUCTLINE	Total Revenue	ALL Revenue	Revenue Contribution %
Classic Cars	\$29,68,546.4	\$82,90,886.79	36%
Vintage Cars	\$16,44,212.05	\$82,90,886.79	20%
Motorcycles	\$9,71,086.29	\$82,90,886.79	12%
Trucks and Buses	\$9,47,355.18	\$82,90,886.79	11%
Planes	\$8,77,942.21	\$82,90,886.79	11%
Ships	\$6,77,940.4	\$82,90,886.79	8%
Trains	\$2,03,804.26	\$82,90,886.79	2%
Total	\$82,90,886.79	\$82,90,886.79	100%

DEALSIZE	Total Revenue	ALL Revenue	Revenue Contribution %
Medium	\$5,58,912.72	\$82,90,886.79	7%
Small	\$3,20,573.57	\$82,90,886.79	4%
Large	\$91,600	\$82,90,886.79	1%
Total	\$9,71,086.29	\$82,90,886.79	12%

Time Intelligence Functions.

Trend Axis ~ Time.



TotalYTD()

--- QTD(), MTD()

[Cumulative results]

1. Total YTD Quantity.

```
TOTALYTD(Expression, Dates, [Filter], [YearEndDate])
```

Evaluates the specified expression over the interval which begins on the first day of the year and ends with the last date in the specified date column after applying specified filters.

```
1 OrderDateOnly =
2     DATE(
3         YEAR('Vehicle Orders'[ORDERDATETIME]),
4         MONTH('Vehicle Orders'[ORDERDATETIME]),
5         DAY('Vehicle Orders'[ORDERDATETIME]))
```

Year	Quantity Sold	Total YTD Quantity
2012	16567	16567
2013	25629	25629
2014	26498	26498
2015	17217	17217
2016	11371	11371
2017	1785	1785
Total	99067	1785

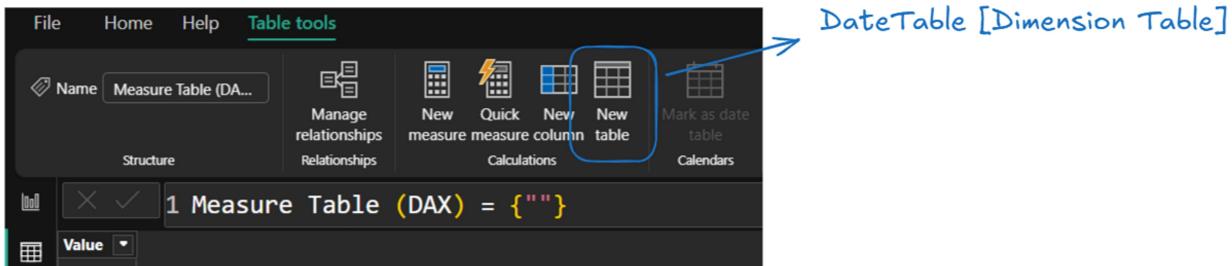
OrderDateOnly
19-05-2014
25-11-2014
13-10-2014
14-08-2016
28-08-2014
17-06-2012
15-05-2014
26-09-2013
12-03-2014
07-05-2014
09-05-2016
02-06-2012
13-09-2013
24-05-2013
28-03-2013
12-09-2012
04-11-2013
13-09-2013
21-10-2013
12-09-2016
02-11-2013
04-04-2015
25-07-2016

Year	Quantity Sold	Total YTD Quantity
2012	16567	16567
Qtr 1	1929	1929
January	501	501
1	22	22
3	46	46
5	42	42
11	28	28
13	34	34
15	54	54
16	21	21
19	82	82
23	27	27
25	42	42
29	62	62
31	41	41

Dates are not continuous.

Dates in order Date time may have duplicates and they are not continuous.

Solution - We have to create an alternate table having continuous date - Dimension Table.



CALENDAR(StartDate, EndDate)

Returns a table with one column of all dates between StartDate and EndDate.

```
1 DataTable = CALENDAR(
2     MIN('Vehicle Orders'[ORDERDATETIME]),
3     MAX(
4         fx MAX
5         fx MAXA
6         fx MAXX
7         fx SAMPLEAXISWITHLOCALMINMAX
8     )
9 )
```

Date

Properties

```
1 DataTable = CALENDAR(
2     MIN('Vehicle Orders'[ORDERDATETIME]),
3     MAX('Vehicle Orders'[ORDERDATETIME]))
```

Date

Vehicle Orders

- DEALSIZE
- Delay in Delivery
- DELIVERYDATETIME
- Discounted Price
- OrderDateOnly
- ORDERDATETIME
- ORDERLINENUMBER
- ORDERNUMBER
- DELIVERYTIME

Year	Quantity Sold	Total YTD Quantity
2012	16567	16567
Qtr 1	1929	1929
Qtr 2	3628	5557
Qtr 3	5278	10835
Qtr 4	5732	16567
2013	25629	25629
2014	26498	26498
2015	17217	17217
2016	11371	11371
2017	1785	1785
Total	99067	1785

Rows

Date	X
Year	X
Quarter	X
Month	X
Day	X

+Add data

Columns

+Add data

Values

Quantity Sold	X
Total YTD Qua...	X

Year	Quantity Sold	Total YTD Quantity
2012	16567	16567
Qtr 1	1929	1929
January	501	501
February	604	1105
March	824	1929
Qtr 2	3628	5557
April	952	2881
May	1144	4025
June	1532	5557
Qtr 3	5278	10835
July	1568	7125
August	1857	8982
September	1853	10835
Qtr 4	5732	16567
October	2024	12859
November	2050	14909
December	1658	16567
Total	99067	16567

```
Total QTD Quantity =
TOTALQTD(
    SUM('Vehicle Orders'[QUANTITYORDERED]),
    DateTable[Date])
```

Year	Quantity Sold	Total YTD Quantity	Total QTD Quantity
2012	16567	16567	5732
Qtr 1	1929	1929	1929
January	501	501	501
February	604	1105	1105
March	824	1929	1929
Qtr 2	3628	5557	3628
April	952	2881	952
May	1144	4025	2096
June	1532	5557	3628
Qtr 3	5278	10835	5278
July	1568	7125	1568
August	1857	8982	3425
September	1853	10835	5278
Qtr 4	5732	16567	5732
October	2024	12859	2024
November	2050	14909	4074
December	1658	16567	5732
2013	25629	25629	7663
Qtr 1	6158	6158	6158

Total MTD Quantity =

```
TOTALMTD(
    SUM('Vehicle Orders'[QUANTITYORDERED]),
    DateTable[Date])
```

DATESYTD, DATESQTD ,
DATESMTD → CALCULATE()

Total YTD Quantity =

```
CALCULATE(
    SUM('Vehicle Orders'[QUANTITYORDERED]),
    DATESYTD(DateTable[Date]))
```

Total QTD Quantity =

```
CALCULATE(
    SUM('Vehicle Orders'[QUANTITYORDERED]),
    DATESQTD(DateTable[Date]))
```

Total MTD Quantity =

```
CALCULATE(
    SUM('Vehicle Orders'[QUANTITYORDERED]),
    DATESMTD(DateTable[Date]))
```

DatesInPeriod

- Running Sum.

Sliding Window.

Jan , Feb , Mar , Apr , May , Jun ; Jul , Aug , Sept , Oct , Nov , Dec

Calculate the number of vehicle order from the last 3 months.

window size is of 3.

```
Orders in Last 3 Months =  
CALCULATE(  
    COUNTROWS('Vehicle Orders'),  
    DATESINPERIOD(  
        DateTable[Date],  
        MAX(DateTable[Date]),  
        -3,  
        MONTH))
```

Year	Orders in Last 3 Months	Count of ORDERNUMBER
2012	167	477
2013	218	737
2014	169	750
2015	100	491
2016	67	314
2017	4	54
Total	4	2823

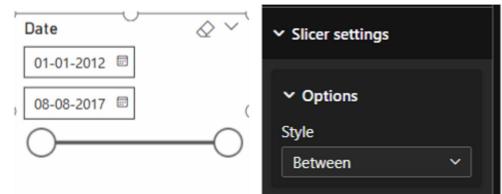
Oct - Dec

Year	Orders in Last 3 Months	Count of ORDERNUMBER
2012	167	477
Qtr 1	55	55
January	15	15
February	32	17
March	55	23
Qtr 2	106	106
April	69	29
1	55	1
2	55	
3	57	3
4	61	4
5	61	1
6	62	1
7	63	1
8	65	2
9	65	
10	66	1
11	65	
12	65	
13	65	1
14	66	1
15	64	
Total	52	2823

DatesBetween()

Total Orders placed in Quarter 1 2012 [1st Jan 2012 - 31st Mar 2012]

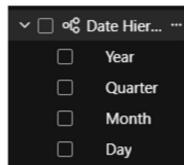
```
Total Orders on Quarter 1 2012 =
CALCULATE(
    COUNTROWS('Vehicle Orders'),
    DATESBETWEEN(
        DateTable[Date],
        DATE(2012,1,1),
        DATE(2012,3,31)))
```



55
Total Orders on Quarter 1 2012

Hierarchy

System Generated [Date]



If you have to create it manually,
Find the top player of the hierarchy.

File Home Help Table tools Column tools

Name CITY Data type Text

Format Text \$ % Auto

Summarization Don't summarize Data category City

Sort by column Sort Data groups Groups Manage relationships Relationships New column Calculations

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	ORDERDATETIME	DELIVERYDATETIME	STATUS	PRODUCTLINE	CITY
10203	20	\$100	8	19-05-2014 23:07:00	21-05-2014 11:19:59	Shipped	Classic Cars	Madrid
10153	20	\$100	11	25-11-2014 01:04:00	25-11-2014 07:56:06	Shipped	Classic Cars	Madrid
10104	34	\$100	1	13-10-2014 00:33:00	14-10-2014 15:46:41	Shipped	Classic Cars	Madrid
10153	42	\$100	12	14-08-2016 23:02:00	16-08-2016 17:27:35	Shipped	Classic Cars	Madrid
10212	39	\$100	16	28-08-2014 00:16:00	29-08-2014 15:53:17	Shipped	Classic Cars	Madrid
10104	41	\$100	9	17-06-2012 23:51:00	20-06-2012 23:46:07	Shipped	Trucks and Buses	Madrid
10246	46	\$100	5	15-05-2014 23:47:00	16-05-2014 09:08:01	Shipped	Trucks and Buses	Madrid
10412	54	\$100	5	26-09-2013 23:42:00	27-09-2013 02:26:05	Shipped	Trucks and Buses	Madrid
10203	47	\$100	5	12-03-2014 00:54:00	13-03-2014 17:57:38	Shipped	Classic Cars	Madrid
10212	33	\$100	15	07-05-2014 01:19:00	09-05-2014 18:31:03	Shipped	Classic Cars	Madrid
10212	29	\$100	10	09-05-2016 00:47:00	09-05-2016 02:47:22	Shipped	Classic Cars	Madrid
10205	36	\$100	2	02-06-2012 23:47:00	05-06-2012 04:30:35	Shipped	Vintage Cars	Madrid

Vehicle Orders CITY

Data category City

Uncategorized

Address

Place

City

County

State or Province

Postal code

Country

Continent

Latitude

Longitude

Web URL

Image URL

Barcode

Data Model Overview:

The screenshot shows the Power BI Data Model interface. It displays four main tables: DateTable, Vehicle Orders, Customers, and Sales Record. Relationships are established between DateTable and Vehicle Orders, and between Vehicle Orders and Customers. The Sales Record table is also visible.

Vehicle Orders Properties:

Properties
Is hidden: No
Uncategorized

- Address
- City
- Continent
- Country/Region
- County
- Latitude
- Longitude
- Place
- Summarize by: None
- Is nullable: Yes

Advanced Settings:

- Sort by column: COUNTRY (Default)
- Data category: Country/Region
- Summarize by: None

Vehicle Orders Advanced Settings:

Advanced

- Sort by column: POSTALCODE (Default)
- Data category: Postal code
- Summarize by: None

Vehicle Orders Properties:

Properties
Is hidden: No
Uncategorized

- Address
- City
- Continent
- Country/Region
- County
- Latitude
- Longitude
- Place
- Summarize by: None
- Is nullable: Yes

Vehicle Orders Advanced Settings:

Advanced

- Sort by column: STATE (Default)
- Data category: State or Province
- Summarize by: None

Vehicle Orders Fields:

- CITY
- CONTACTFIRSTNAME
- CONTACTLASTNAME
- COUNTRY
- DEALSIZE
- Delay in Delivery
- DELIVERYDATETIME
- Discounted Price
- OrderDateOnly
- POSTALCODE
- PRICEEACH
- PRODUCTLINE
- QUANTITYORDERED
- Sales Value
- STATE
- STATUS
- Target Delivery Date
- TERRITORY

Country > State > City

Power BI Field Well:

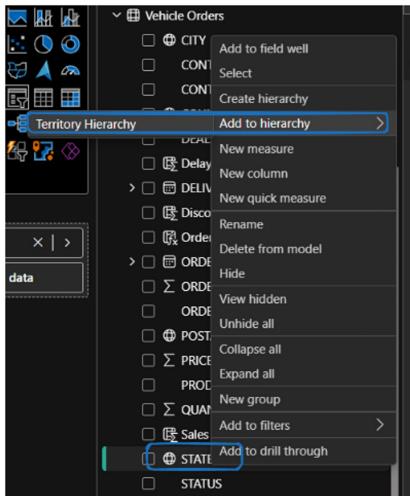
- Add to field well: COUNTRY
- Select
- Create hierarchy
- New measure
- New column
- New quick measure

Power BI Context Panel:

- COUNTRY Hierarchy
- COUNTRY

Power BI Rename Dialog:

Rename



drill Down will get activated
in the hierarchy

