

[Download from GitHub](#)

LazyDAX

V 4.20220619

Main

- Data Model
- Visuals
- DAX Functions
- Aggregation
- Date and Time
- Filter
- Information
- Logical
- Time Intelligence
- Text
- Statistical
- Relationship
- Parent Child
- Math
- Misc
- Index

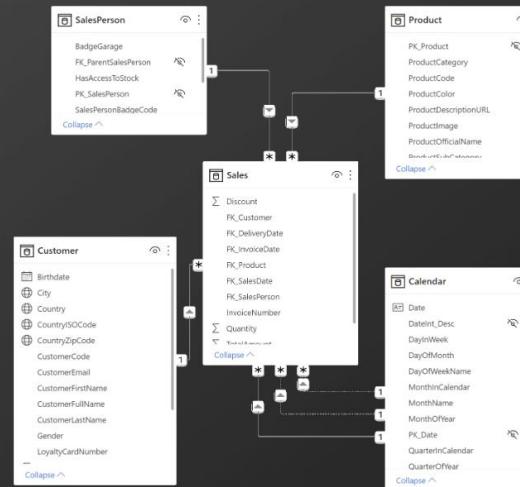


Arnaud Gastelblum (LinkedIn)

In this Power BI file you will be able to practice [DAX](#), debug your own expressions on a pretty [simple data model](#).

This model can be used for [presentation](#) or explanation.

Understandable data model



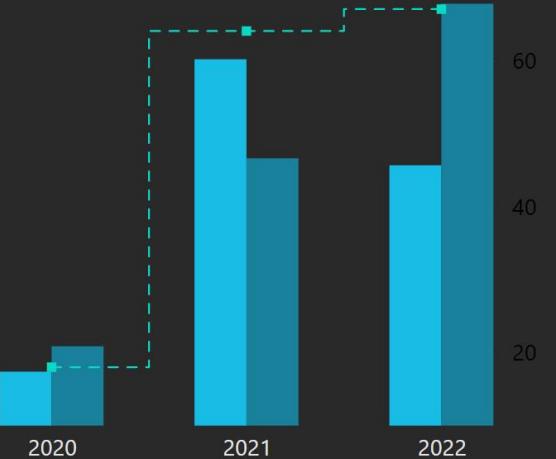
DAX Library

Fields

Search

- Measures Group
 - Date and Time functions
 - Filter functions
 - Logical functions
 - Math and Trig functions
 - Relationship functions
 - Statistical functions
 - Text functions
 - Time intelligence functions

Examples





A clean / easy / well thought data model will:

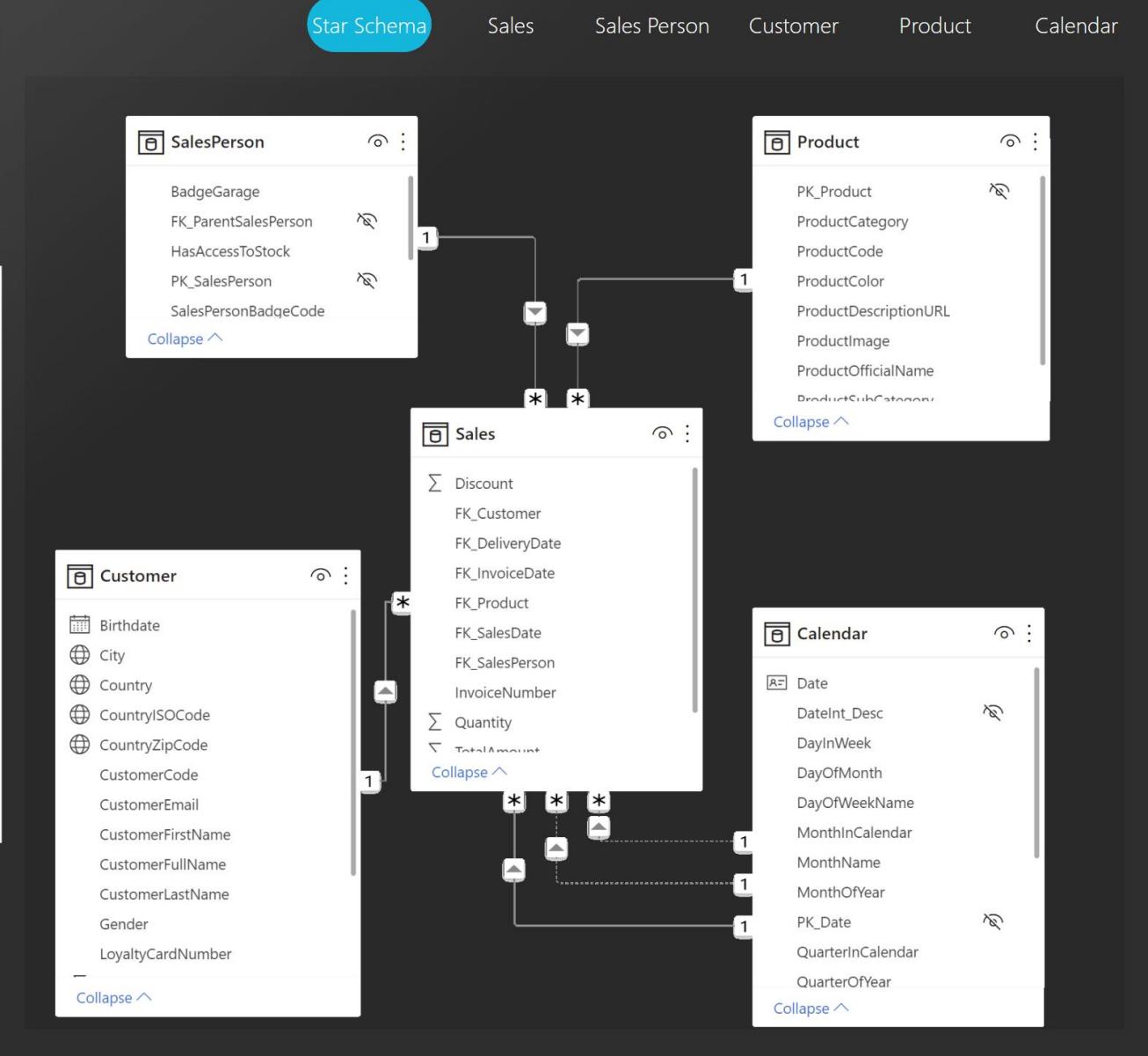
- Solve your problems
 - Reduce DAX complexity by 90%
 - Improve performance
 - Provide a smile to your users

Star schema

From Wikipedia, the free encyclopedia

In computing, the **star schema** is the simplest style of data mart schema and is the approach most widely used to develop data warehouses and dimensional data marts.^[1] The star schema consists of one or more **fact tables** referencing any number of **dimension tables**. The star schema is an important special case of the **snowflake schema**, and is more effective for handling simpler queries.^[2]

The star schema gets its name from the physical model's^[3] resemblance to a **star shape** with a fact table at its center and the dimension tables surrounding it representing the star's points.





Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

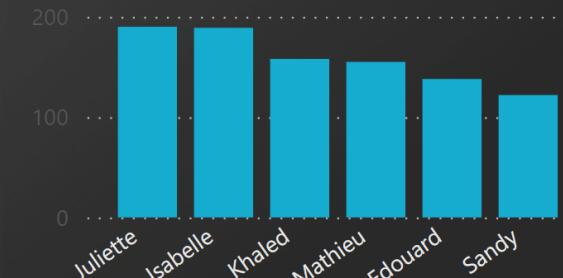
Math

Misc

Index

Sales
Σ Discount
FK_Customer
FK_DeliveryDate
FK_InvoiceDate
FK_Product
FK_SalesDate
FK_SalesPerson
InvoiceNumber
Σ Quantity
Σ TotalAmount
Σ UnitPrice

Quantity by SalesPerson



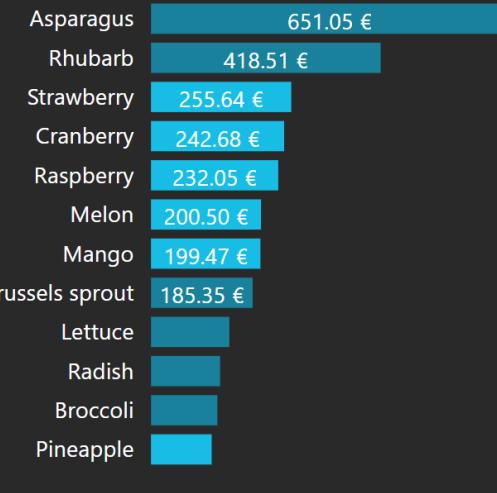
3.73K€

SUM TotalAmount

952

SUM Quantity

TotalAmount by Product

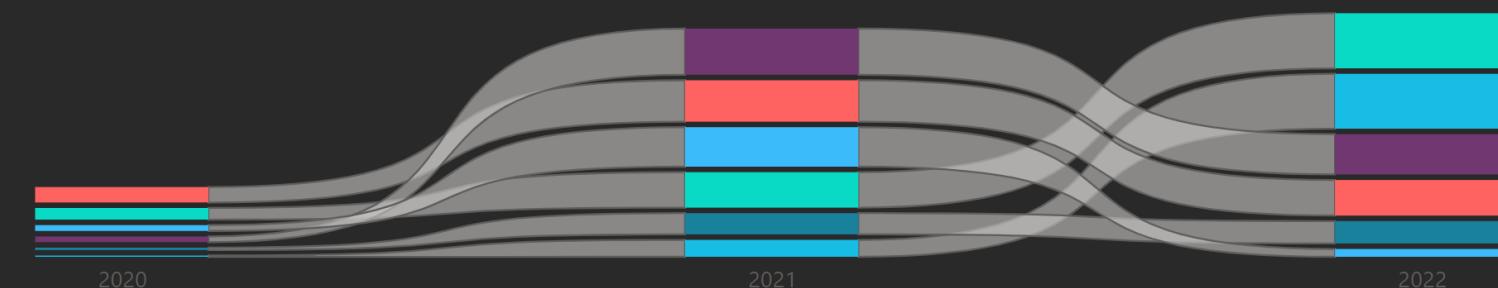


0 €

500 €

TotalAmount by Year and Sales Person

- Edouard
- Isabelle
- Juliette
- Khaled
- Mathieu
- Sandy





- Main
- Data Model
- Visuals
- DAX Functions
- Aggregation
- Date and Time
- Filter
- Information
- Logical
- Time Intelligence
- Text
- Statistical
- Relationship
- Parent Child
- Math
- Misc
- Index

- Customer
 - Address
 - Birthdate
 - CustomerCode
 - CustomerFirstName
 - CustomerFullName
 - CustomerLastName
 - Gender
 - LoyaltyCardNumber
 - LoyaltyPoints
 - NbChilds
 - RegisterDate

	Edouard	Clément
	Mathieu	Ricour
	Isabelle	VanCampenhoudt
	Khaled	Kahoul
	Sandy	Gram
	Juliette	Kool



Star Schema

Sales

Sales Person

Customer

Product

Calendar

Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

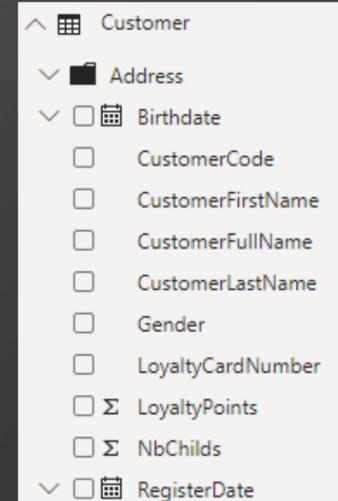
Relationship

Parent Child

Math

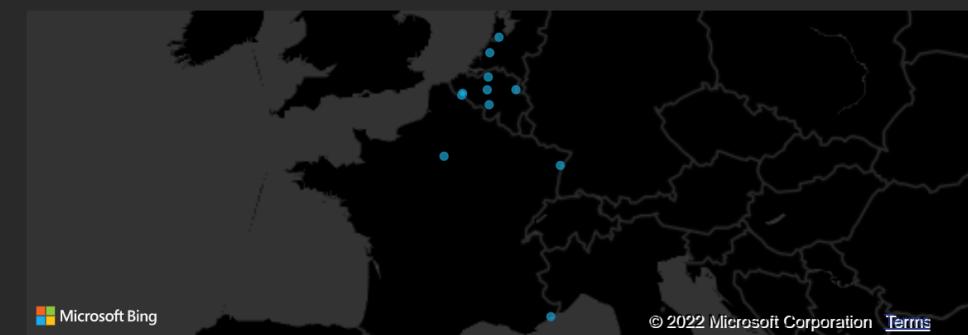
Misc

Index



CustomerCode	CustomerFirstName	CustomerLastName	Gender	LoyaltyCardNumber	LoyaltyPoints	NbChilds	RegisterDate	Birthdate
B42W912	Julien	Pomodoro	M	41715	2481	1	11/1/2018	11/27/1985
H59L252	Antoine	Legrand	M	32500	6622	3	10/13/2019	6/8/1984
I85S191	Sarah	Croche	F	51676	5454	0	11/18/2018	5/11/1959
K49A336	Amina	Loo	F	70750	8088	4	3/16/2019	10/23/1940
K74L961	Theresa	Limande	F	97997	9668	0	6/25/2017	6/12/1974
L75A698	Mike	Jeff	M	87895	6991	0	11/30/2018	12/12/1976
N79H709	Arnaud	Gastelblum	M	46843	3540	2	9/4/2017	4/9/1982
O30R794	Coralie	Brent	F	97747	6351	2	6/22/2017	4/20/1962
Q44B467	Bjorn	Bio	M	80955	4866	1	1/3/2019	8/23/1945
V17E452	Hilde	Vanderelst	F	56491	6954	0	12/25/2017	10/19/1969
Z91K849	Lisa	Dagusti	F	79338	4908	2	10/20/2017	11/28/1957
Z92R903	Pauline	Peanut	F	87260	3612	0	9/27/2018	6/23/1993

Latitude and Longitude





Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Product

- ProductCategory
- ProductCode
- ProductColor
- ProductDescriptionURL
- ProductImage
- ProductOfficialName
- ProductSubCategory
- ProductUnitPrice



ProductOfficialName	ProductUnitPrice	ProductColor	ProductCode	ProductCategory
Apple	€1.13	Red	APP	Fruit
Apricot	€2.2	Orange	APR	Fruit
Asparagus	€12.12	White	ASP	Vegetable
Banana	€2.04	Yellow	BAN	Fruit
Broccoli	€3.73	Green	BRO	Vegetable
Brussels sprout	€5.81	Green	BRU	Vegetable
Carrot	€1.79	Orange	CAR	Vegetable
Celery	€1.3	White	CEL	Vegetable
Cranberry	€11.34	Red	CRA	Fruit
Kale	€2.78	Green	KAL	Vegetable
Kiwifruit	€3.24	Green	KIW	Fruit
Lemon	€1.5	Yellow	LEM	Fruit
Lettuce	€5.95	Green	LET	Vegetable
Mango	€4.58	Yellow	MAN	Fruit
Melon	€4.93	Orange	MEL	Fruit
Onion	€0.8	Orange	ONI	Vegetable
Orange	€1.4	Orange	ORA	Fruit
Papaya	€1.95	Orange	PAP	Fruit
Peach	€3.88	Orange	PEA	Fruit
Pineapple	€2.55	Yellow	PIN	Fruit
Radish	€4.13	Pink	RAD	Vegetable
Raspberry	€7.32	Pink	RAS	Fruit
Rhubarb	€7.46	Pink	RHU	Vegetable
Strawberry	€10.52	Red	STR	Fruit
Tomato	€1.8	Red	TOM	Fruit





You can create a Calendar Dimension in Power Query. (By using the following M Code)

[More Information](#)

```
let CreateDataTable = (StartDate as date, EndDate as date, optional Culture as nullable text) as table =>
    let
        DayCount = Duration.Days(Duration.From(EndDate - StartDate)),
        Source = List.Dates(StartDate, DayCount, #duration(1,0,0,0)),
        TableFromList = Table.FromList(Source, Splitter.SplitByNothing()),
        ChangedType = Table.TransformColumnTypes(TableFromList,{{"Column1", type date}}),
        RenamedColumns = Table.RenameColumns(ChangedType,{{"Column1", "Date"}}),
        InsertYear = Table.AddColumn(RenamedColumns, "Year", each Date.Year([Date])),
        InsertQuarter = Table.AddColumn(InsertYear, "QuarterOfYear", each Date.QuarterOfYear([Date])),
        InsertMonth = Table.AddColumn(InsertQuarter, "MonthOfYear", each Date.Month([Date])),
        InsertDay = Table.AddColumn(InsertMonth, "DayOfMonth", each Date.Day([Date])),
        InsertDayInt = Table.AddColumn(InsertDay, "DateInt", each [Year] * 10000 + [MonthOfYear] * 100 + [DayOfMonth]),
        InsertMonthName = Table.AddColumn(InsertDayInt, "MonthName", each Date.ToText([Date], "MMMM", Culture), type text),
        InsertCalendarMonth = Table.AddColumn(InsertMonthName, "MonthInCalendar", each (try(Text.Range([MonthName],0,3)) otherwise [MonthName]) & " " & Number.ToText([Year])),
        InsertCalendarQtr = Table.AddColumn(InsertCalendarMonth, "QuarterInCalendar", each "Q" & Number.ToText([QuarterOfYear]) & " " & Number.ToText([Year])),
        InsertDayWeek = Table.AddColumn(InsertCalendarQtr, "DayInWeek", each Date.DayOfWeek([Date])),
        InsertDayName = Table.AddColumn(InsertDayWeek, "DayOfWeekName", each Date.ToText([Date], "dddd", Culture), type text),
        InsertWeekEnding = Table.AddColumn(InsertDayName, "WeekEnding", each Date.EndOfWeek([Date]), type date)
    in
        InsertWeekEnding
in
```



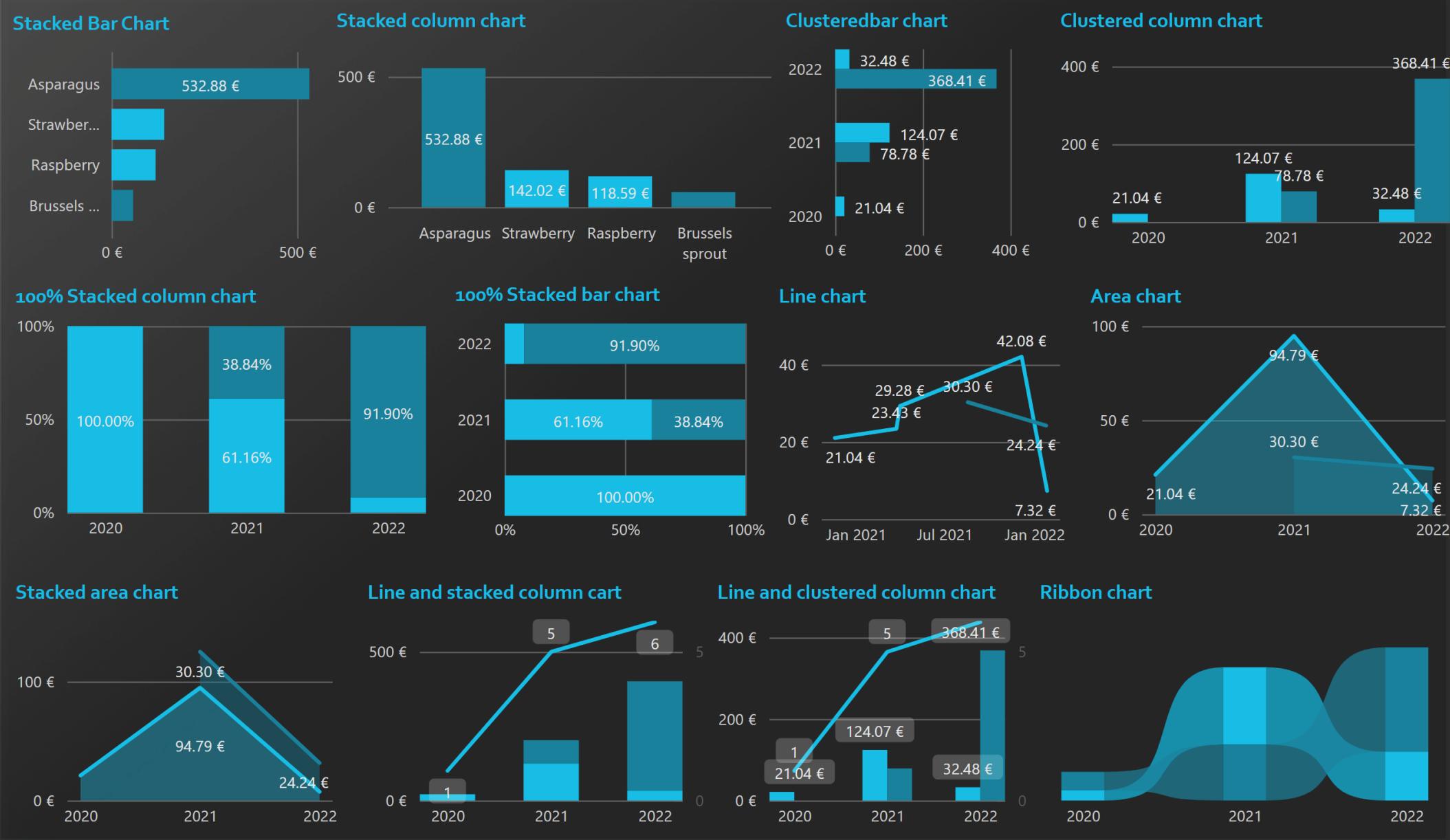
LazyDAX

V 4.20220619

Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Visuals

◀ ▷ Filters





Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Aggregation
functionsDate and Time
functions

Filter functions

Information
functions

Logical functions

Math and trig
functionsParent Child
functionsRelationship
functionsStatistical
function

Text functions

Time Intelligence
functions

Misc



- Main
- Data Model
- Visuals
- DAX Functions
- Aggregation**
- Date and Time
- Filter
- Information
- Logical
- Time Intelligence
- Text
- Statistical
- Relationship
- Parent Child
- Math
- Misc
- Index

AVERAGE	4.27
AVERAGEA	4.27
AVERAGEEX	14.80
AVERAGEX Error	15.07
AVERAGEX Articles by InvoiceNumber	1.94
COUNT	6
COUNTA	4
COUNTBLANK	2
COUNTROWS Customer	12
DISTINCTCOUNT Customer On Sales	12
DISTINCTCOUNTNOBLANK BadgeGarage	3
MAX Quantity	6
MAX 10 or 20	20
MIN Quantity	1
MIN 10 or 20	10
SUM TotalAmount	3,728.53 €
Semi Additive NbChild	15

Semi Additive AVERAGEX

InvoiceNumber	AVERAGEX Articles by InvoiceNumber	COUNTROWS Sales
33	1.00	1
34	1.00	1
35	1.00	1
36	1.00	1
37	2.00	2
38	1.00	1
39	3.00	3
40	2.00	2
41	1.00	1
42	1.00	1
43	1.00	1
44	5.00	5
45	1.00	1
Total	1.62	21



Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Today date and time

TODAY	6/21/2022
NOW	6/21/2022 9:31:42 AM
UTCNOW	6/21/2022 9:31:42 AM

Difference between dates

TODAY	6/21/2022
DATE	12/31/2022 12:00:00 AM
DATEDIFF Nb Days	193
DATEDIFF Nb Month	6
DATEDIFF Nb Year	0

Extract part

YEAR	2022
MONTH	6
DAY	21
HOUR	9
MINUTE	31
SECOND	42

Week Number

WEEKDAY	3
WEEKNUM Week Begin on Monday	26
WEEKNUM Week Begin on Sunday	26

Formatting Date

FORMAT Date General Date	6/21/2022
FORMAT Date ddMMYYYY	21/06/2022
DATEVALUE String To Date	1/12/2022 12:00:00 AM



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

FILTER() - ALL()

CustomerFullName	ProductCategory	SUM TotalAmount	Amount Fruit	Amount Fruit (All Product)
Pauline Peanut	Fruit	151.98 €	151.98	151.98
Pauline Peanut	Vegetable	85.42 €		151.98

FILTER()

Year	SUM TotalAmount	Amount Y-1	Amount Y-1 Filter	Amount Y-1 Filter 2
	692.61 €			
2020	160.74 €			
2021	746.78 €	160.74	160.74	160.74
2022	858.56 €	746.78	746.78	746.78

SAMEPERIODLASTYEAR() - PARALLELPERIOD()

Year	SUM TotalAmount	SAMEPERIODLASTYEAR	PARALLELPERIOD Y-1
	692.61 €		
2020	160.74 €		
2021	746.78 €	160.74	160.74
2022	858.56 €	746.78	746.78



Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Customer FullName

All



Customer FullName

Country	HASONEFILTER FullName	HASONEVALUE FullName	ISFILTERED FullName	ISCROSSFILTERED FullName	SELECTEDVALUE Country	SELECTEDVALUE FullName
Belgium					1 Belgium	No Name
Amina			1		1 Belgium	Amina Loo
Arnaud			1		1 Belgium	Arnaud Gastelblum
Bjorn			1		1 Belgium	Bjorn Bio
Lisa			1		1 Belgium	Lisa Dagusti
France					1 France	No Name
Julien			1		1 France	Julien Pomodoro
Pauline			1		1 France	Pauline Peanut
Sarah			1		1 France	Sarah Croche
Theresa			1		1 France	Theresa Limande
Total					1 No Country	No Name

Country

All

Customer FirstName

All

Customer FirstName

Country	HASONEFILTER FirstName	HASONEVALUE FirstName	ISFILTERED FirstName	ISCROSSFILTERED FirstName	SELECTEDVALUE Country	SELECTEDVALUE FullName
Belgium					1 Belgium	No Name
Amina	1	1	1	1	1 Belgium	Amina Loo
Arnaud	1	1	1	1	1 Belgium	Arnaud Gastelblum
Bjorn	1	1	1	1	1 Belgium	Bjorn Bio
Lisa	1	1	1	1	1 Belgium	Lisa Dagusti
France					1 France	No Name
Julien	1	1	1	1	1 France	Julien Pomodoro
Pauline	1	1	1	1	1 France	Pauline Peanut
Sarah	1	1	1	1	1 France	Sarah Croche
Theresa	1	1	1	1	1 France	Theresa Limande
Total					1 No Country	No Name



Main
 Data Model
 Visuals
 DAX Functions
 Aggregation
 Date and Time
 Filter
 Information
Logical
 Time Intelligence
 Text
 Statistical
 Relationship
 Parent Child
 Math
 Misc
 Index

Switch

If

Logical

Coalesce

Logical

Filters

```
SWITCH =
    VAR CurrentGender = FIRSTNONBLANK(Customer[Gender]; 0)
    RETURN
        SWITCH(
            CurrentGender
            ; "F"; "Women"
            ; "M"; "Men"
            ;      "Transgender"
        )
    )
```

CustomerFullName	Gender	SWITCH
Julien Pomodoro	M	Men
Pauline Peanut	F	Women

```
SWITCH Multi condition =
    // Using SWITCH() with TRUE() in the Expression
    // allow you to specifies mutliple Expression in Values fields
    // --> In this example: We can create a Range based on a variable
    VAR Sum_LoyaltyPoints = SUM(Customer[LoyaltyPoints])
    RETURN
        SWITCH(
            TRUE()
            // Case 1
            ;Sum_LoyaltyPoints < 3000 ; "Platinum"
            // Case 2
            ;AND(Sum_LoyaltyPoints >= 3000; Sum_LoyaltyPoints < 6500); "Silver"
            // Else
            ; "Gold"
        )
    )
```

CustomerFullName	LoyaltyPoints	SWITCH Multi condition
Julien Pomodoro	2481	Platinum
Pauline Peanut	3612	Silver



Text box_Switch

```
IF =  
    VAR CurrentGender = FIRSTNONBLANK(Customer[Gender]; 0)  
    RETURN  
        SWITCH(  
            CurrentGender  
            ; "F"; "Women"  
            ; "M"; "Men"  
            ; "Transgender"  
        )
```

CustomerFullName	Gender	IF
Pauline Peanut	F	Women
Julien Pomodoro	M	Not a Women

```
IF Multi condition =  
    // --> In this example: We can create a Range based on a variable  
    VAR Sum_LoyaltyPoints = SUM(Customer[LoyaltyPoints])  
    RETURN  
        IF(  
            // Case 1  
            Sum_LoyaltyPoints < 3000  
            , "Platinum"  
            ,  
            IF(  
                // Case 2  
                AND(Sum_LoyaltyPoints >= 3000, Sum_LoyaltyPoints < 6500)  
                , "Silver"  
                // Else  
                , "Gold"  
            )  
        )
```

CustomerFullName	LoyaltyPoints	IF Multi condition
Julien Pomodoro	2481	Platinum
Pauline Peanut	3612	Silver



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Switch If Logical Coalesce

Logical

ProductCategory	AND	OR	IN	NOT AND	NOT OR	IFERROR
Fruit	No	Yes	Yes	Yes	No	No
Vegetable	No	Yes	Yes	Yes	No	No



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Switch If Logical Coalesce

Logical

Filters

CustomerFirstName	COALESCE
Amina	133.32
Arnaud	37.52
Bjorn	75.32
Julien	139.61
Lisa	33.77
Pauline	58.18
Sarah	24.24
Theresa	48.48
Total	550.44



Period

TotalYTD

Year To Date

DateAdd

Calculation

Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Year	SUM TotalAmount	PARALLELPERIOD Y-1	PARALLELPERIOD M-1	SAMEPERIODLASTYEAR	% Evolution / Total This Year	% Evolution / Period Y-1	YTD % Evolution / Total This Year
2021	746.78 €	160.74	725.33	160.74	100.00%	464.59%	100.00%
January	51.63 €	160.74	63.41		6.91%		6.91%
February	63.44 €	160.74	51.63		8.50%		15.41%
March	76.44 €	160.74	63.44		10.24%		25.64%
April	111.42 €	160.74	76.44		14.92%		40.56%
May	57.10 €	160.74	111.42		7.65%		48.21%
June	74.33 €	160.74	57.10		9.95%		58.16%
July	39.02 €	160.74	74.33		5.23%		63.39%
August	73.07 €	160.74	39.02		9.78%		73.17%
September	34.68 €	160.74	73.07		4.64%		77.82%
October	65.67 €	160.74	34.68	46.00	8.79%	142.76%	86.61%
November	15.12 €	160.74	65.67	51.33	2.02%	29.46%	88.64%
December	84.86 €	160.74	15.12	63.41	11.36%	133.83%	100.00%
2022	858.56 €	746.78	882.82	746.78	100.00%	114.97%	100.00%
January	51.69 €	746.78	84.86	51.63	6.02%	100.12%	6.02%
February	125.91 €	746.78	51.69	63.44	14.67%	198.47%	20.69%
March	131.70 €	746.78	125.91	76.44	15.34%	172.29%	36.03%
April	29.21 €	746.78	131.70	111.42	3.40%	26.22%	39.43%
May	36.64 €	746.78	29.21	57.10	4.27%	64.17%	43.70%
June	32.10 €	746.78	36.64	74.33	3.74%	43.19%	47.43%
July	91.34 €	746.78	32.10	39.02	10.64%	234.09%	58.07%
August	168.35 €	746.78	91.34	73.07	19.61%	230.40%	77.68%
September	29.00 €	746.78	168.35	34.68	3.38%	83.62%	81.06%
October	35.07 €	746.78	29.00	65.67	4.08%	53.40%	85.14%
November	66.95 €	746.78	35.07	15.12	7.80%	442.79%	92.94%
December	60.60 €	746.78	66.95	84.86	7.06%	71.41%	100.00%
Total	1,605.34 €	907.52	1,608.15	907.52	186.98%	176.89%	100.00%



Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Period

TotalYTD

Year To Date

DateAdd

Calculation

Year		
Year	SUM TotalAmount	TOTALYTD
2022	858.56 €	858.56
2021	746.78 €	746.78
Total	1,605.34 €	858.56

Year - Quarter				
Year	SUM TotalAmount	TOTALYTD	TOTALYTD	Fiscal 31March
2022	858.56 €	858.56	549.26	
4	162.62 €	858.56	549.26	
3	288.69 €	695.94	386.64	
2	97.95 €	407.25	97.95	
1	309.30 €	309.30	864.57	
2021	746.78 €	746.78	555.27	
4	165.65 €	746.78	555.27	
3	146.77 €	581.13	389.62	
2	242.85 €	434.36	242.85	
1	191.51 €	191.51	352.25	
Total	1,605.34 €	858.56	549.26	

Year - Month

Year	SUM TotalAmount	TOTALYTD
2021	746.78 €	746.78
1	51.63 €	51.63
2	63.44 €	115.07
3	76.44 €	191.51
4	111.42 €	302.93
5	57.10 €	360.03
6	74.33 €	434.36
7	39.02 €	473.38
8	73.07 €	546.45
9	34.68 €	581.13
10	65.67 €	646.80
11	15.12 €	661.92
12	84.86 €	746.78
2022	858.56 €	858.56
1	51.69 €	51.69
2	125.91 €	177.60
3	131.70 €	309.30
4	29.21 €	338.51
5	36.64 €	375.15
6	32.10 €	407.25
7	91.34 €	498.59
8	168.35 €	666.94
9	29.00 €	695.94
10	35.07 €	731.01
11	66.95 €	797.96
12	60.60 €	858.56
Total	1,605.34 €	858.56



Period

TotalYTD

Year To Date

DateAdd

Calculation

Main
 Data Model
 Visuals
 DAX Functions
 Aggregation
 Date and Time
 Filter
 Information
 Logical
Time Intelligence
 Text
 Statistical
 Relationship
 Parent Child
 Math
 Misc
 Index

Year		SUM TotalAmount	DATEBETWEEN	Cumulate	DATESINPERIOD	DATESYTD	DATESQTD	DATESMTD	TOTALYTD	TOTALQTD
	⊖	692.61 €		1,766.08						
	⊖	692.61 €		1,766.08						
		692.61 €		1,766.08						
	⊖ 2020	160.74 €		160.74	160.74	160.74	160.74	63.41	160.74	160.74
	⊖ Q4 2020	160.74 €		160.74	160.74	160.74	160.74	63.41	160.74	160.74
	October	46.00 €		46.00	46.00	46.00	46.00	46.00	46.00	46.00
	November	51.33 €		97.33	97.33	97.33	97.33	51.33	97.33	97.33
	December	63.41 €		160.74	160.74	160.74	160.74	63.41	160.74	160.74
	⊖ 2021	746.78 €		907.52	746.78	746.78	165.65	84.86	746.78	165.65
	⊖ Q1 2021	191.51 €		352.25	352.25	191.51	191.51	76.44	191.51	191.51
	January	51.63 €		212.37	212.37	51.63	51.63	51.63	51.63	51.63
	February	63.44 €		275.81	275.81	115.07	115.07	63.44	115.07	115.07
	March	76.44 €		352.25	352.25	191.51	191.51	76.44	191.51	191.51
	⊖ Q2 2021	242.85 €		595.10	595.10	434.36	242.85	74.33	434.36	242.85
	April	111.42 €		463.67	463.67	302.93	111.42	111.42	302.93	111.42
	May	57.10 €		520.77	520.77	360.03	168.52	57.10	360.03	168.52
	June	74.33 €		595.10	595.10	434.36	242.85	74.33	434.36	242.85
	⊖ Q3 2021	146.77 €		741.87	741.87	581.13	146.77	34.68	581.13	146.77
	July	39.02 €		634.12	634.12	473.38	39.02	39.02	473.38	39.02
	August	73.07 €		707.19	707.19	546.45	112.09	73.07	546.45	112.09
	September	34.68 €		741.87	741.87	581.13	146.77	34.68	581.13	146.77
	⊖ Q4 2021	165.65 €		907.52	746.78	746.78	165.65	84.86	746.78	165.65
	October	65.67 €		807.54	761.54	646.80	65.67	65.67	646.80	65.67
	November	15.12 €		822.66	725.33	661.92	80.79	15.12	661.92	80.79
	December	84.86 €		907.52	746.78	746.78	165.65	84.86	746.78	165.65
	⊖ 2022	858.56 €		1,766.08	858.56	858.56	162.62	60.60	858.56	162.62
	⊖ Q1 2022	309.30 €		1,216.82	864.57	309.30	309.30	131.70	309.30	309.30
	January	51.69 €		959.21	746.84	51.69	51.69	51.69	51.69	51.69
	Total	2,458.69 €		1,766.08	858.56	858.56	162.62	60.60	858.56	162.62



Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Period

TotalYTD

Year To Date

DateAdd

Calculation

Display and summarize sales by year. (Add month in details)

Do not SUM sales for each SalesPerson when the sale date is older than the latest sale date of Khaled

SalesPersonFirstName Year	Khaled	Mathieu		Sandy
	SUM TotalAmount	TotalAmount until Khaled	SUM TotalAmount	TotalAmount until Khaled
2021	189.84 €	189.84	181.51 €	181.51
January	10.20 €	10.20	24.07 €	13.33
February	29.28 €	29.28		17.36 €
March	23.43 €	23.43	48.48 €	48.48
April	95.71 €	95.71	8.25 €	8.25
June	2.20 €	2.20	22.68 €	22.68
July				2.59 €
August			1.79 €	11.40 €
September				34.68 €
October	18.65 €	18.65	34.16 €	9.20 €
November				9.00 €
December	10.37 €	10.37	42.08 €	42.08
Total	189.84 €	189.84	181.51 €	181.51
				84.23 €
				84.23



Main

Data Model

Visuals

DAX Functions

Aggregation

Date and Time

Filter

Information

Logical

Time Intelligence

Text

Statistical

Relationship

Parent Child

Math

Misc

Index

Period

TotalYTD

Year To Date

DateAdd

Calculation

Display and summarize sales by year. (Add month in details)

Do not SUM sales for each SalesPerson when the sale date is older than the latest sale date of Khaled

SalesPersonFirstName Year	Khaled	Mathieu		Sandy		
	SUM TotalAmount	TotalAmount until Khaled	SUM TotalAmount	TotalAmount until Khaled	SUM TotalAmount	TotalAmount until Khaled
2021	189.84 €	189.84	181.51 €	181.51	84.23 €	84.23
January	10.20 €	10.20	24.07 €	13.33	17.36 €	12.96
February	29.28 €	29.28				
March	23.43 €	23.43	48.48 €	48.48		
April	95.71 €	95.71	8.25 €	8.25		
June	2.20 €	2.20	22.68 €	22.68		
July					2.59 €	
August			1.79 €		11.40 €	
September					34.68 €	
October	18.65 €	18.65	34.16 €		9.20 €	
November					9.00 €	
December	10.37 €	10.37	42.08 €	42.08		
Total	189.84 €	189.84	181.51 €	181.51	84.23 €	84.23



12.35K

[FORMAT to Integer](#)

\$ 1,234,568

[FORMAT Numeric](#)

33.00%

[FORMAT Percent](#)

No

[FORMAT Yes/No](#)

21/06/2022

[FORMAT Date ddMMYYYY](#)

6/21/2022

[FORMAT Date General Date](#)

VALUE("12345")

FORMAT(1234567,89; "\$ #,##0;(#,##0)")

FORMAT(0,33; "Percent")

FORMAT(0; "Yes/No")

FORMAT(TODAY(); "dd/MM/YYYY")

FORMAT(TODAY(); "General Date")



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence

Text

Statistical
Relationship
Parent Child
Math
Misc
Index

Format

Text

Concatenate

Unichar

Unicode

Text

CONCAT	ABCDEF
CONCAT (No function)	ABCDEF
UNICHAR	¼

LEN	12
LEFT	I am
MID	Snail
RIGHT	Snail
REPT	SnailSnailSnailSnailSnail

SUBSTITUTE	I am a Snail
TRIM	I am a Snail
UPPER	SNAIL
LOWER	snail

CONTAINSSTRING	Ok
CONTAINSSTRINGEXACT	Ko
CONTAINSSTRING Wild Card	Ok
EXACT	True
FIND	8



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Format

Text

ConcatenateX

Unichar

Unicode

Text

Filters

CONCATENATEX

arnaud@gastelblum.be; julien@pomodoro.fr; bjorn@bio.be; lisa@dagusti.be

CONCATENEX New Line
(Old)arnaud@gastelblum.be
julien@pomodoro.fr
bjorn@bio.be
lisa@dagusti.be

CONCATENATEX New Line

arnaud@gastelblum.be
julien@pomodoro.fr
bjorn@bio.be
lisa@dagusti.be



Format

Text

Concatenate

Unichar

Unicode

- Main
- Data Model
- Visuals
- DAX Functions
- Aggregation
- Date and Time
- Filter
- Information
- Logical
- Time Intelligence
- Text**
- Statistical
- Relationship
- Parent Child
- Math
- Misc
- Index

UNICHAR_91	[
UNICHAR_92	\
UNICHAR_93]
UNICHAR_94	^
UNICHAR_95	-
UNICHAR_96	,
UNICHAR_123	{
UNICHAR_124	
UNICHAR_125	}
UNICHAR_126	~
UNICHAR_127	

UNICHAR_65	A
UNICHAR_66	B
UNICHAR_67	C
UNICHAR_68	D
UNICHAR_70	F
UNICHAR_71	G
UNICHAR_72	H
UNICHAR_73	I
UNICHAR_74	J
UNICHAR_75	K
UNICHAR_76	L
UNICHAR_77	M
UNICHAR_78	N
UNICHAR_79	O
UNICHAR_80	P
UNICHAR_81	Q
UNICHAR_82	R
UNICHAR_83	S
UNICHAR_84	T
UNICHAR_85	U
UNICHAR_86	V
UNICHAR_87	W
UNICHAR_88	X
UNICHAR_89	Y
UNICHAR_90	Z

UNICHAR_97	a
UNICHAR_98	b
UNICHAR_99	c
UNICHAR_100	d
UNICHAR_101	e
UNICHAR_102	f
UNICHAR_103	g
UNICHAR_104	h
UNICHAR_105	i
UNICHAR_106	j
UNICHAR_107	k
UNICHAR_108	l
UNICHAR_109	m
UNICHAR_110	n
UNICHAR_111	o
UNICHAR_112	p
UNICHAR_113	q
UNICHAR_114	r
UNICHAR_115	s
UNICHAR_116	t
UNICHAR_117	u
UNICHAR_118	v
UNICHAR_119	w
UNICHAR_120	x
UNICHAR_121	y
UNICHAR_122	z



Format

Text

ConcatenateX

Unichar

Unicode

Main
 Data Model
 Visuals
 DAX Functions
 Aggregation
 Date and Time
 Filter
 Information
 Logical
 Time Intelligence
Text
 Statistical
 Relationship
 Parent Child
 Math
 Misc
 Index

UNICODE_33 !	33	UNICODE_65 A	65	UNICODE_97 a	97	UNICODE_123 {	123	UNICODE_188 ¼	188
UNICODE_34 "	34	UNICODE_66 B	66	UNICODE_98 b	98	UNICODE_124	124	UNICODE_189 ½	189
UNICODE_35 #	35	UNICODE_67 C	67	UNICODE_99 c	99	UNICODE_125 }	125	UNICODE_190 ¾	190
UNICODE_36 \$	36	UNICODE_68 D	68	UNICODE_100 d LowerCase	100	UNICODE_126 ~	126	UNICODE_191 ȝ	191
UNICODE_37 %	37	UNICODE_69 E	69	UNICODE_101 e LowerCase	101	UNICODE_127 ȝ	127	UNICODE_192 Å	192
UNICODE_38 &	38	UNICODE_70 F	70	UNICODE_102 f LowerCase	102	UNICODE_161 i	161	UNICODE_193 Á	193
UNICODE_39 '	39	UNICODE_71 G	71	UNICODE_103 g LowerCase	103	UNICODE_162 ¢	162	UNICODE_194 Â	194
UNICODE_40 (40	UNICODE_72 H	72	UNICODE_104 h LowerCase	104	UNICODE_163 £	163	UNICODE_195 Ã	195
UNICODE_41)	41	UNICODE_73 I	73	UNICODE_105 i LowerCase	105	UNICODE_164 ☰	164	UNICODE_196 Ä	196
UNICODE_42 *	42	UNICODE_74 J	74	UNICODE_106 j LowerCase	106	UNICODE_165 ¥	165	UNICODE_197 Å	197
UNICODE_43 +	43	UNICODE_75 K	75	UNICODE_107 k LowerCase	107	UNICODE_166 ¡	166	UNICODE_198 Æ	198
UNICODE_44 ,	44	UNICODE_76 L	76	UNICODE_108 l LowerCase	108	UNICODE_167 §	167	UNICODE_199 Ç	199
UNICODE_45 -	45	UNICODE_77 M	77	UNICODE_109 m LowerCase	109	UNICODE_168 ՞	168	UNICODE_200 Ě	200
UNICODE_46 .	46	UNICODE_78 N	78	UNICODE_110 n LowerCase	110	UNICODE_169 ©	169	UNICODE_201 É	201
UNICODE_47 /	47	UNICODE_79 O	79	UNICODE_111 o LowerCase	111	UNICODE_170 ՞	170	UNICODE_202 Ĕ	202
UNICODE_58 :	58	UNICODE_80 P	80	UNICODE_112 p LowerCase	112	UNICODE_171 «	171	UNICODE_203 ĕ	203
UNICODE_59 ;	59	UNICODE_81 Q	81	UNICODE_113 q LowerCase	113	UNICODE_172 ՞	172	UNICODE_204 ī	204
UNICODE_48 0	48	UNICODE_60 <	60	UNICODE_82 R	82	UNICODE_114 r LowerCase	114	UNICODE_205 í	205
UNICODE_49 1	49	UNICODE_61 =	61	UNICODE_83 S	83	UNICODE_115 s LowerCase	115	UNICODE_206 î	206
UNICODE_50 2	50	UNICODE_62 >	62	UNICODE_84 T	84	UNICODE_116 t LowerCase	116	UNICODE_207 ī	207
UNICODE_51 3	51	UNICODE_63 ?	63	UNICODE_85 U	85	UNICODE_117 u LowerCase	117	UNICODE_208 Đ	208
UNICODE_52 4	52	UNICODE_64 @	64	UNICODE_86 V	86	UNICODE_118 v LowerCase	118	UNICODE_209 Ñ	209
UNICODE_53 5	53	UNICODE_91 [91	UNICODE_87 W	87	UNICODE_119 w LowerCase	119	UNICODE_210 Ò	210
UNICODE_54 6	54	UNICODE_92 \	92	UNICODE_88 X	88	UNICODE_120 x LowerCase	120	UNICODE_211 Ó	211
UNICODE_55 7	55	UNICODE_94 ^	94	UNICODE_89 Y	89	UNICODE_121 y LowerCase	121	UNICODE_212 Ô	212
UNICODE_56 8	56	UNICODE_95 _	95	UNICODE_90 Z	90	UNICODE_122 z LowerCase	122	UNICODE_213 Õ	213
UNICODE_57 9	57	UNICODE_96 `	96			UNICODE_182 ¶	182	UNICODE_214 Ö	214
						UNICODE_183 ·	183	UNICODE_215 ×	215
						UNICODE_184 ,	184	UNICODE_216 Ø	216
						UNICODE_185 ¹	185	UNICODE_217 Ù	217
						UNICODE_186 °	186		
						UNICODE_187 »	187		



- Main
- Data Model
- Visuals
- DAX Functions
- Aggregation
- Date and Time
- Filter
- Information
- Logical
- Time Intelligence
- Text
- Statistical**
- Relationship
- Parent Child
- Math
- Misc
- Index

COUNT() - DISTINCTCOUNT()

CustomerFullName	COUNT Customer On Sales	DISTINCTCOUNT Customer On Customer
Amina Loo	25	1
Pauline Peanut	21	1
Total	46	2

COUNTROWS()

COUNTROWS Customer	COUNTROWS Product	COUNTROWS Sales
8	25	179



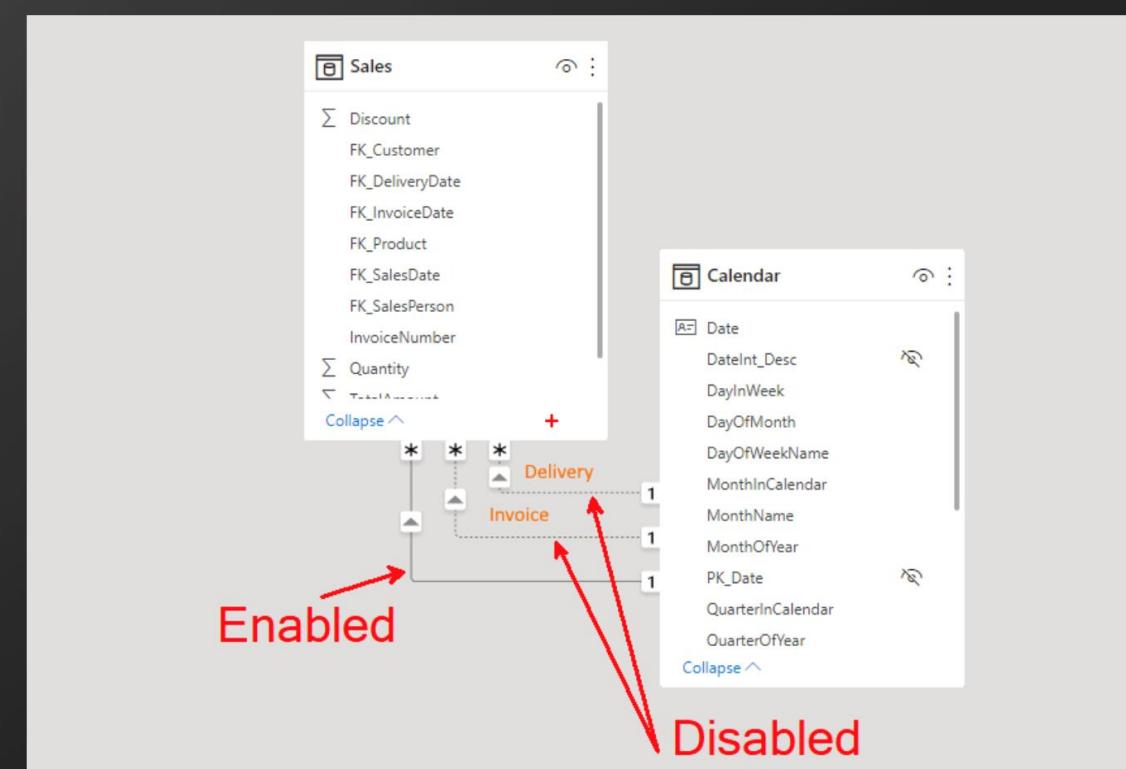
Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

USERELATIONSHIP()

InvoiceNumber	Date	SUM TotalAmount	USERELATIONSHIP InvoiceDate	USERELATIONSHIP DeliveryDate
129		1.50 €	1.50	1.50
Total		1.50 €	1.50	1.50

USERELATIONSHIP()

Specifies the relationship to be used in a specific calculation as the one that exists between columnName1 and columnName2.





Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Calculated Column: PATH | PATHITEM

PK_SalesPerson	SalesPersonFirstName	PATH	PATHITEM_1	PATHITEM_2	PATHITEM_3
1	Edouard	1	1		
3	Isabelle	1 3	1	3	
6	Juliette	1 3 6	1	3	6
4	Khaled	1 2 4	1	2	4
2	Mathieu	1 2	1	2	
5	Sandy	1 2 5	1	2	5

Calculated Column PATHLENGTH

SalesPersonFirstName	PATH	PATHLENGTH
Edouard	1	1
Mathieu	1 2	2
Isabelle	1 3	2
Khaled	1 2 4	3
Sandy	1 2 5	3
Juliette	1 3 6	3

Calculated Column: PATH | PATHITEMREVERSE

PK_SalesPerson	SalesPersonFirstName	PATH	PATHITEMREVERSE_1	PATHITEMREVERSE_2	PATHITEMREVERSE_3
1	Edouard	1	1		
3	Isabelle	1 3	3	1	
6	Juliette	1 3 6	6	3	1
4	Khaled	1 2 4	4	2	1
2	Mathieu	1 2	2	1	
5	Sandy	1 2 5	5	2	1

Calculated Column: LOKUPVALUE

lvl1_FirstName	lvl2_FirstName	lvl3_FirstName	SalesPersonFirstName	SalesPersonLastName	TotalAmount
Edouard			Edouard	Clément	322.65 €
	Isabelle		Isabelle	VanCampenhoudt	435.17 €
		Juliette	Juliette	Kool	488.53 €
		Mathieu	Mathieu	Ricour	474.67 €
			Khaled	Kahoul	533.45 €
			Sandy	Sandy	204.22 €



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

ABS	10	MROUND	9.90
ACOS	2.21	ODD	11
ACOSH	2.99	PI	3.14
ACOT	0.10	POWER	1,000.00
ACOTH	0.10	QUOTIENT	2
ASIN	-0.52	RADIANS	0.17
ASINH	3.00	RAND	0.26
ATAN	1.47	RANDBETWEEN	16
ATANH	-0.10	ROUND	7.70
CEILING	10.35	ROUNDDOWN	7.60
CONVERT	10	ROUNDUP	7.70
COS	-0.84	SIGN	1
COSH	11,013.23	SIN	-0.54
COT	1.54	SINH	11,013.23
COTH	1.00	SQRT	3.16
CURRENCY	10.00	SQRTPI	5.60
DEGREES	572.96	TAN	0.65
DIVIDE	5.00	TANH	1.00
EVEN	10	TRUNC	10
EXP	22,026.47		
FACT	3,628,800.00		
FLOOR	10.00		
GCD	2		
INT	10		
ISO.CEILING	10		
LCM	10		
LN	2.30		
LOG	1.00		
LOG10	1.00		

Random Numbers

RAND	0.70
RANDBETWEEN	55



Dump Filters

Fields Parameter

DAX is case insensitive

DAX in Description

Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Dump Filters

10 €

5 €

0 €

Pauline Peanut

ProductCategory

- Fruit
- Vegetable

DumpFilters

Add a tooltip in your graphic to help you understand the filter context of your calculation.

This Measure is auto generated by DAX Studio and should be added in your model

ProductColor

- Green
- Orange
- Pink
- Red
- Yellow

CustomerFullName

Pauline Peanut

SUM TotalAmount

58.86 €

DumpFilters

'Calendar'[Year] = 2019, 2020, 2021
Customer[Country] = France
Customer[CustomerFullName] = Pauline Peanut
'Product'[ProductCategory] = Fruit
'Product'[ProductColor] = Red, Green



Dump Filters

Fields Parameter

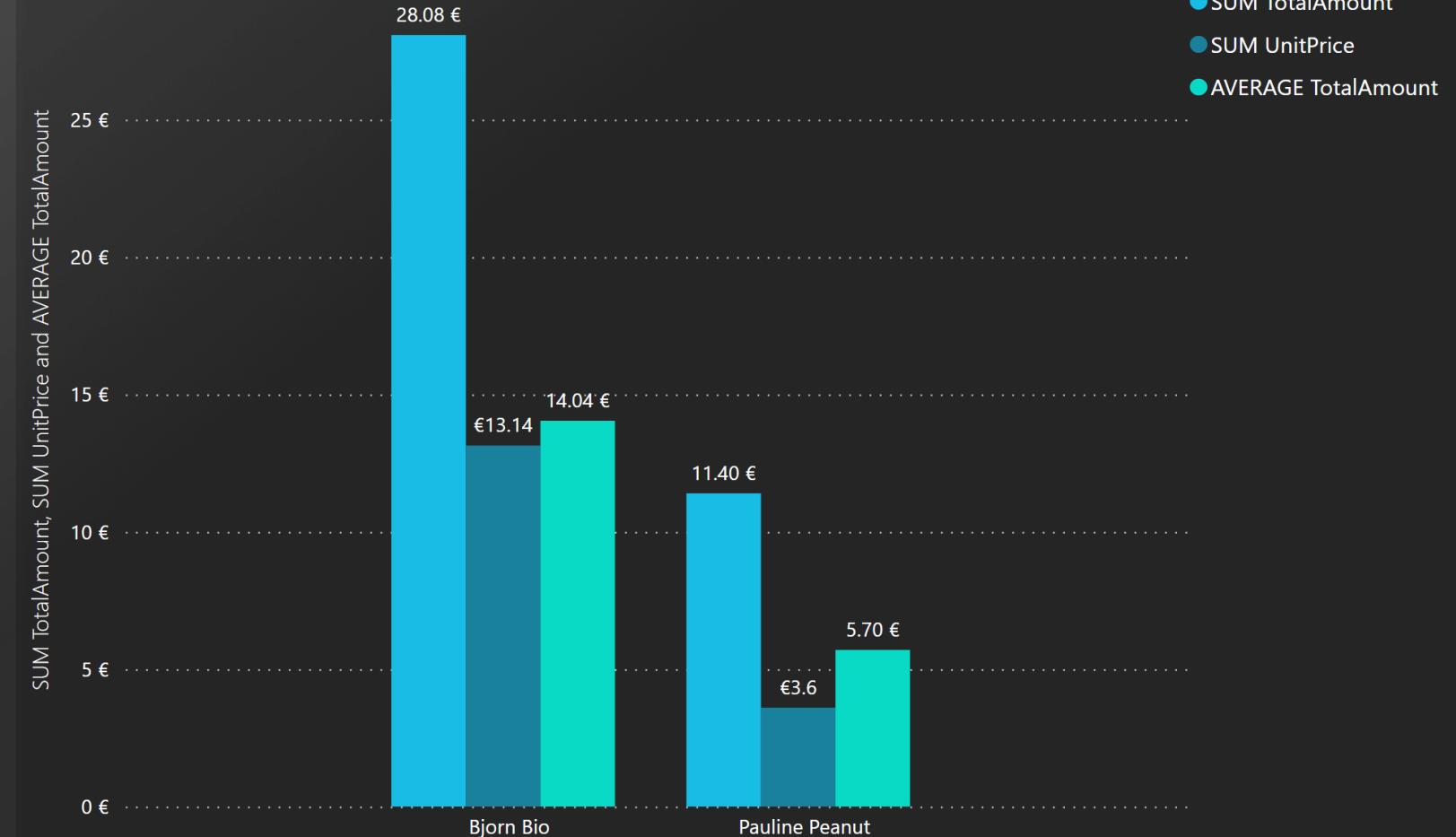
DAX is case insensitive

DAX in Description

Fields_Parameter

- SUM Discount
- SUM Quantity
- SUM TotalAmount
- SUM UnitPrice
- AVERAGE TotalAmount
- COUNTROWS Sales
- COUNTROWS Product

SUM TotalAmount, SUM UnitPrice, AVERAGE TotalAmount





Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

Dump Filters

Fields Parameter

DAX is case insensitive

Misc

Filters

ProductCategory

All

ProductCategory	Nb Vegetable	Nb VEGETABLE _	Nb VeGeTaBLE _
Vegetable	10	10	10
Total	10	10	10

DAX is Case Insensitive

Which means in this example that you can filter for **Vegetable** or **VEGETABLE** or **VeGeTaBLE**, the result will be the same

ProductCategory	ProductCode	Nb Product
Fruit	APP	1
Fruit	CRA	1
Fruit	KIW	1
Fruit	STR	1
Fruit	TOM	1
Total		5

```
Nb Vegetable =  
CALCULATE(  
    COUNTROWS('Product')  
    ,FILTER('Product', 'Product'[ProductCategory] = "Vegetable")  
)  
  
Nb VEGETABLE _ =  
CALCULATE(  
    COUNTROWS('Product')  
    ,FILTER('Product', 'Product'[ProductCategory] = "VEGETABLE")  
)  
  
Nb VeGeTaBLE _ =  
CALCULATE(  
    COUNTROWS('Product')  
    ,FILTER('Product', 'Product'[ProductCategory] = "VeGeTaBLE")  
)
```



Main
Data Model
Visuals
DAX Functions
Aggregation
Date and Time
Filter
Information
Logical
Time Intelligence
Text
Statistical
Relationship
Parent Child
Math
Misc
Index

With Tabular Editor, you can generate task by using C# code.
This example loop on all existing measure and store de DAX code into the description field.
On the Fields pane, when your mouse hover a measure, a tooltip shows you the DAX Expression

Name [COUNTROWS SalesPerson]
Description // Nb Rows in table SalesPerson
COUNTROWS (SalesPerson)

```
foreach (var m in Model.AllMeasures)
{
    m.Description = FormatDax(m.Expression);
}
```



Main
 Data Model
 Visuals
 DAX Functions
 Aggregation
 Date and Time
 Filter
 Information
 Logical
 Time Intelligence
 Text
 Statistical
 Relationship
 Parent Child
 Math
 Misc
 Index

Measure Name

- ✓ Date and Time functions
- ✓ Filter functions
- ✓ Logical functions
- ✓ Math and Trig functions
- ✓ Relationship functions
- ✓ Statistical functions\Count
- ✓ Text functions
- ✓ Text functions\Format
- ✓ Time intelligence functions
- ✓ Time intelligence functions\Evolution

DAX Code

MeasureName

Expression

% Evolution / Period Y-1

```
DIVIDE(
    [SUM TotalAmount]
    , [SAMEPERIODLASTYEAR]
)
```

% Evolution / Total This Year

```
VAR CurrentYear = MAX('Calendar'[Year])
VAR CurrentYearTotalAmount =
CALCULATE(
    [SUM TotalAmount]
    , FILTER( ALL('Calendar') , 'Calendar'[Year] = CurrentYear)
)
// Also works with:
// CALCULATE(
//     [SUM TotalAmount]
//     ; 'Calendar'[Year] = CurrentYear
//     ; ALL('Calendar')
// )
RETURN
DIVIDE(
    [SUM TotalAmount]
    , CurrentYearTotalAmount
)
```

Amount Fruit

```
CALCULATE(
    SUM(Sales[TotalAmount])
    , FILTER('Product' , 'Product'[ProductCategory] = "Fruit")
)
```

Amount Fruit (All Product)

```
CALCULATE(
    CALCULATE(
        SUM(Sales[TotalAmount])
        , FILTER('Product' , 'Product'[ProductCategory] = "Fruit")
    )
    , ALL('Product')
)
```

Amount Y-1

```
VAR CurrentYear = MAX('Calendar'[Year])
RETURN
CALCULATE(
    [SUM TotalAmount]
```