Microsoft

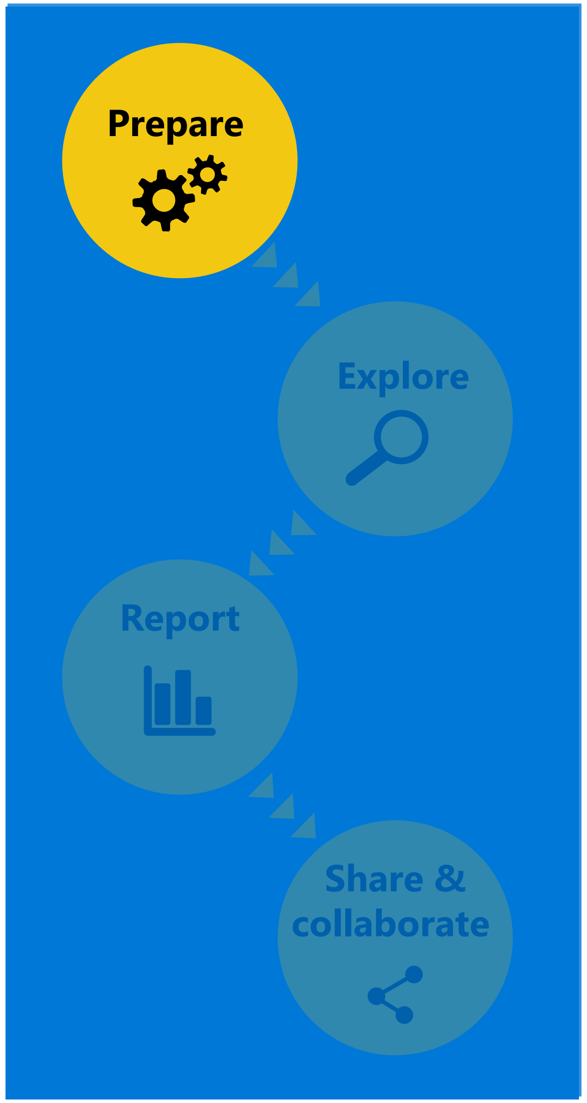
Business Analytics with Power BI

Microsoft Services

Module 1: Power BI Desktop

Lesson 5: Calculations

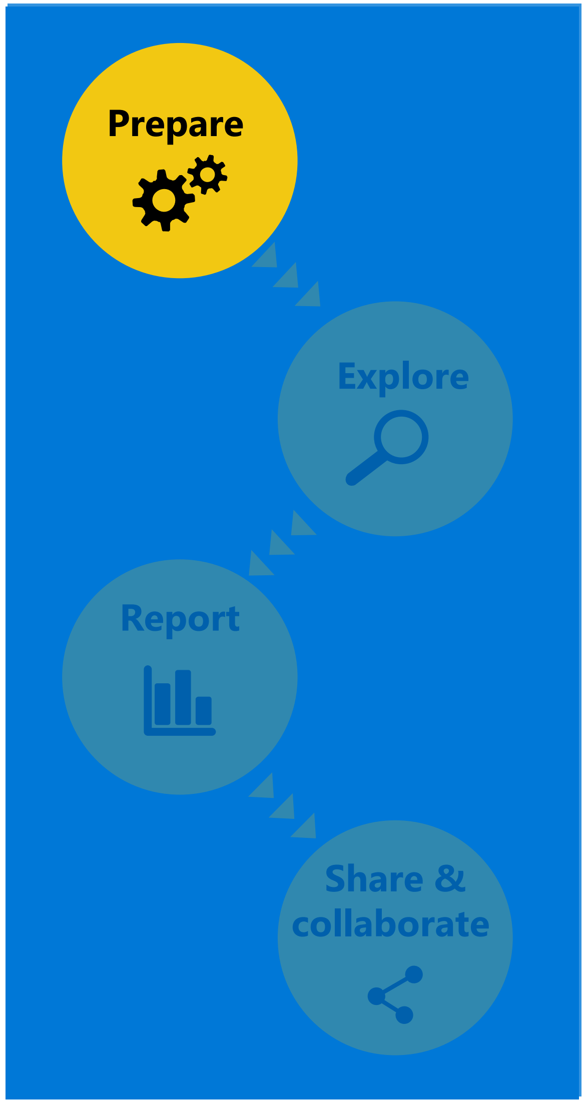
Calculations



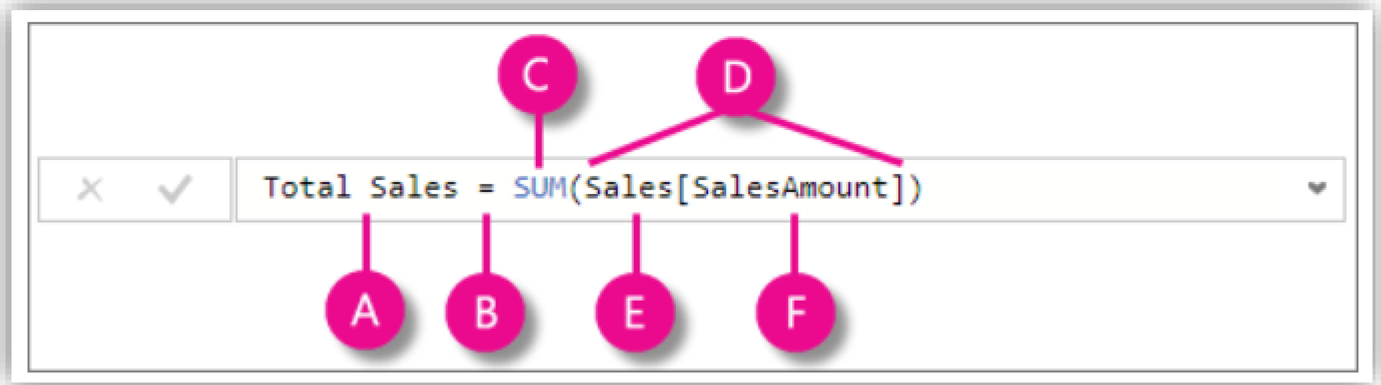
DAX Basics

- **Microsoft Excel like Language** that allows the **extension** of the model with additional business logic
- Ideal for **complex calculations** such as time intelligence, growth formulas, ratios, and more complex KPIs
- **Business rules** can be **materialized at the row level (Calculated Column)** or can **be calculated on-the-fly (Measures)**
- Certain capabilities **might overlap with Formula Language**

Calculations



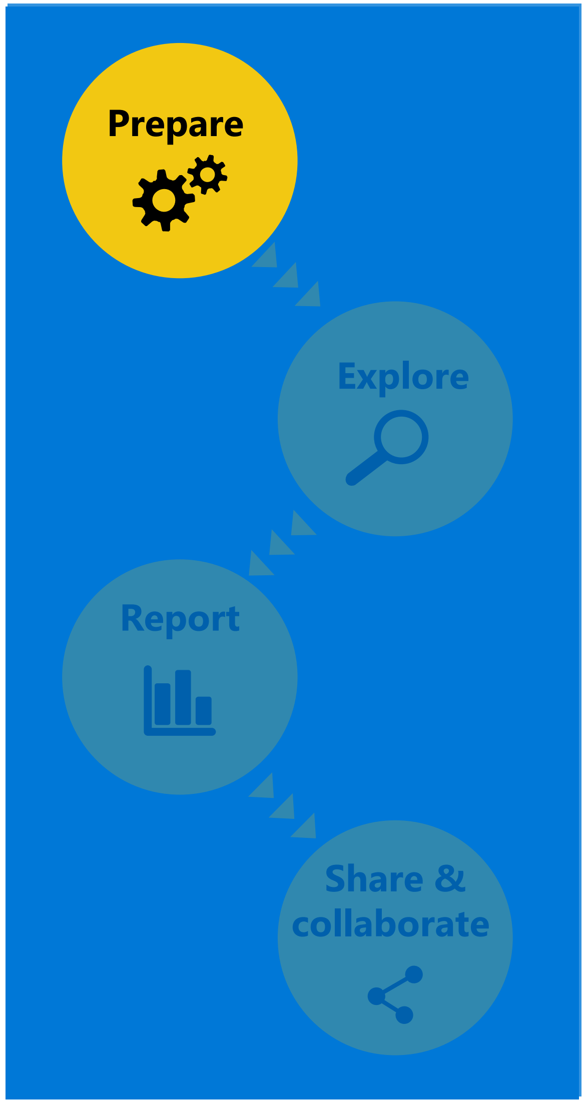
DAX Basics - Syntax



```
Total Sales = SUM(Sales[SalesAmount])
```

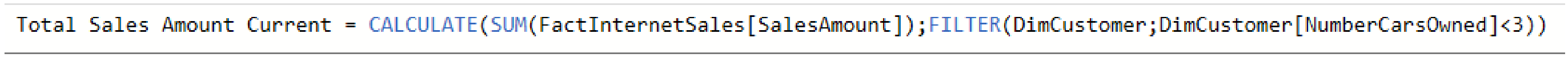
- A An expression always **starts with the name of the calculation**
- B **The equal sign** indicates the **beginning of the formula**
- C **A function or a combination of functions** is applied which will **return a value**
- D **The arguments of the function** (can be a reference to columns or additional functions)
- E **The table** that is being referenced
- F **The column** that is being referenced for the specified table

Calculations



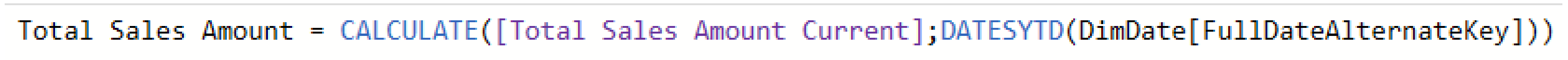
DAX Basics - Functions

- A function always **references a column or a table**. Filters can be added to filter context of evaluation
- A function **always returns a value or a table**. When a table is returned, further functions should be applied to obtain a value



```
Total Sales Amount Current = CALCULATE(SUM(FactInternetSales[SalesAmount]);FILTER(DimCustomer;DimCustomer[NumberCarsOwned]<3))
```

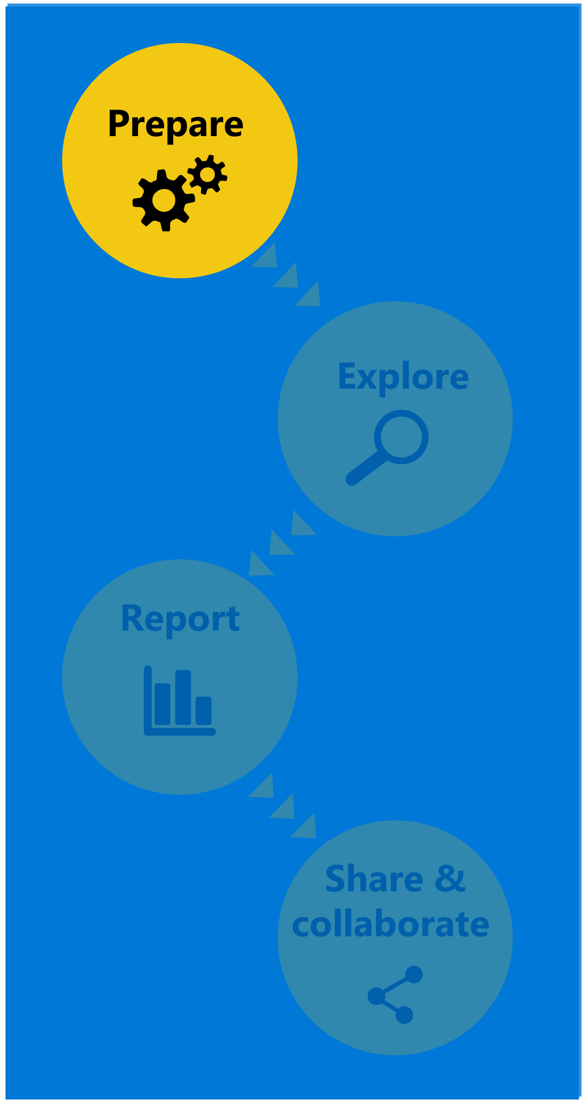
- Several **time intelligence** functions exist out of the box



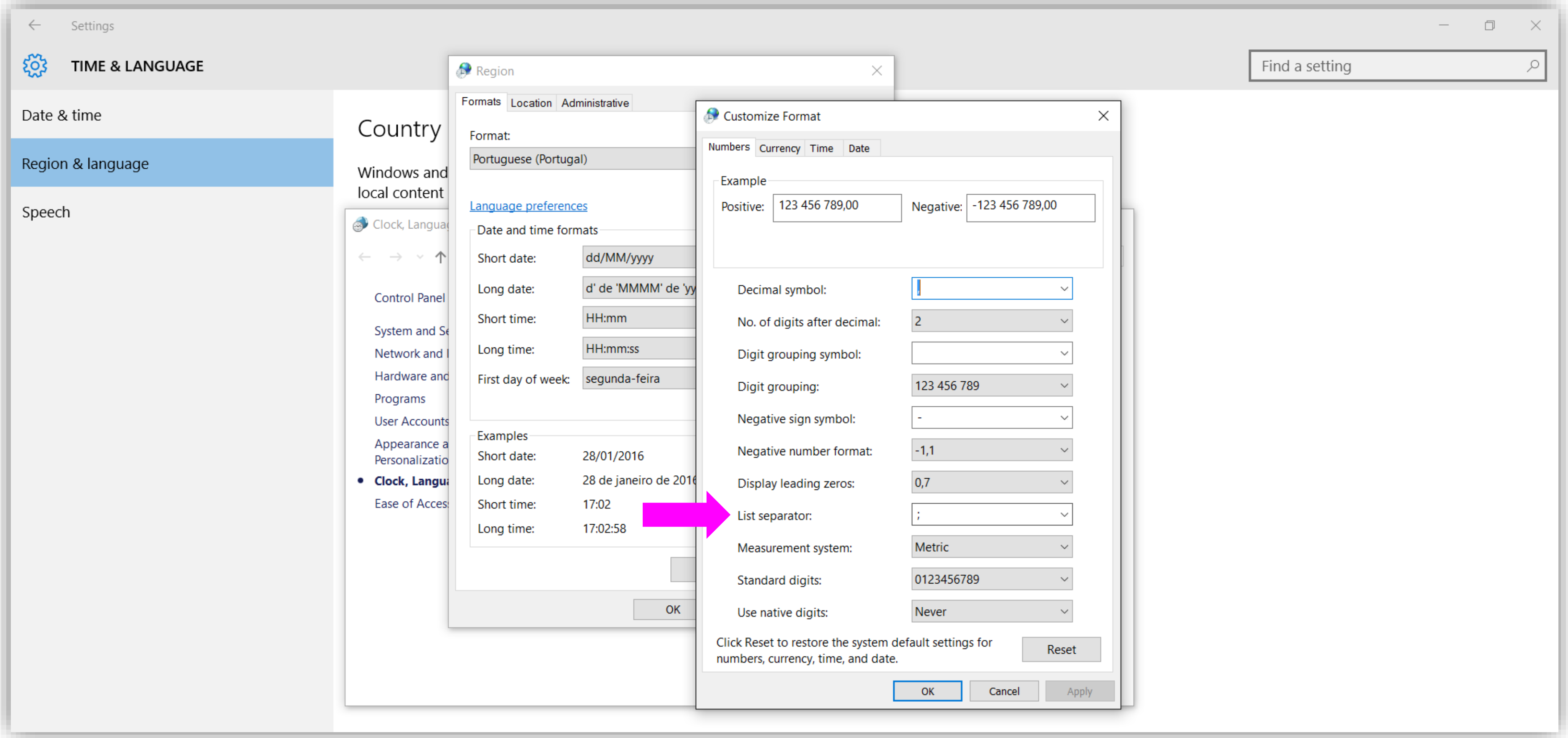
```
Total Sales Amount = CALCULATE([Total Sales Amount Current];DATESYTD(DimDate[FullDateAlternateKey]))
```

- Some **Excel functions** are also valid (like MONTH, FLOOR…)
- **Full reference** of available functions here:
  https://msdn.microsoft.com/en-us/library/ee634396.aspx

Calculations



DAX Basics – Semi-Colon and the List Separator



Format: Portuguese (Portugal)

Date and time formats

Short date:	dd/MM/yyyy
Long date:	d' de 'MMMM' de 'yy
Short time:	HH:mm
Long time:	HH:mm:ss
First day of week:	segunda-feira

Examples

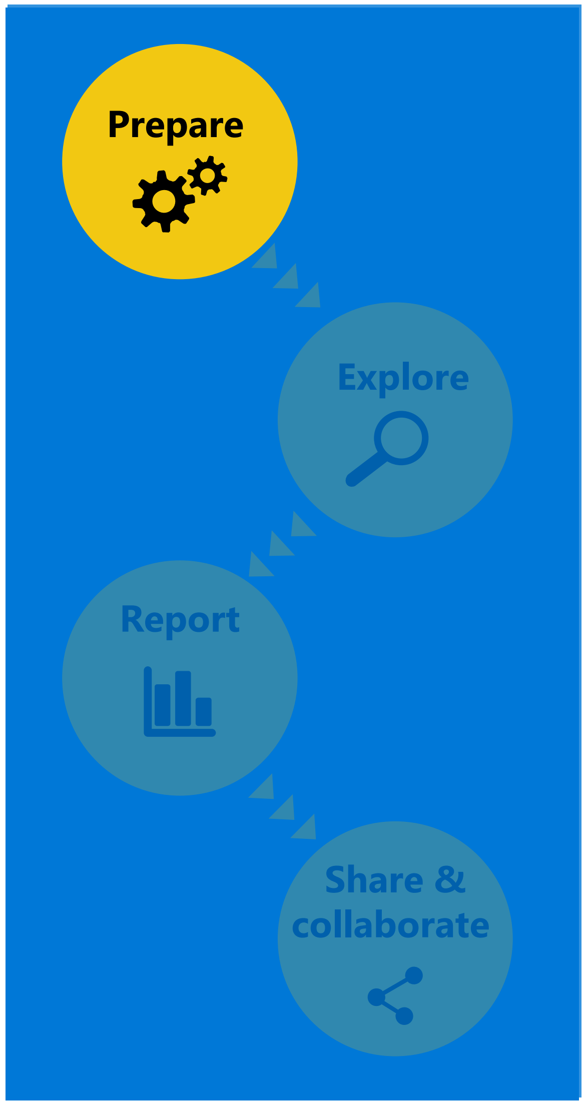
Short date:	28/01/2016
Long date:	28 de janeiro de 2016
Short time:	17:02
Long time:	17:02:58

Example
Positive: 123 456 789,00 Negative: -123 456 789,00

Decimal symbol:	,
No. of digits after decimal:	2
Digit grouping symbol:	
Digit grouping:	123 456 789
Negative sign symbol:	-
Negative number format:	-1,1
Display leading zeros:	0,7
List separator:	;
Measurement system:	Metric
Standard digits:	0123456789
Use native digits:	Never

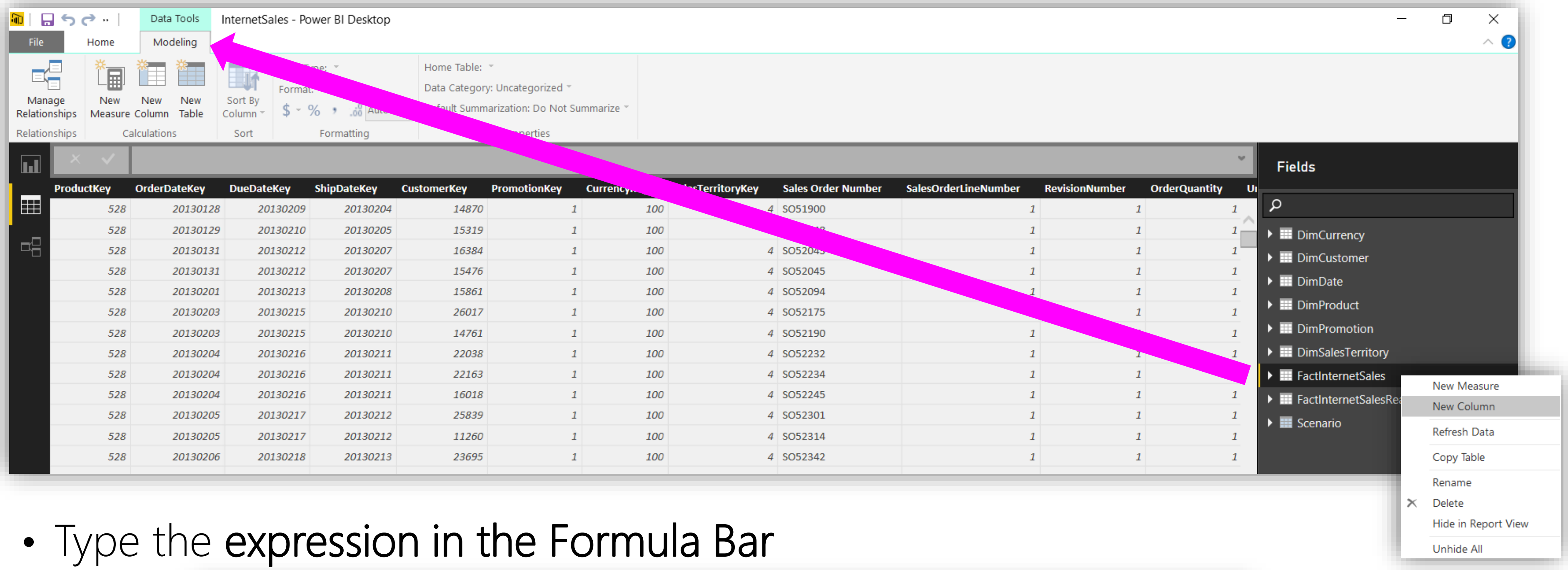
Click Reset to restore the system default settings for numbers, currency, time, and date.

Calculations



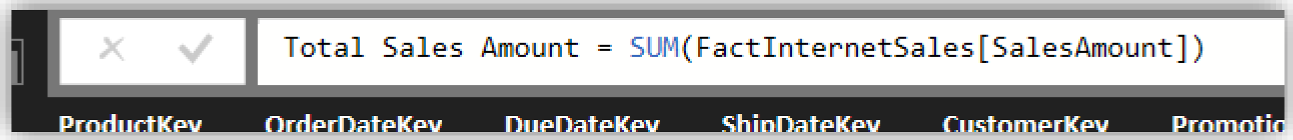
DAX Basics – Creating Calculations

- Select the **Data View** and then the **Modeling tab for the target table** (or with the table contextual menu)



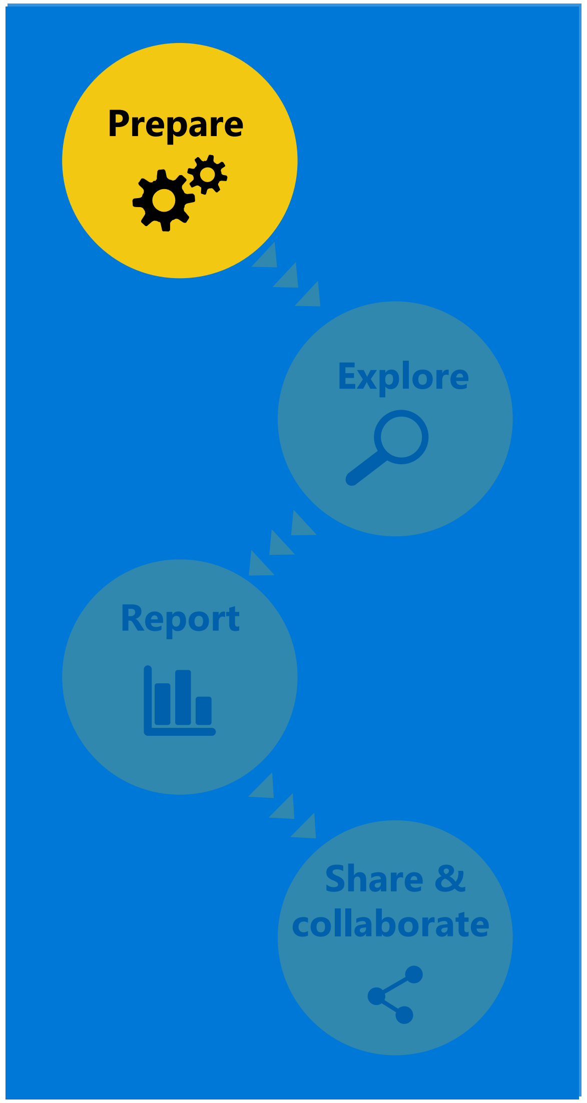
ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	Currency[illegible]	[illegible]esTerritoryKey	Sales Order Number	SalesOrderLineNumber	RevisionNumber	OrderQuantity
528	20130128	20130209	20130204	14870	1	100	4	SO51900	1	1	1
528	20130129	20130210	20130205	15319	1	100	[illegible]	[illegible]	1	1	1
528	20130131	20130212	20130207	16384	1	100	4	SO52043	1	1	1
528	20130131	20130212	20130207	15476	1	100	4	SO52045	1	1	1
528	20130201	20130213	20130208	15861	1	100	4	SO52094	1	1	1
528	20130203	20130215	20130210	26017	1	100	4	SO52175	[illegible]	1	1
528	20130203	20130215	20130210	14761	1	100	4	SO52190	1	[illegible]	1
528	20130204	20130216	20130211	22038	1	100	4	SO52232	1	1	1
528	20130204	20130216	20130211	22163	1	100	4	SO52234	1	1	[illegible]
528	20130204	20130216	20130211	16018	1	100	4	SO52245	1	1	1
528	20130205	20130217	20130212	25839	1	100	4	SO52301	1	1	1
528	20130205	20130217	20130212	11260	1	100	4	SO52314	1	1	1
528	20130206	20130218	20130213	23695	1	100	4	SO52342	1	1	1

- Type the **expression in the Formula Bar**



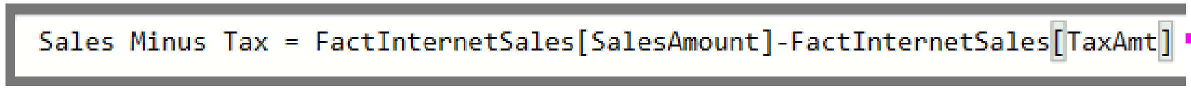
```
Total Sales Amount = SUM(FactInternetSales[SalesAmount])
```

Calculations



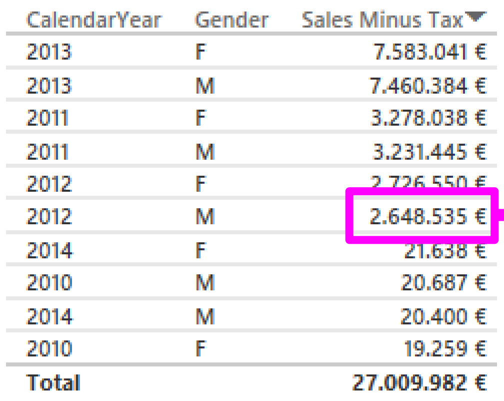
DAX Basics – How are calculations applied?

- **Row context** can be thought of as the current row - like an iterator
- **Filter context** determines the **conditions that filter data**, for a particular visualization, and can be applied on top of Row Context
- An example:



```
Sales Minus Tax = FactInternetSales[SalesAmount]-FactInternetSales[TaxAmt]
```

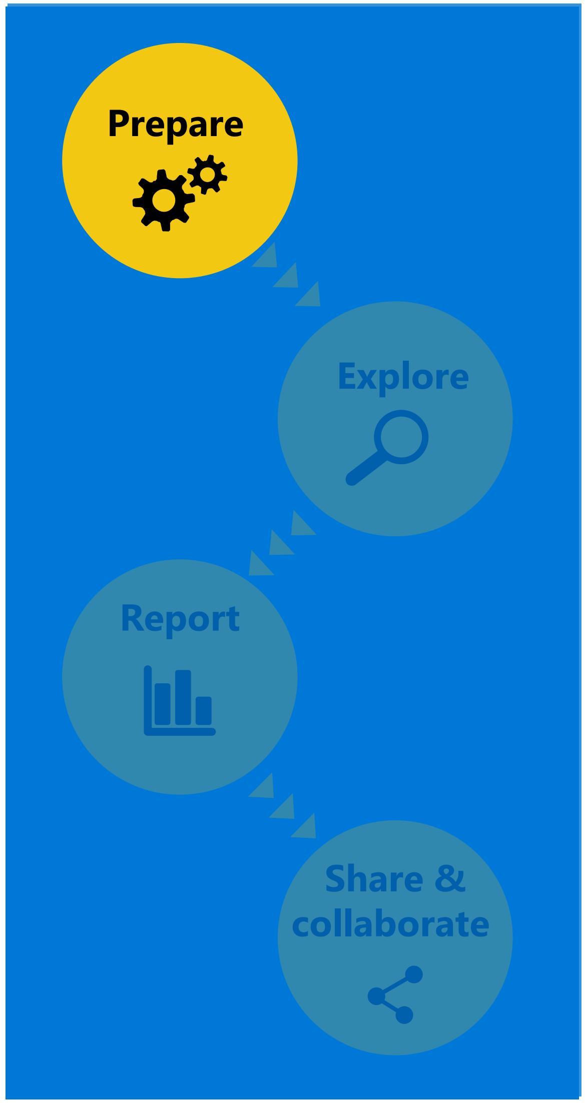
Row Context



CalendarYear	Gender	Sales Minus Tax
2013	F	7.583.041 €
2013	M	7.460.384 €
2011	F	3.278.038 €
2011	M	3.231.445 €
2012	F	2.726.550 €
2012	M	2.648.535 €
2014	F	21.638 €
2010	M	20.687 €
2014	M	20.400 €
2010	F	19.259 €
Total		**27.009.982 €**

This cell corresponds to the sum of "Sales Minus Tax" for all rows that belong to the Calendar Year "2012" and Gender is "M"

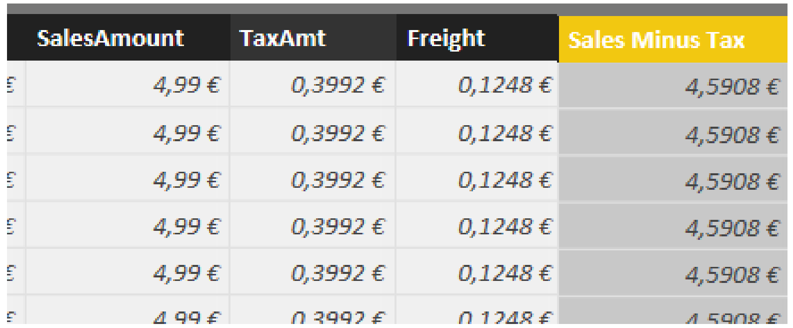
Calculations



Calculated Columns

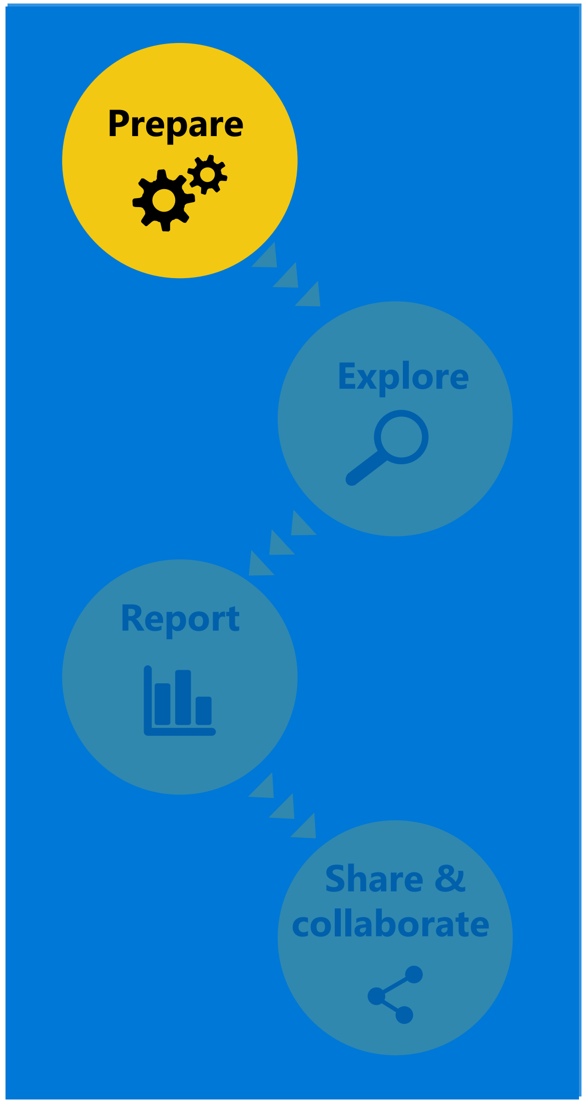
- They are **persisted** in the model and are **calculated row-by-row** (Row Context)
- They **increase the size and memory** requirements for the model
- They **can be re-used** in other calculations
- **DAX Functions can be used** in their definition (beware of aggregations)

```
Sales Minus Tax = FactInternetSales[SalesAmount]-FactInternetSales[TaxAmt]
```



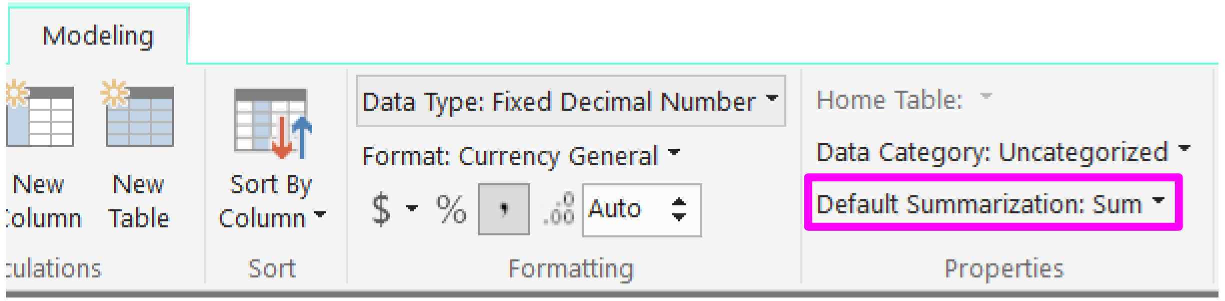
SalesAmount	TaxAmt	Freight	Sales Minus Tax
4,99 €	0,3992 €	0,1248 €	4,5908 €
4,99 €	0,3992 €	0,1248 €	4,5908 €
4,99 €	0,3992 €	0,1248 €	4,5908 €
4,99 €	0,3992 €	0,1248 €	4,5908 €
4,99 €	0,3992 €	0,1248 €	4,5908 €
4,99 €	0,3992 €	0,1248 €	4,5908 €

Calculations

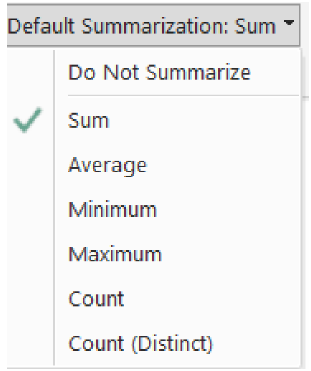


Calculated Columns

- A **calculation** or a **numeric field** has a "**Default Summarization**" – defines how it is aggregated when placed on a visualization, by default



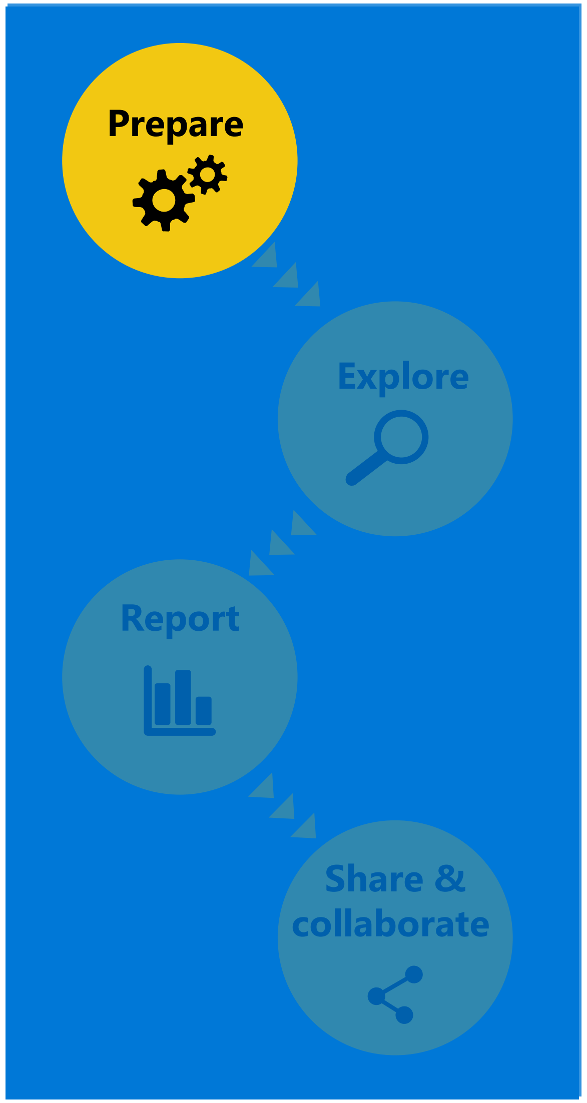
Data Type: Fixed Decimal Number
Format: Currency General
Home Table:
Data Category: Uncategorized
Default Summarization: Sum



- A Calculated Column with a "Default Summarization" of "**Do not summarize**" will appear with ScenarioKey

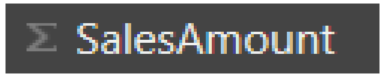
- With a **different summarization** will appear with Σ Sales Minus Tax

Calculations

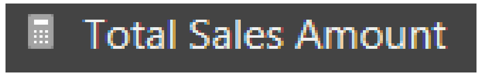


Measures

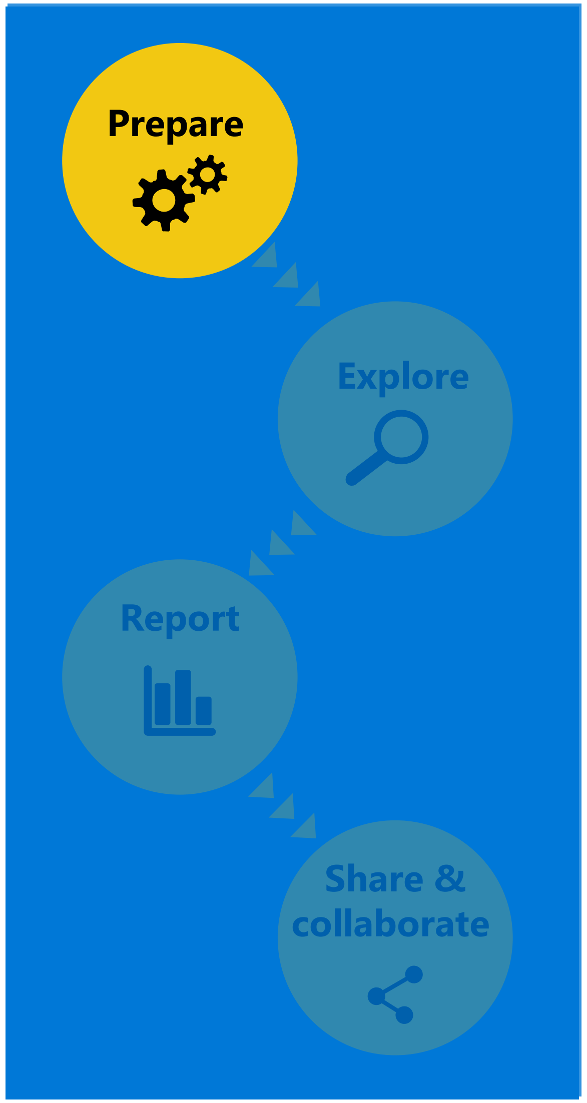
- Measures are **calculated when they are used** in a particular visualization
- And they **can also be calculated on a row-by-row** basis
- They can be **implicit** – an aggregation of a field with "Default Summarization" different from "Do Not Summarize"


Σ SalesAmount

- They can be **explicit**, where they are the result of a DAX expression. You cannot control the aggregation for this type.

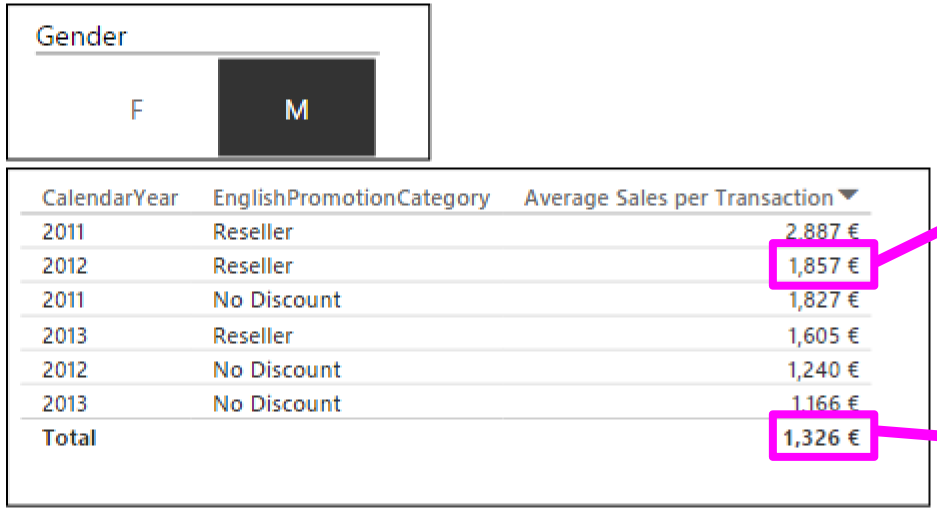

Total Sales Amount

Calculations



Measures

- They can (and should) be **referenced from other measures** – this is a best practice
- Measures are **evaluated for each cell** they appear in (Filter Context)

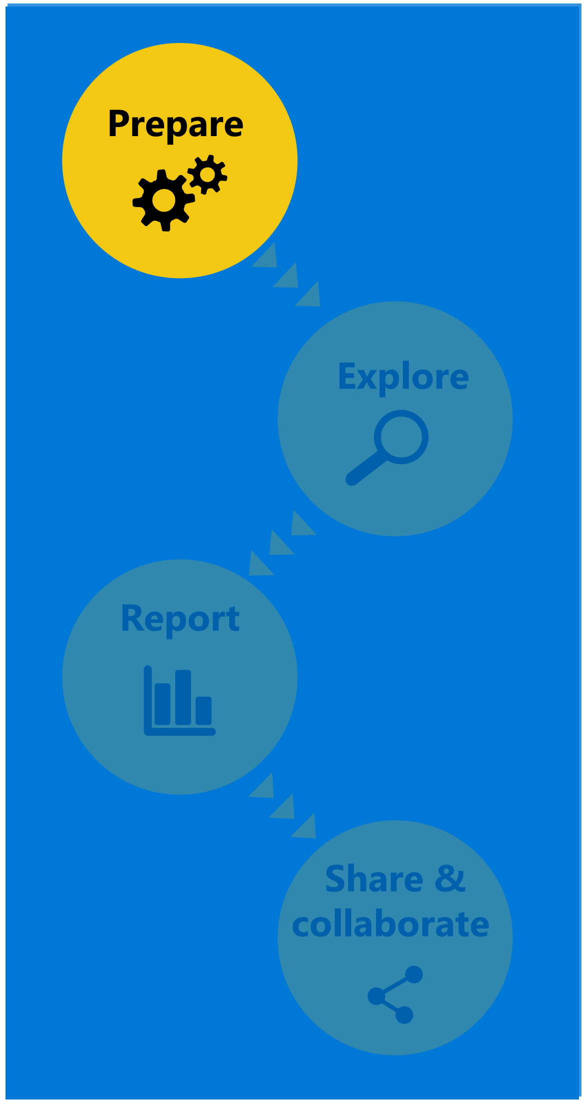


CalendarYear	EnglishPromotionCategory	Average Sales per Transaction
2011	Reseller	2,887 €
2012	Reseller	1,857 €
2011	No Discount	1,827 €
2013	Reseller	1,605 €
2012	No Discount	1,240 €
2013	No Discount	1,166 €
Total		**1,326 €**

CalendarYear=2012
Promotion="Reseller"
Gender="M"

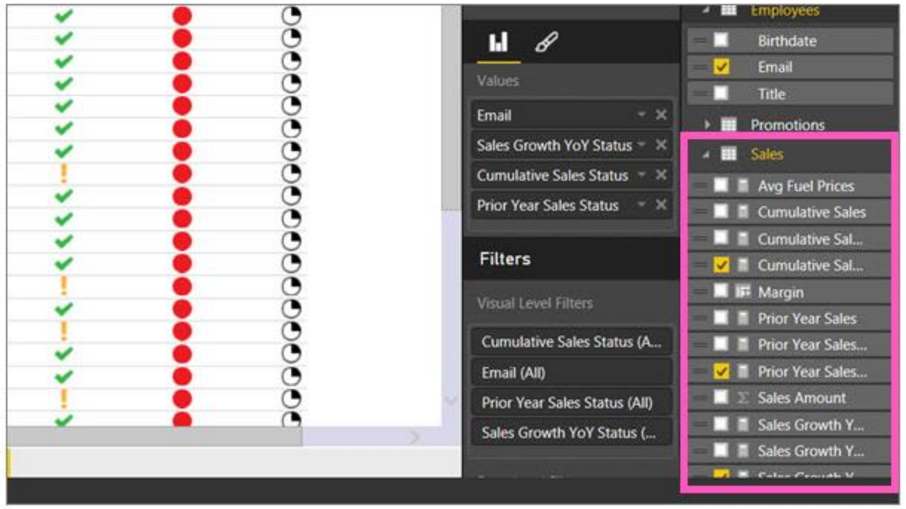
The total is also calculated independently. It is not an aggregation of the rows.

Calculations



Key Performance Indicators

- KPIs don’t exist yet as first class objects, but can **come from SSAS or Power Pivot**



Values
- Email
- Sales Growth YoY Status
- Cumulative Sales Status
- Prior Year Sales Status

Filters

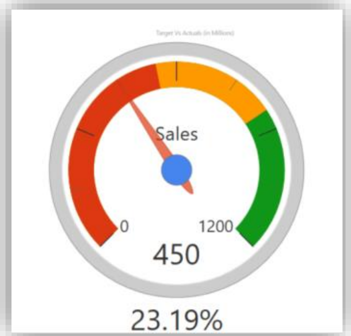
Visual Level Filters
- Cumulative Sales Status (A...
- Email (All)
- Prior Year Sales Status (All)
- Sales Growth YoY Status (...

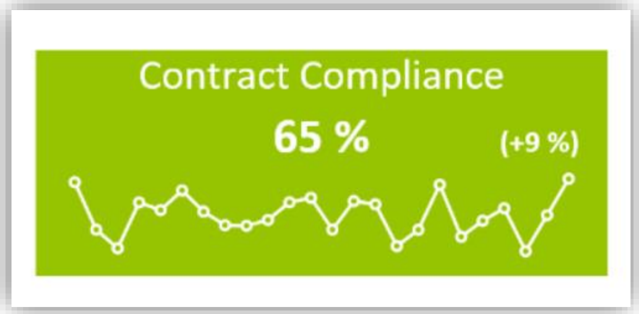
Employees
- [ ] Birthdate
- [✓] Email
- [ ] Title

Promotions

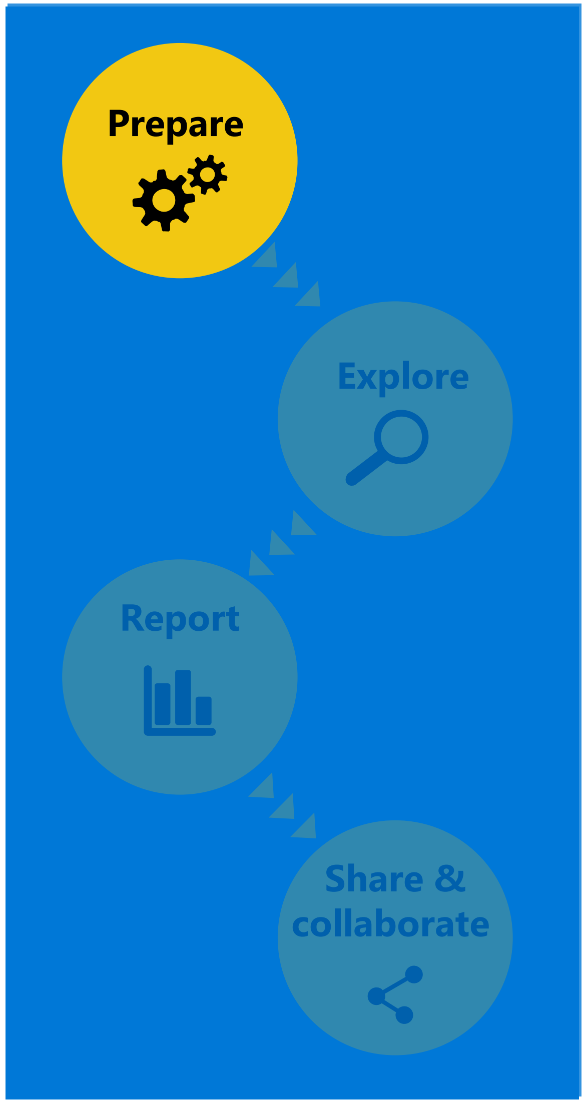
Sales
- [ ] Avg Fuel Prices
- [ ] Cumulative Sales
- [ ] Cumulative Sal...
- [✓] Cumulative Sal...
- [ ] Margin
- [ ] Prior Year Sales
- [ ] Prior Year Sales...
- [✓] Prior Year Sales...
- [ ] Sales Amount
- [ ] Sales Growth Y...
- [ ] Sales Growth Y...

- **Unstructured alternatives**



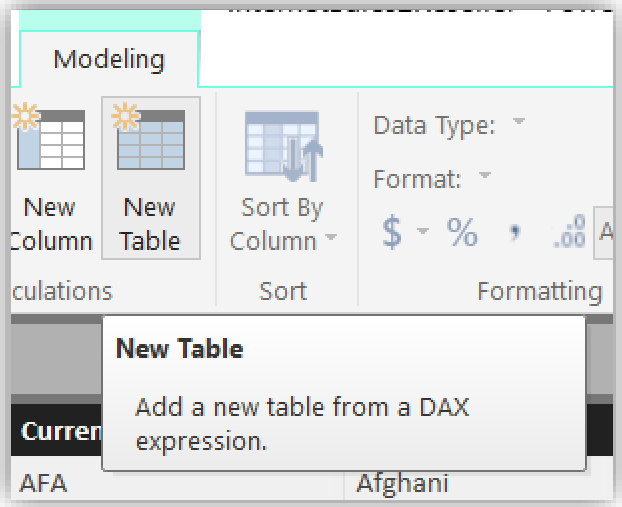


Calculations



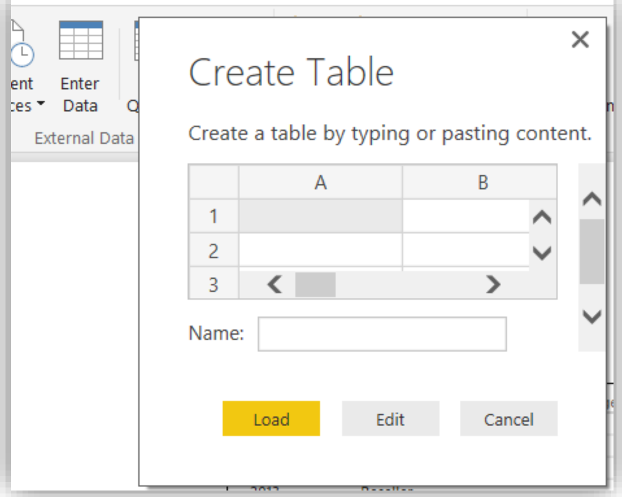
Calculated Tables

- Become new tables which are the **result of a DAX Expression**
- They behave **like regular tables** and **created via Modeling Tab**
- New tables **can also be created in the Query Editor** with Formula Language **or by typing or pasting content**



New Table

Add a new table from a DAX expression.

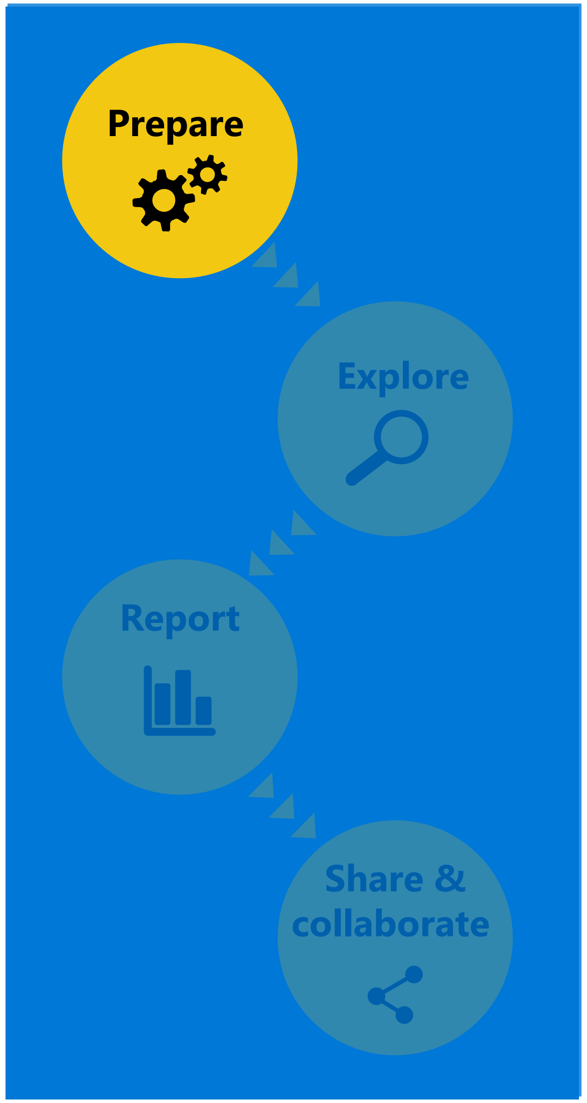


Create Table

Create a table by typing or pasting content.

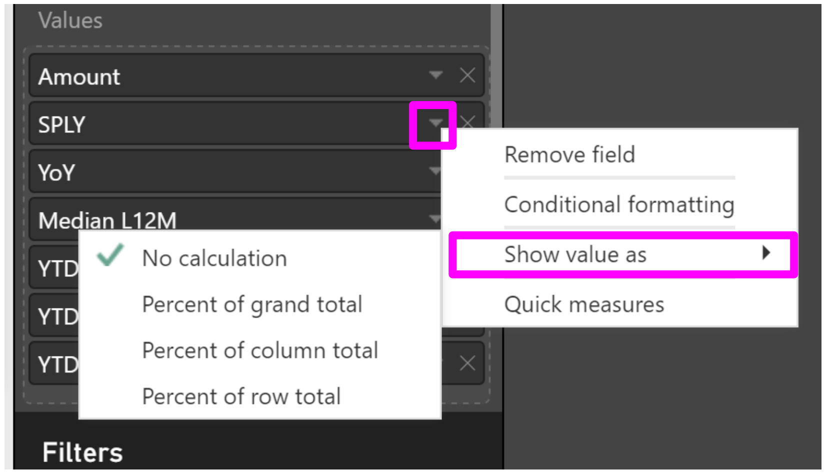
Name:

Calculations

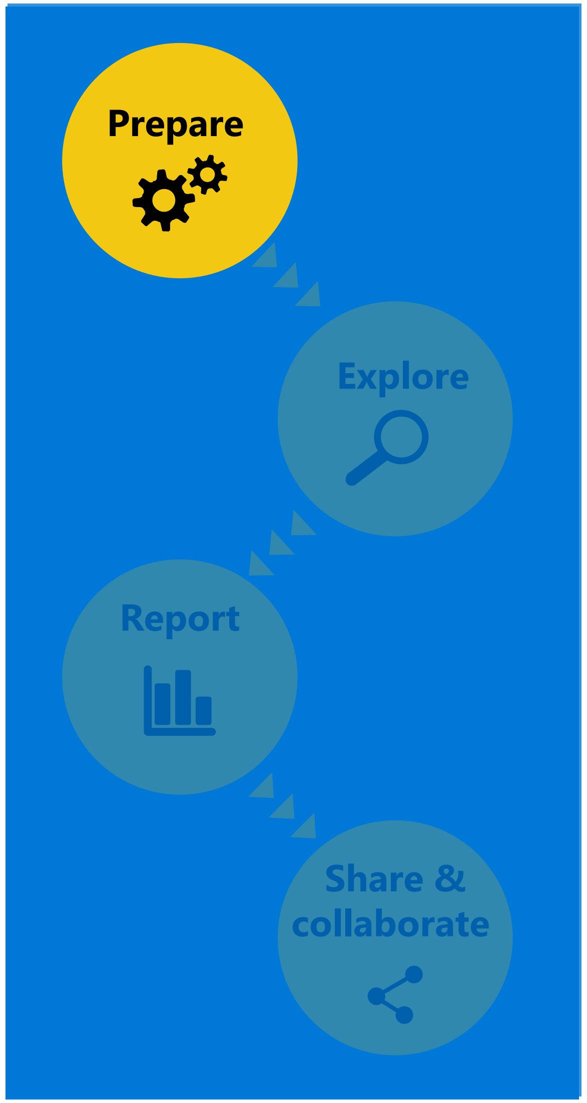


Show value as

- **Out-of-the-box calculations** that can be applied without coding
- Accessible **via field well**, you can change how a numerical value or measure is displayed



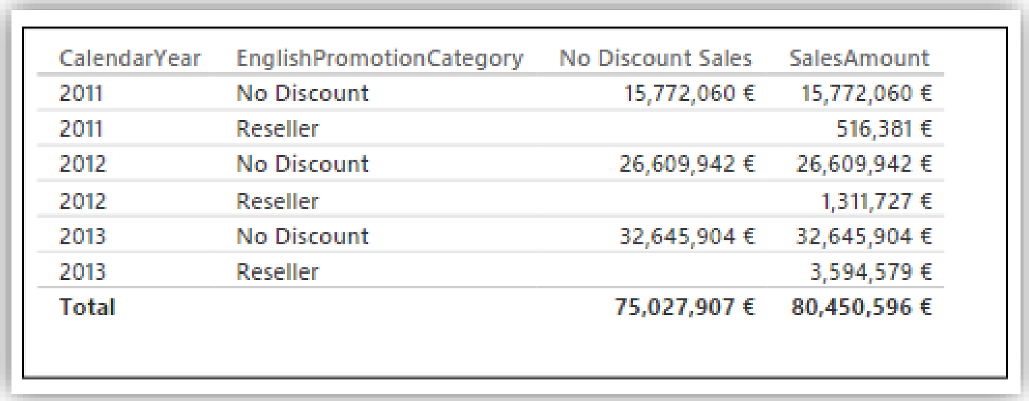
Calculations



Advanced Calculations

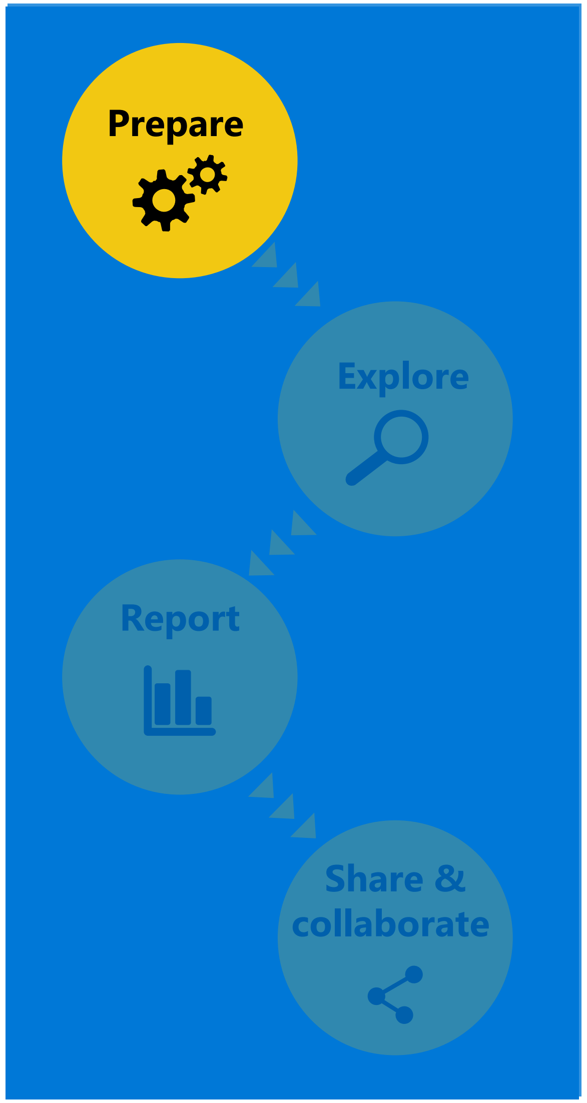
- CALCULATE(<measure expression>, <filter1>, <filter2>, …) allows us to change the filter context

```
No Discount Sales =
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    DimPromotion[EnglishPromotionName] = "No Discount"
)
```



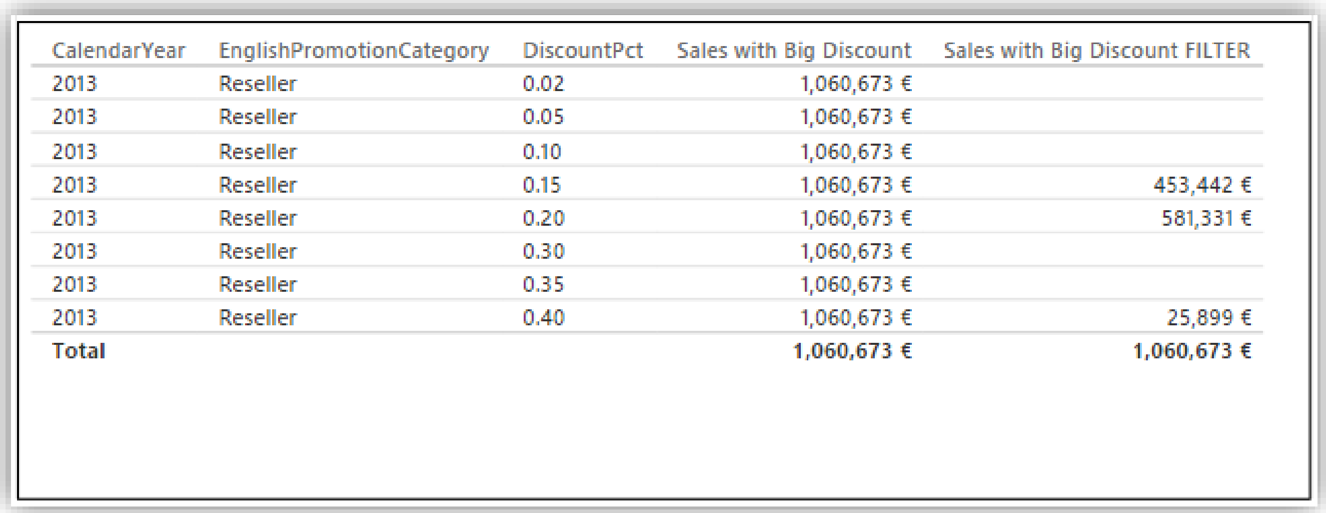
CalendarYear	EnglishPromotionCategory	No Discount Sales	SalesAmount
2011	No Discount	15,772,060 €	15,772,060 €
2011	Reseller		516,381 €
2012	No Discount	26,609,942 €	26,609,942 €
2012	Reseller		1,311,727 €
2013	No Discount	32,645,904 €	32,645,904 €
2013	Reseller		3,594,579 €
Total		**75,027,907 €**	**80,450,596 €**

Calculations



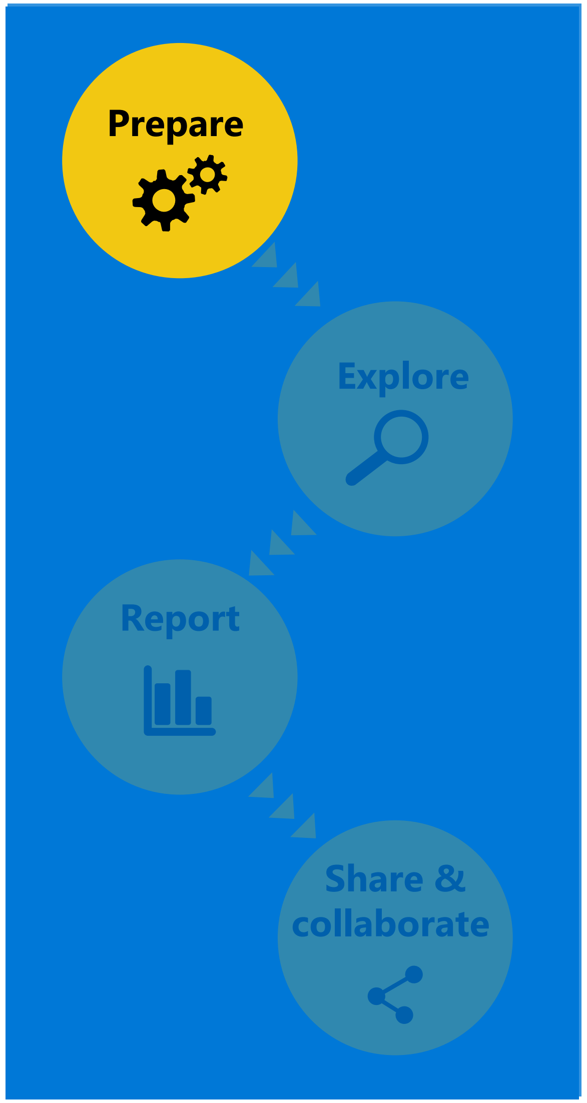
Advanced Calculations

- **FILTER** allows more complex filtering but is less performant than **CALCULATE**
- Should be **used to filter on smaller tables** like Dimensions
- It **follows the relationships** in order to filter, so its results are more user-friendly



CalendarYear	EnglishPromotionCategory	DiscountPct	Sales with Big Discount	Sales with Big Discount FILTER
2013	Reseller	0.02	1,060,673 €	
2013	Reseller	0.05	1,060,673 €	
2013	Reseller	0.10	1,060,673 €	
2013	Reseller	0.15	1,060,673 €	453,442 €
2013	Reseller	0.20	1,060,673 €	581,331 €
2013	Reseller	0.30	1,060,673 €	
2013	Reseller	0.35	1,060,673 €	
2013	Reseller	0.40	1,060,673 €	25,899 €
Total			**1,060,673 €**	**1,060,673 €**

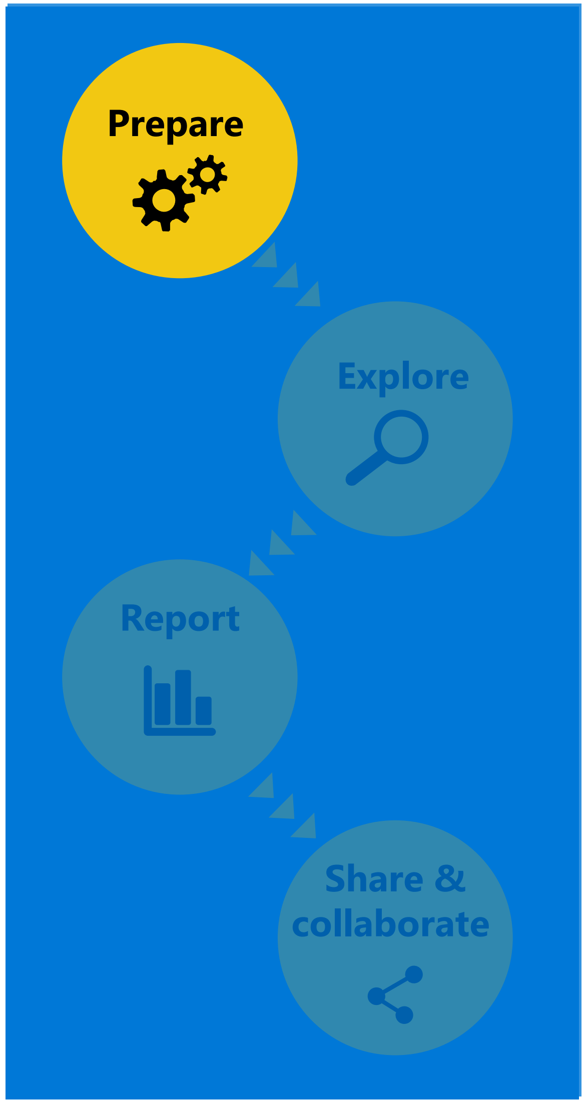
Calculations



Advanced Calculations

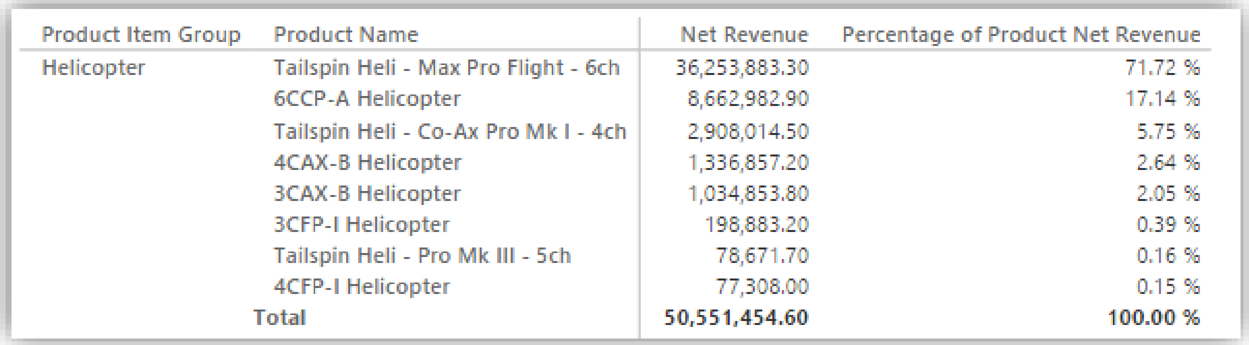
```
Sales with Big Discount =
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    DimPromotion[DiscountPct] > 0,1
)

Sales with Big Discount FILTER =
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    FILTER ( DimPromotion; DimPromotion[DiscountPct] > 0,1 )
)
```

Calculations



Advanced Calculations

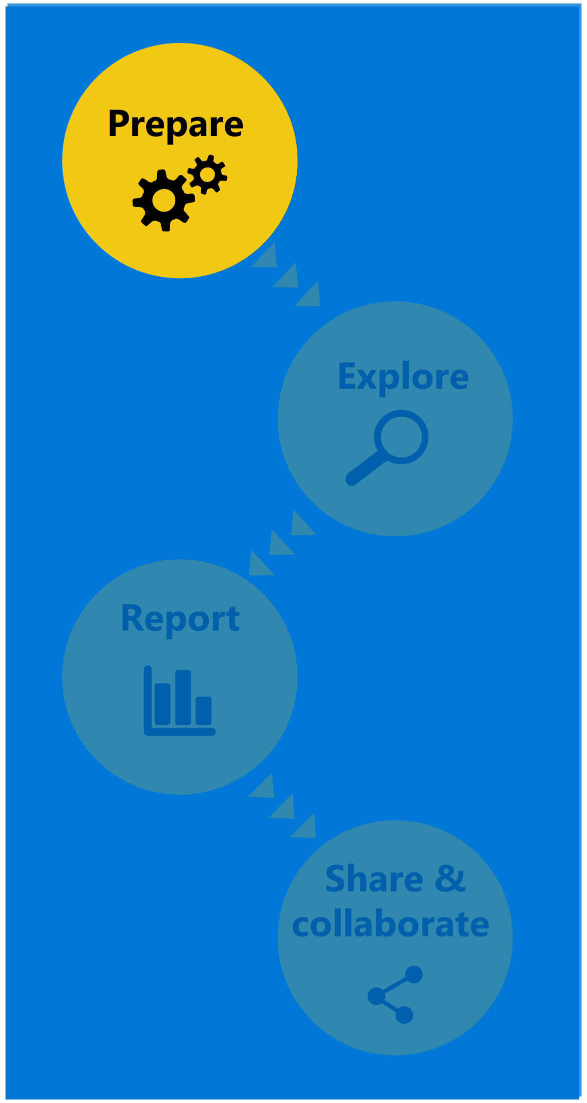
- **ALL removes the filter** applied to a table or column.
- Useful for **ratio-to-parent calculations**



Product Item Group	Product Name	Net Revenue	Percentage of Product Net Revenue
Helicopter	Tailspin Heli - Max Pro Flight - 6ch	36,253,883.30	71.72 %
	6CCP-A Helicopter	8,662,982.90	17.14 %
	Tailspin Heli - Co-Ax Pro Mk I - 4ch	2,908,014.50	5.75 %
	4CAX-B Helicopter	1,336,857.20	2.64 %
	3CAX-B Helicopter	1,034,853.80	2.05 %
	3CFP-I Helicopter	198,883.20	0.39 %
	Tailspin Heli - Pro Mk III - 5ch	78,671.70	0.16 %
	4CFP-I Helicopter	77,308.00	0.15 %
	Total	**50,551,454.60**	**100.00 %**

```
Percentage of Product Net Revenue =
DIVIDE (
    SUM (Sales[Net Revenue]);
    CALCULATE (
        SUM (Sales[Net Revenue]);
        ALL ( Product[Product Name] )
    )
)
```

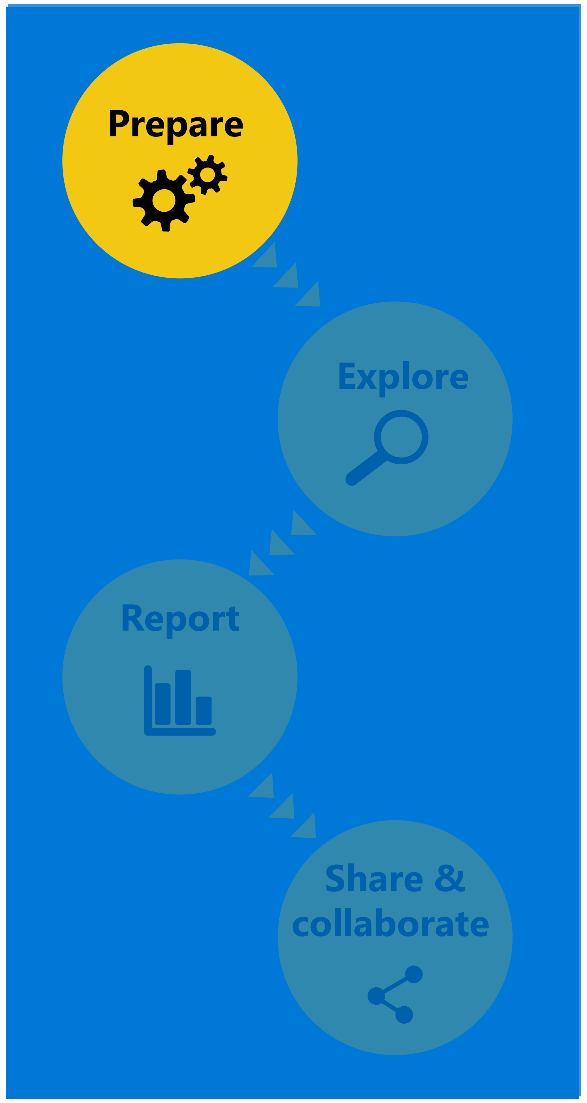
Calculations



Advanced Calculations

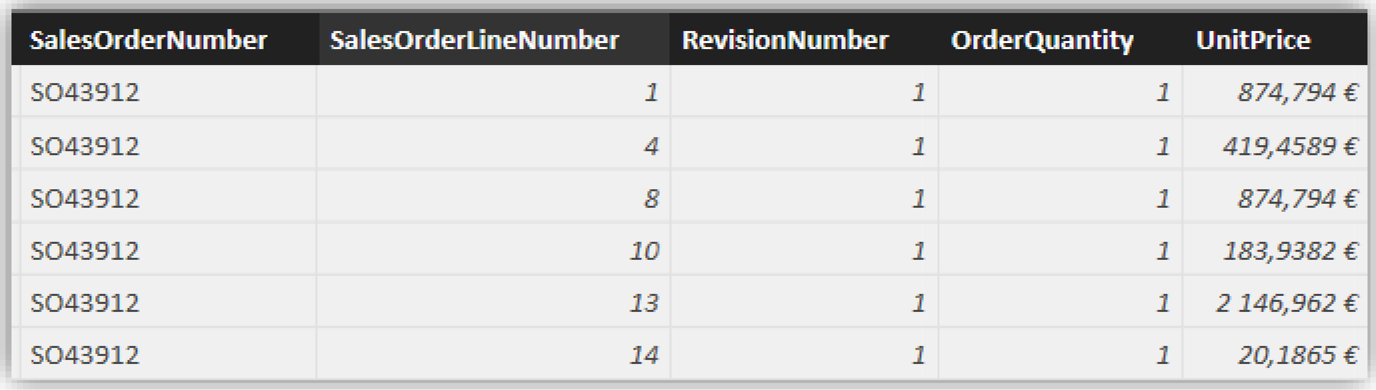
- X Functions (SUMX, MAXX, MINX, AVERAGEX, COUNTX, COUNTAX)
- Parameters (<table or table expression>,<arithmetic expression>)
- It will iterate each row on the first table, and apply the arithmetic expression.
- Useful for row-by-row calculations, correcting TOTALS and *hidden calculations*
- Might, potentially, be slower than SUM

Calculations



Advanced Calculations

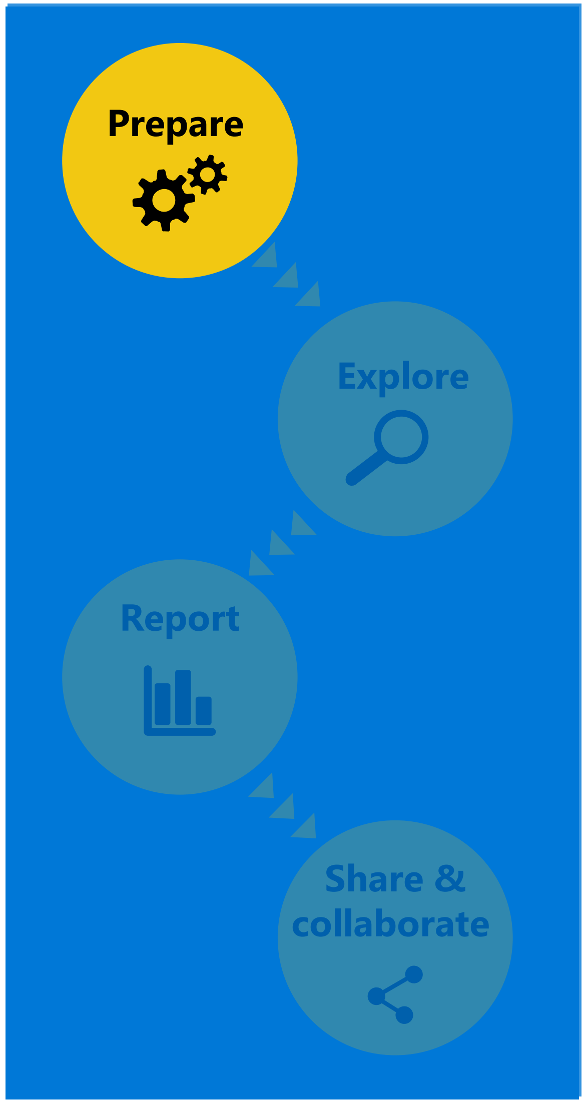
- X Functions - Row-by-row



SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	OrderQuantity	UnitPrice
SO43912	1	1	1	874,794 €
SO43912	4	1	1	419,4589 €
SO43912	8	1	1	874,794 €
SO43912	10	1	1	183,9382 €
SO43912	13	1	1	2 146,962 €
SO43912	14	1	1	20,1865 €

```
SalesAmountwithSUMX =
SUMX (
    FactResellerSales;
    FactResellerSales[OrderQuantity] * FactResellerSales[UnitPrice]
)
```

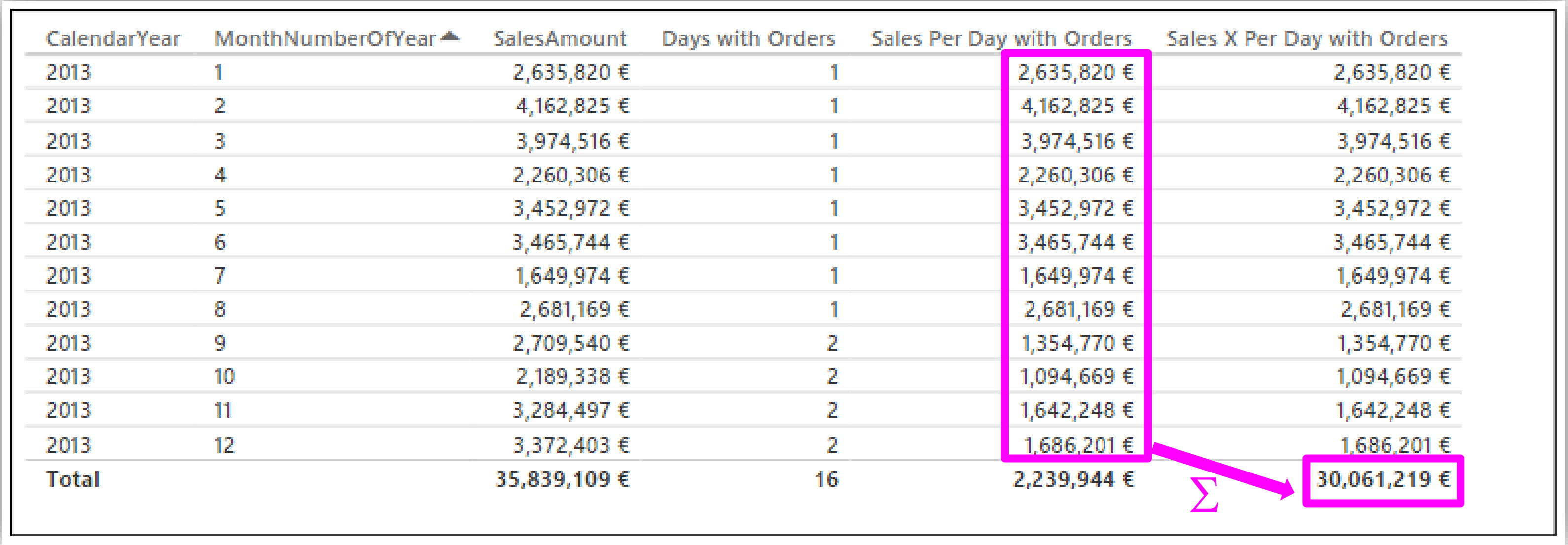
Calculations



Advanced Calculations

- X Functions - Corrected Totals

Sales Per Day with Orders =
DIVIDE (SUM (FactResellerSales[SalesAmount]); [Days with Orders])

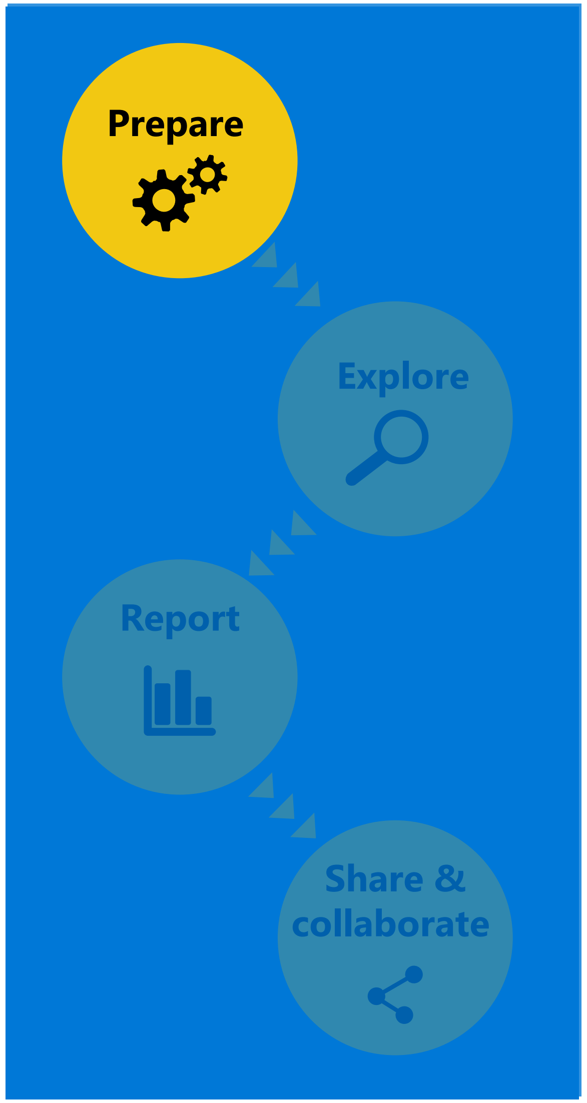


CalendarYear	MonthNumberOfYear	SalesAmount	Days with Orders	Sales Per Day with Orders	Sales X Per Day with Orders
2013	1	2,635,820 €	1	2,635,820 €	2,635,820 €
2013	2	4,162,825 €	1	4,162,825 €	4,162,825 €
2013	3	3,974,516 €	1	3,974,516 €	3,974,516 €
2013	4	2,260,306 €	1	2,260,306 €	2,260,306 €
2013	5	3,452,972 €	1	3,452,972 €	3,452,972 €
2013	6	3,465,744 €	1	3,465,744 €	3,465,744 €
2013	7	1,649,974 €	1	1,649,974 €	1,649,974 €
2013	8	2,681,169 €	1	2,681,169 €	2,681,169 €
2013	9	2,709,540 €	2	1,354,770 €	1,354,770 €
2013	10	2,189,338 €	2	1,094,669 €	1,094,669 €
2013	11	3,284,497 €	2	1,642,248 €	1,642,248 €
2013	12	3,372,403 €	2	1,686,201 €	1,686,201 €
Total		**35,839,109 €**	**16**	**2,239,944 €**	**30,061,219 €**

Σ

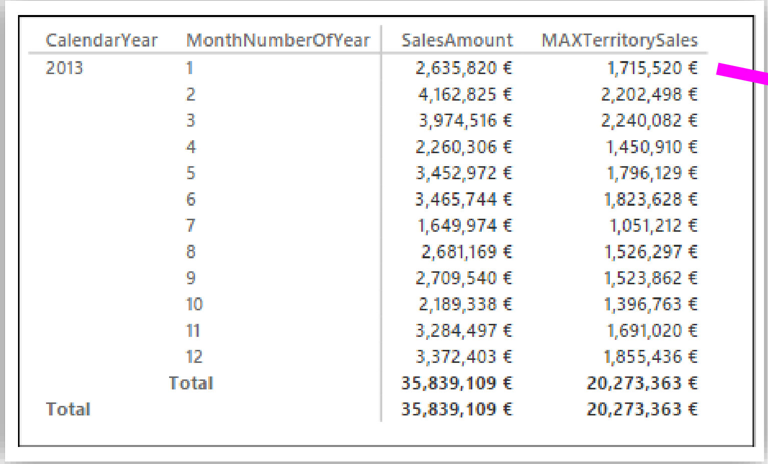
Sales X Per Day with Orders =
SUMX (VALUES (DimDate[MonthNumberOfYear]); [Sales Per Day with Orders])

Calculations

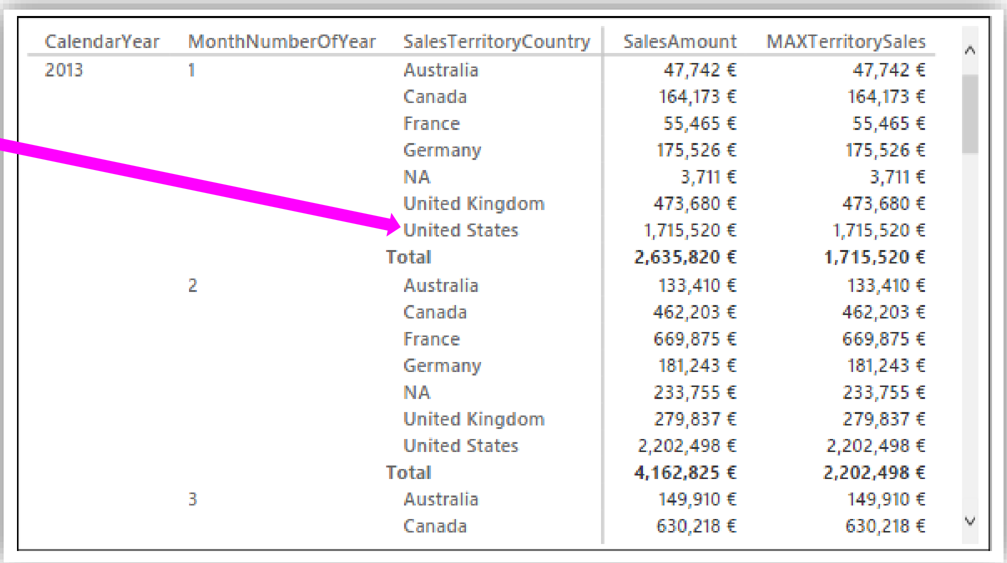


Advanced Calculations

- X Functions - Hidden Calculations



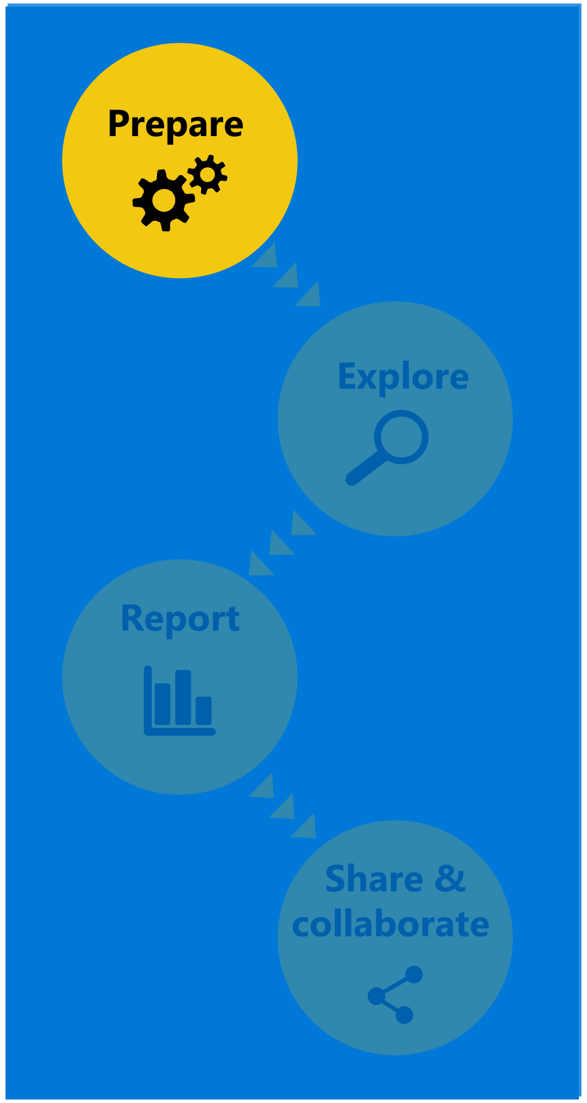
CalendarYear	MonthNumberOfYear	SalesAmount	MAXTerritorySales
2013	1	2,635,820 €	1,715,520 €
	2	4,162,825 €	2,202,498 €
	3	3,974,516 €	2,240,082 €
	4	2,260,306 €	1,450,910 €
	5	3,452,972 €	1,796,129 €
	6	3,465,744 €	1,823,628 €
	7	1,649,974 €	1,051,212 €
	8	2,681,169 €	1,526,297 €
	9	2,709,540 €	1,523,862 €
	10	2,189,338 €	1,396,763 €
	11	3,284,497 €	1,691,020 €
	12	3,372,403 €	1,855,436 €
	Total	**35,839,109 €**	**20,273,363 €**
Total		**35,839,109 €**	**20,273,363 €**



CalendarYear	MonthNumberOfYear	SalesTerritoryCountry	SalesAmount	MAXTerritorySales
2013	1	Australia	47,742 €	47,742 €
		Canada	164,173 €	164,173 €
		France	55,465 €	55,465 €
		Germany	175,526 €	175,526 €
		NA	3,711 €	3,711 €
		United Kingdom	473,680 €	473,680 €
		United States	1,715,520 €	1,715,520 €
		Total	**2,635,820 €**	**1,715,520 €**
	2	Australia	133,410 €	133,410 €
		Canada	462,203 €	462,203 €
		France	669,875 €	669,875 €
		Germany	181,243 €	181,243 €
		NA	233,755 €	233,755 €
		United Kingdom	279,837 €	279,837 €
		United States	2,202,498 €	2,202,498 €
		Total	**4,162,825 €**	**2,202,498 €**
	3	Australia	149,910 €	149,910 €
		Canada	630,218 €	630,218 €

```
MAXTerritorySales =
MAXX (
    VALUES ( DimSalesTerritory[SalesTerritoryCountry] );
    FactResellerSales[Sales]
)
```

Calculations



Advanced Calculations

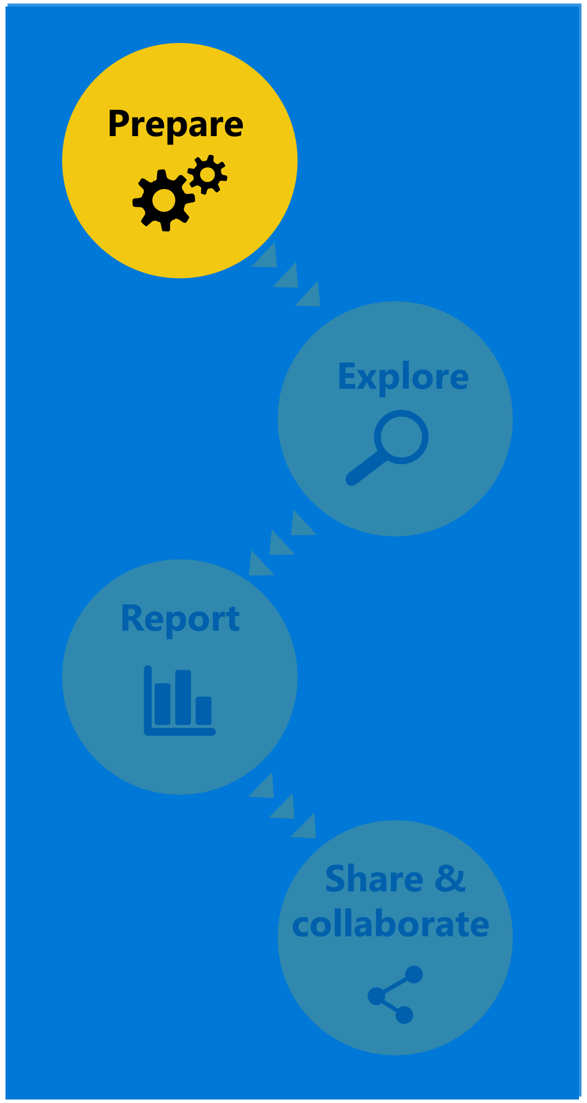
- Time Intelligence
- Many DAX functions exist to support time calculations. There are two types.
- Functions that **require a CALCULATE**

```
SalesYTD =
CALCULATE ( [Sales]; DATESYTD ( DimDate[FullDateAlternateKey] ) )
```

- Functions that **return a scalar** (*syntactic sugar*)

```
SalesYTDNoCalculate =
TOTALYTD ( [Sales]; DimDate[FullDateAlternateKey] )
```

Calculations



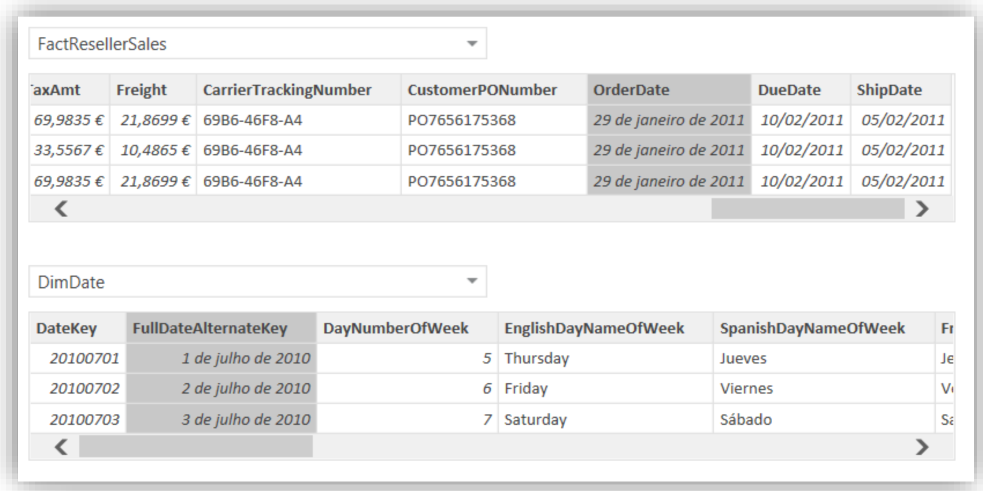
Advanced Calculations

- Time Intelligence

- **To enable** time intelligence **with your date tables, two conditions** are necessary:

  - The **relationship** between the fact table and the date table must be done through a **date field on both sides;**

  - The **calculation must then target the date column** on the date dimension which must be **contiguous**

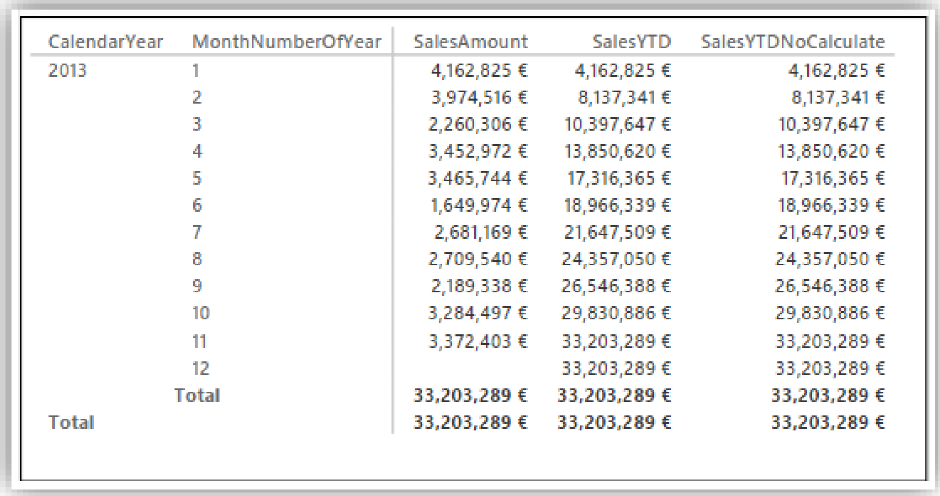


FactResellerSales

ʼaxAmt	Freight	CarrierTrackingNumber	CustomerPONumber	OrderDate	DueDate	ShipDate
69,9835 €	21,8699 €	69B6-46F8-A4	PO7656175368	29 de janeiro de 2011	10/02/2011	05/02/2011
33,5567 €	10,4865 €	69B6-46F8-A4	PO7656175368	29 de janeiro de 2011	10/02/2011	05/02/2011
69,9835 €	21,8699 €	69B6-46F8-A4	PO7656175368	29 de janeiro de 2011	10/02/2011	05/02/2011

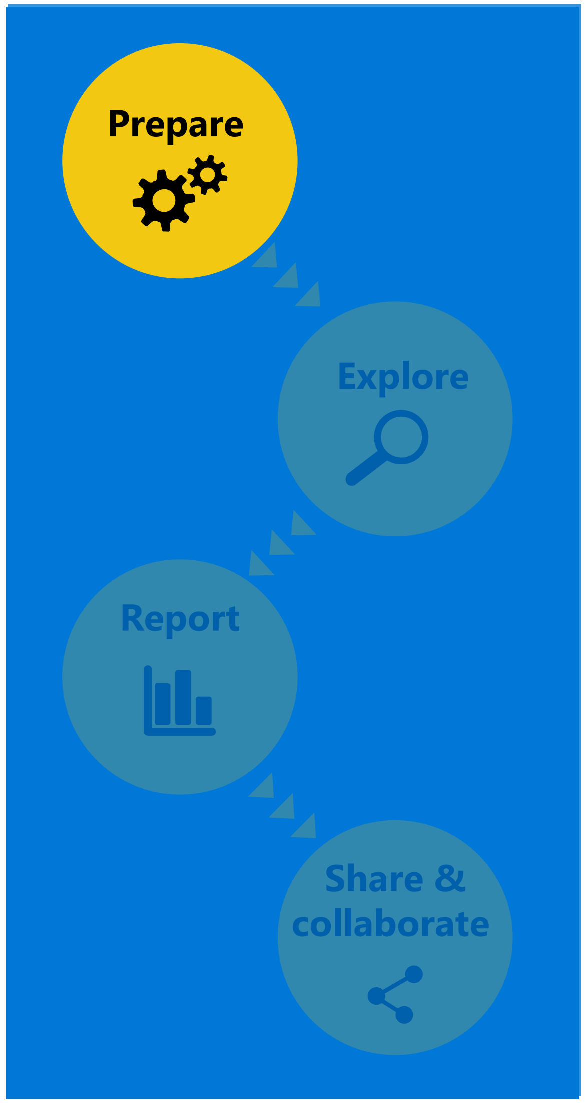
DimDate

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	Fr
20100701	1 de julho de 2010	5	Thursday	Jueves	Je
20100702	2 de julho de 2010	6	Friday	Viernes	V
20100703	3 de julho de 2010	7	Saturday	Sábado	Sa



CalendarYear	MonthNumberOfYear	SalesAmount	SalesYTD	SalesYTDNoCalculate
2013	1	4,162,825 €	4,162,825 €	4,162,825 €
	2	3,974,516 €	8,137,341 €	8,137,341 €
	3	2,260,306 €	10,397,647 €	10,397,647 €
	4	3,452,972 €	13,850,620 €	13,850,620 €
	5	3,465,744 €	17,316,365 €	17,316,365 €
	6	1,649,974 €	18,966,339 €	18,966,339 €
	7	2,681,169 €	21,647,509 €	21,647,509 €
	8	2,709,540 €	24,357,050 €	24,357,050 €
	9	2,189,338 €	26,546,388 €	26,546,388 €
	10	3,284,497 €	29,830,886 €	29,830,886 €
	11	3,372,403 €	33,203,289 €	33,203,289 €
	12		33,203,289 €	33,203,289 €
	Total	**33,203,289 €**	**33,203,289 €**	**33,203,289 €**
Total		**33,203,289 €**	**33,203,289 €**	**33,203,289 €**

Calculations

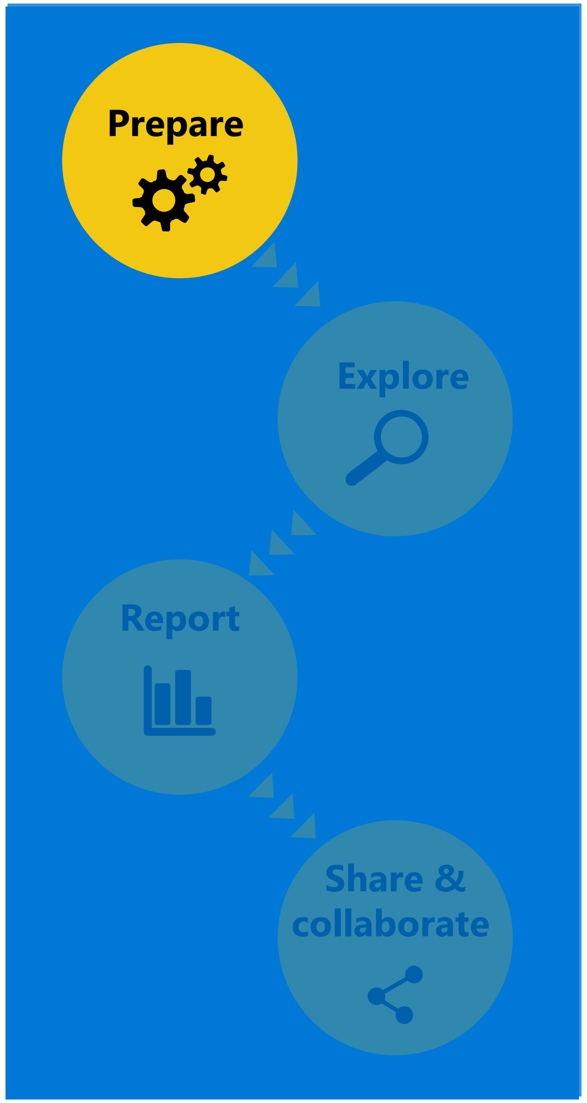


Advanced Calculations

- Time Intelligence with Auto Date/Time
- A **hierarchy is automatically generated** for each date field on each table
- Calculations need to be done with the “in-line” notation

```
SalesAmountYTD =
CALCULATE (
    SUM ( FactInternetSales[SalesAmount] );
    DATESYTD ( FactInternetSales[OrderDate].[Date] )
)
```

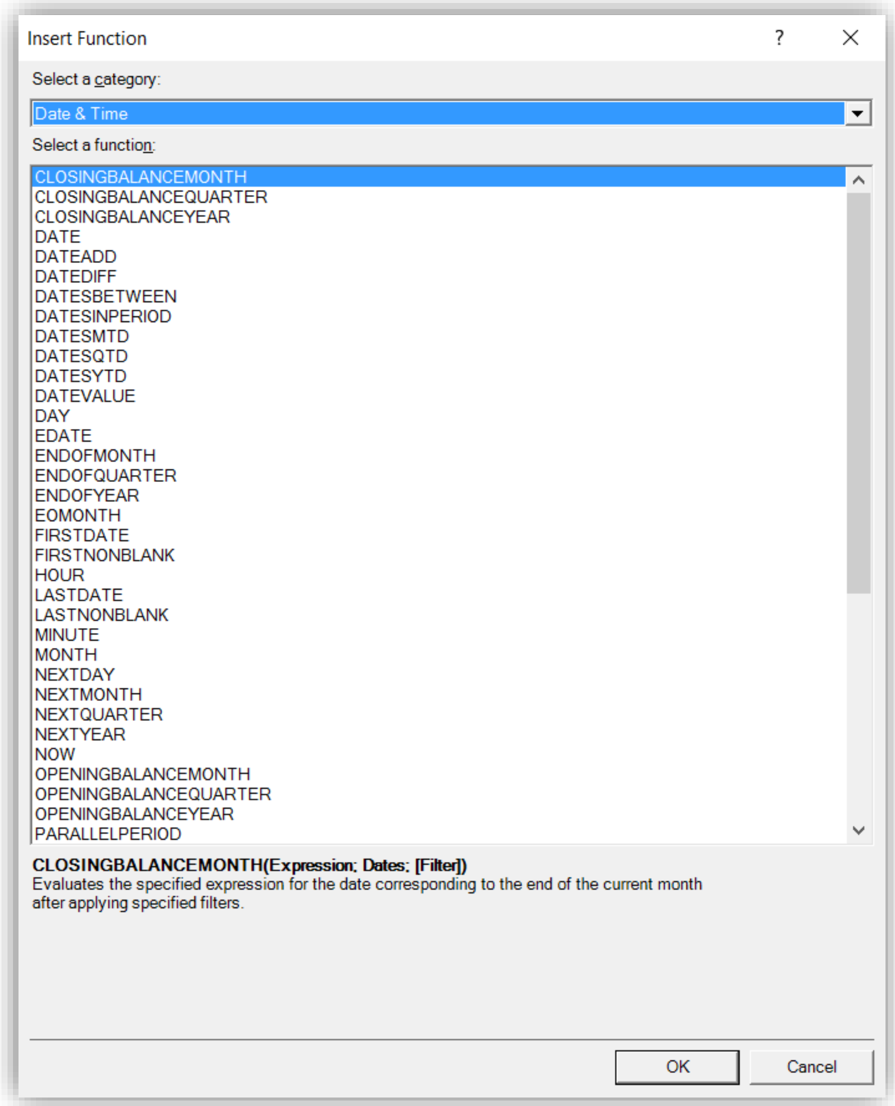
Calculations



Advanced Calculations

- Time Intelligence
- Function Types:
  - Semi-additive
  - Dates Until
  - Periods Between
  - Time Navigation

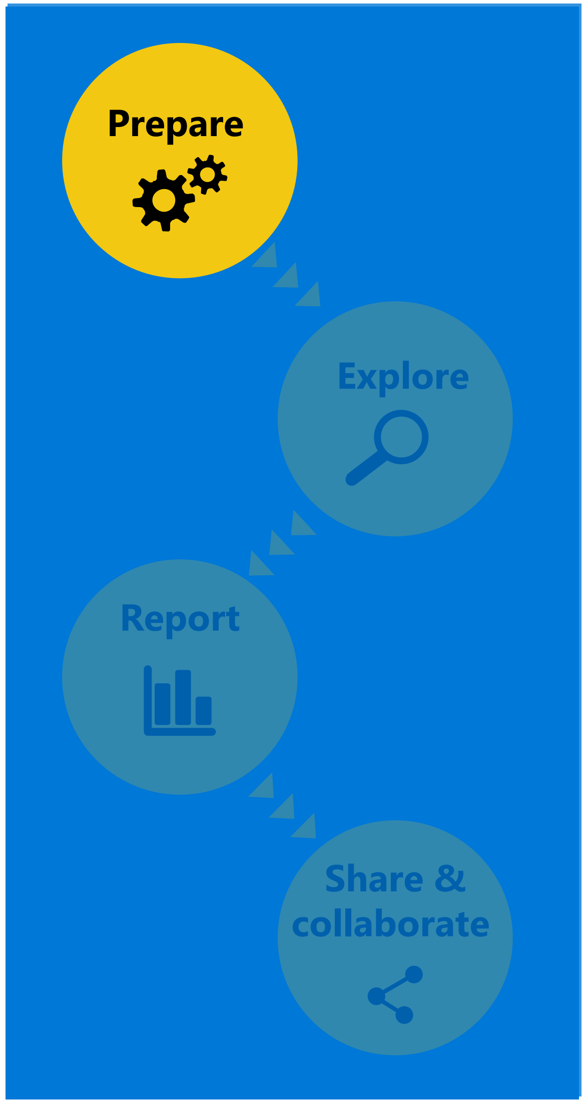
*Excel screenshot



CLOSINGBALANCEMONTH
CLOSINGBALANCEQUARTER
CLOSINGBALANCEYEAR
DATE
DATEADD
DATEDIFF
DATESBETWEEN
DATESINPERIOD
DATESMTD
DATESQTD
DATESYTD
DATEVALUE
DAY
EDATE
ENDOFMONTH
ENDOFQUARTER
ENDOFYEAR
EOMONTH
FIRSTDATE
FIRSTNONBLANK
HOUR
LASTDATE
LASTNONBLANK
MINUTE
MONTH
NEXTDAY
NEXTMONTH
NEXTQUARTER
NEXTYEAR
NOW
OPENINGBALANCEMONTH
OPENINGBALANCEQUARTER
OPENINGBALANCEYEAR
PARALLELPERIOD

CLOSINGBALANCEMONTH(Expression; Dates; [Filter])
Evaluates the specified expression for the date corresponding to the end of the current month after applying specified filters.

Calculations

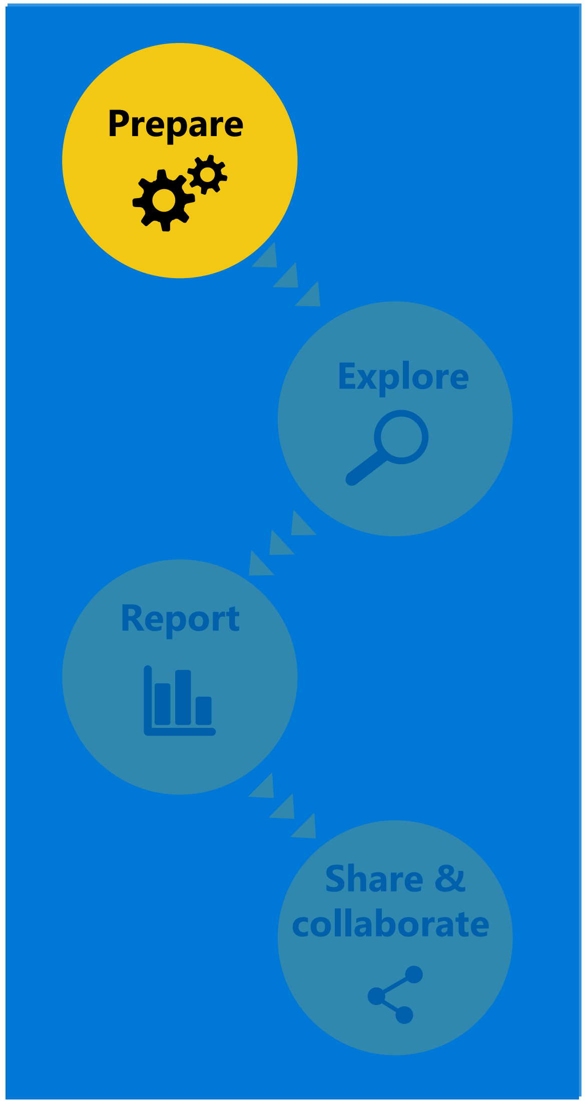


Advanced Calculations

- Time Intelligence

```
SalesLastMonth =
CALCULATE ( [Sales];
DATEADD ( DimDate[FullDateAlternateKey]; -1; MONTH ) )

SalesSamePeriodLastYear =
CALCULATE ( [Sales];
SAMEPERIODLASTYEAR ( DimDate[FullDateAlternateKey] ) )

MonthOverMonth =
DIVIDE ( ( [Sales] - [SalesLastMonth] ); [SalesLastMonth] )
```

Calculations

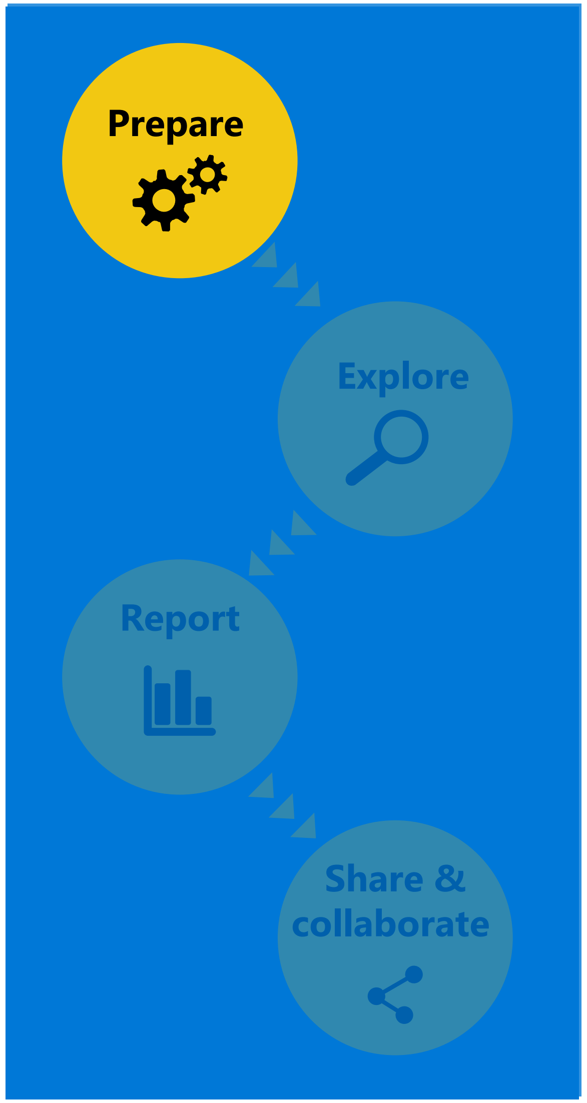


Advanced Calculations

- Time Intelligence

```
SalesSinceEver =
CALCULATE (
    [Sales];
    DATESBETWEEN (
        DimDate[FullDateAlternateKey];
        FIRSTDATE ( ALL ( DimDate[FullDateAlternateKey] ) );
        LASTDATE ( DimDate[FullDateAlternateKey] )
    )
)
```

Calculations

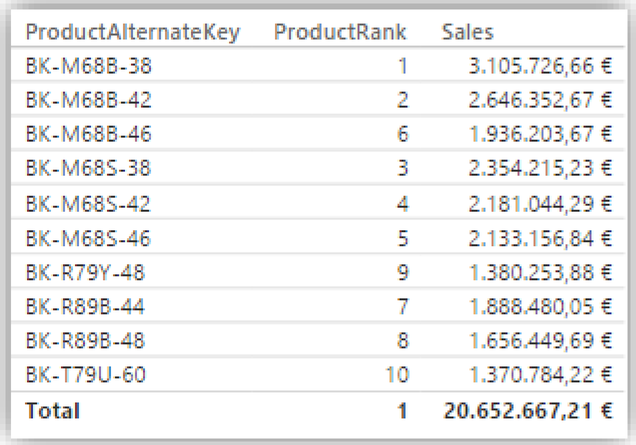


Advanced Calculations:

- RANKX(<table expression>;<arithmetic expression>;<sort order>;<tie handler>)

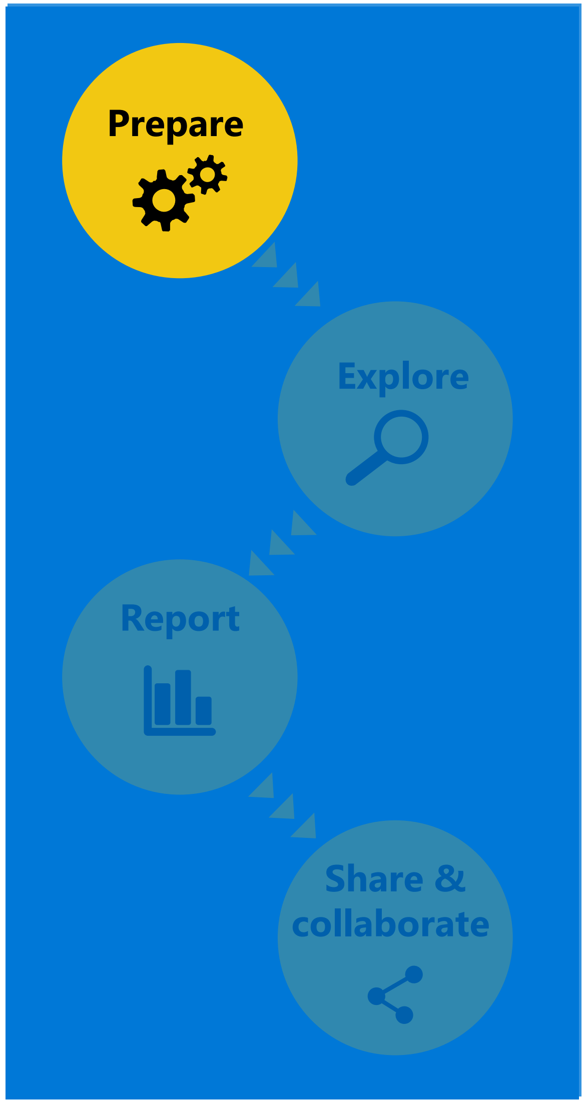
- Gets an ordinal position for the selected column

```
ProductRank =
RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP )
```



ProductAlternateKey	ProductRank	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68B-46	6	1.936.203,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-R79Y-48	9	1.380.253,88 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-T79U-60	10	1.370.784,22 €
Total	**1**	**20.652.667,21 €**

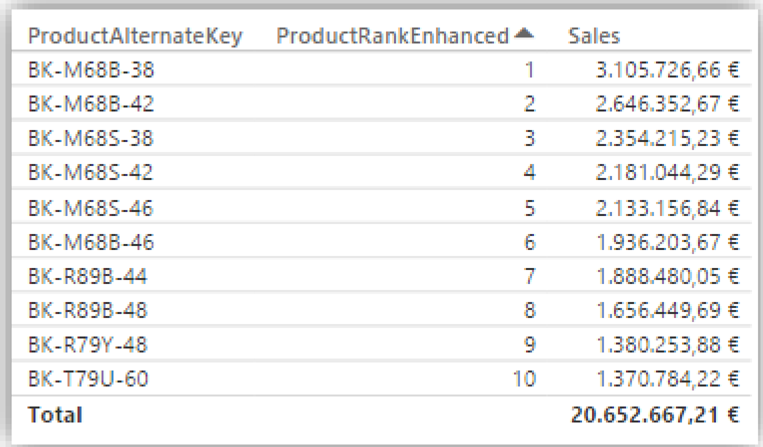
Calculations



Advanced Calculations

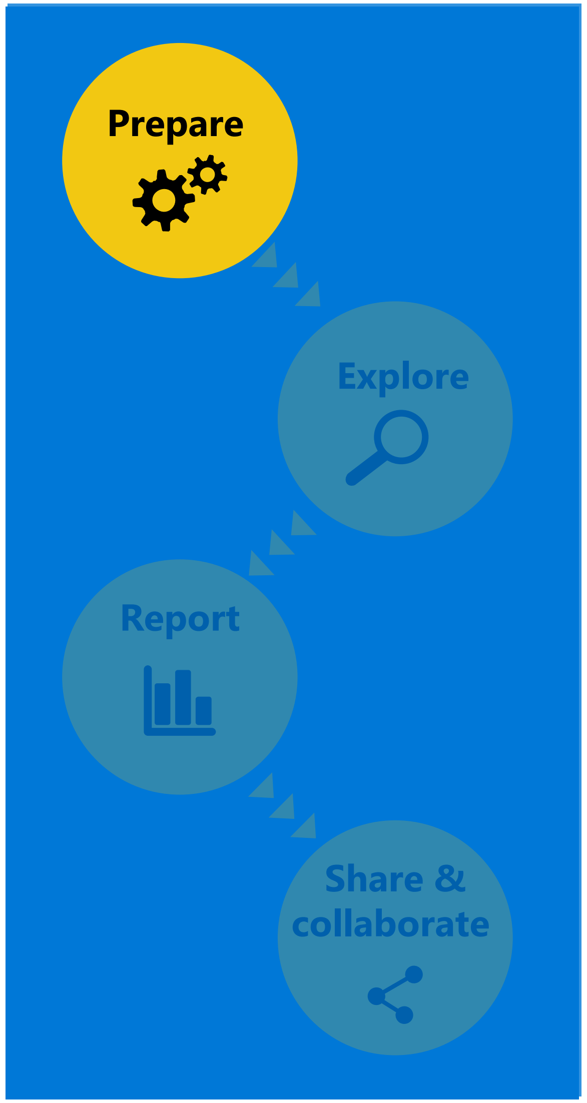
- HASONEVALUE lets us test if only a single element is selected for a column

```
ProductRankEnhanced =
IF (
    HASONEVALUE ( DimProduct[ProductAlternateKey] );
    RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP );
    BLANK ()
)
```



ProductAlternateKey	ProductRankEnhanced	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-M68B-46	6	1.936.203,67 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-R79Y-48	9	1.380.253,88 €
BK-T79U-60	10	1.370.784,22 €
Total		**20.652.667,21 €**

Calculations

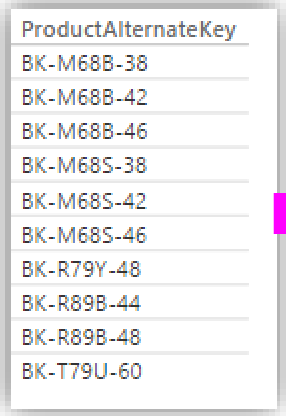


Advanced Calculations

- **TOPN**(<n>;<table>;<order by expression>) returns the top N rows for a table

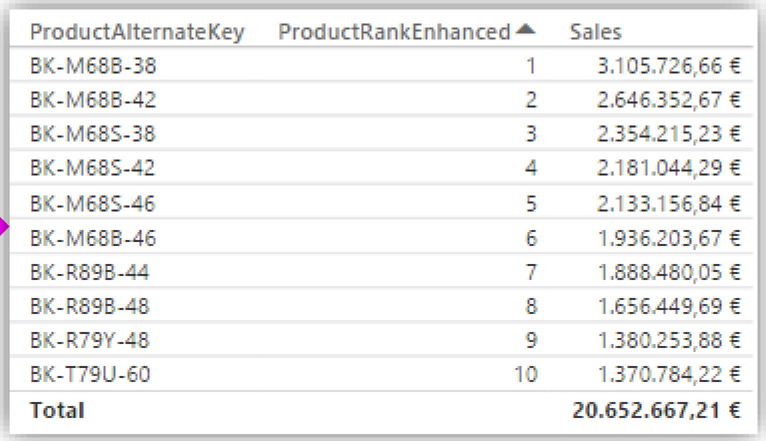
```
TOPN ( 10; VALUES ( DimProduct[ProductAlternateKey] ); [Sales]; DESC )
```

- The **VALUES** function is required to remove duplicates, if they exist. Not needed if they don't



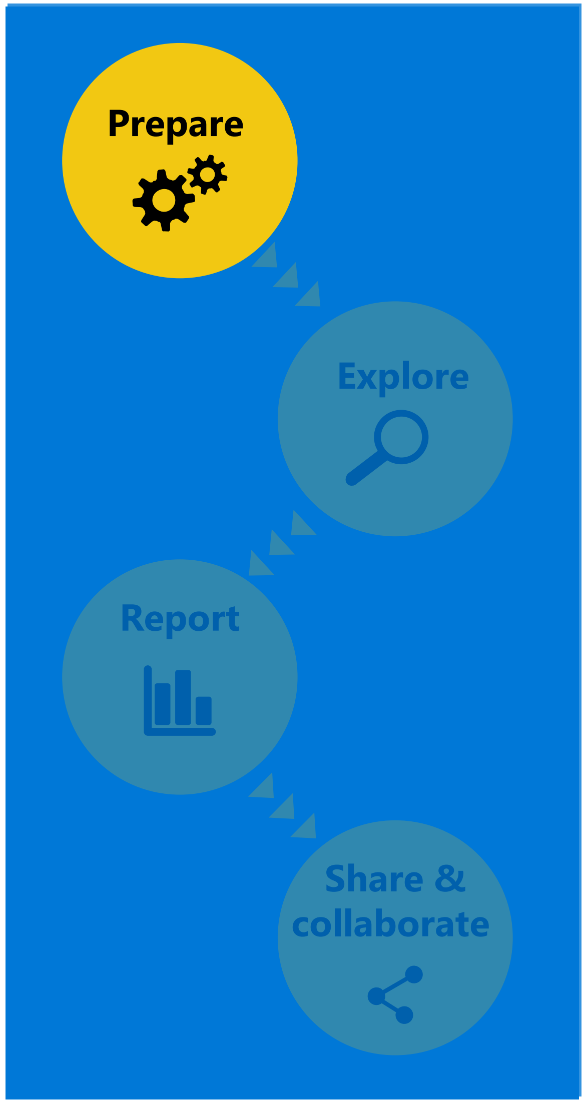
ProductAlternateKey
BK-M68B-38
BK-M68B-42
BK-M68B-46
BK-M68S-38
BK-M68S-42
BK-M68S-46
BK-R79Y-48
BK-R89B-44
BK-R89B-48
BK-T79U-60

TOPN matches RankX



ProductAlternateKey	ProductRankEnhanced	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-M68B-46	6	1.936.203,67 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-R79Y-48	9	1.380.253,88 €
BK-T79U-60	10	1.370.784,22 €
Total		**20.652.667,21 €**

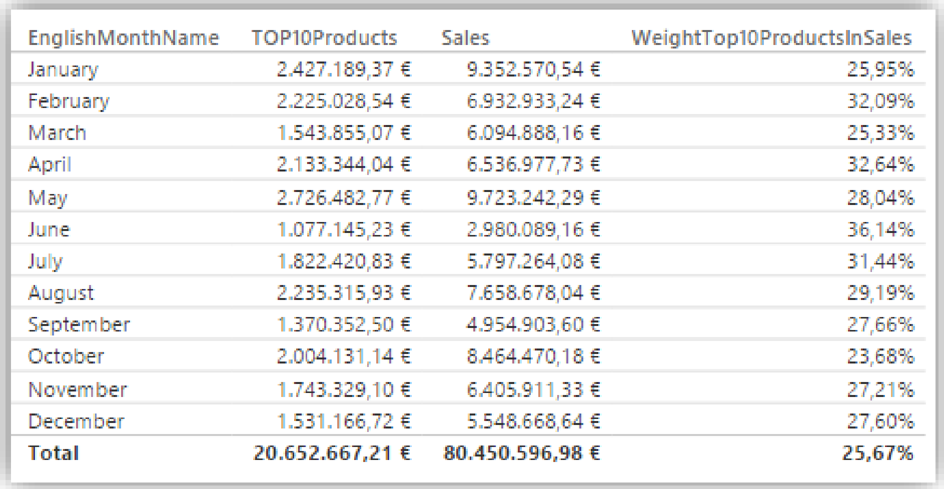
Calculations



Advanced Calculations:

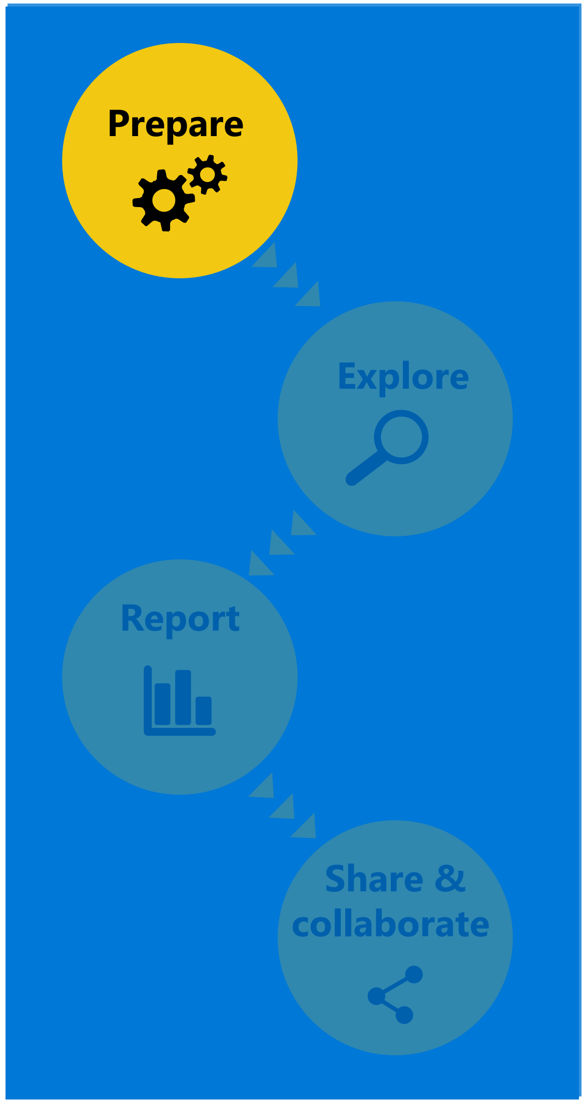
- **TOPN** does not return a scalar
- It should be then **used as context to return a scalar**

```
TOP10Products =
CALCULATE (
    [Sales];
    TOPN ( 10; VALUES ( DimProduct[ProductAlternateKey] );
[Sales]; DESC )
)
```



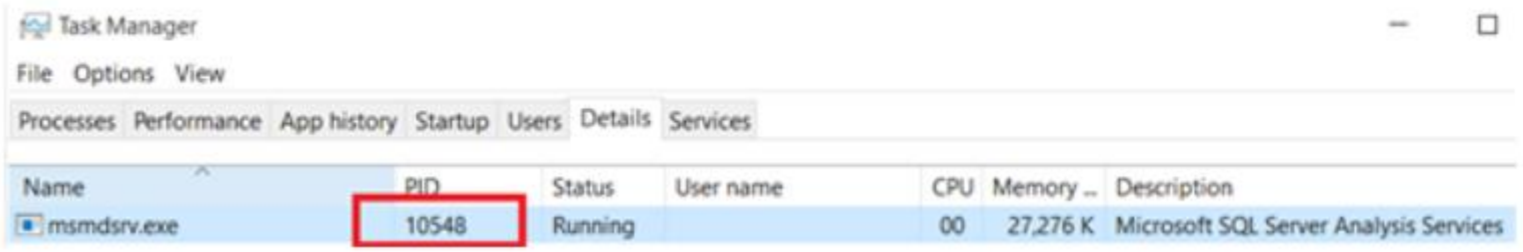
EnglishMonthName	TOP10Products	Sales	WeightTop10ProductsInSales
January	2.427.189,37 €	9.352.570,54 €	25,95%
February	2.225.028,54 €	6.932.933,24 €	32,09%
March	1.543.855,07 €	6.094.888,16 €	25,33%
April	2.133.344,04 €	6.536.977,73 €	32,64%
May	2.726.482,77 €	9.723.242,29 €	28,04%
June	1.077.145,23 €	2.980.089,16 €	36,14%
July	1.822.420,83 €	5.797.264,08 €	31,44%
August	2.235.315,93 €	7.658.678,04 €	29,19%
September	1.370.352,50 €	4.954.903,60 €	27,66%
October	2.004.131,14 €	8.464.470,18 €	23,68%
November	1.743.329,10 €	6.405.911,33 €	27,21%
December	1.531.166,72 €	5.548.668,64 €	27,60%
Total	**20.652.667,21 €**	**80.450.596,98 €**	**25,67%**

Calculations



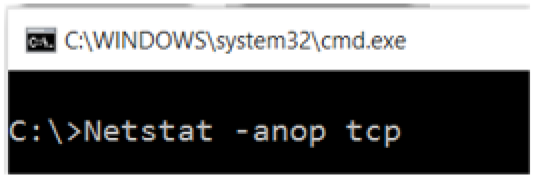
Solving DAX Problems

Step #1 - Get the process ID for the AS instance running as msmdsvr.exe from Task Manager:


Task Manager
File Options View
Processes Performance App history Startup Users Details Services

Name	PID	Status	User name	CPU	Memory ...	Description
msmdsrv.exe	10548	Running		00	27,276 K	Microsoft SQL Server Analysis Services

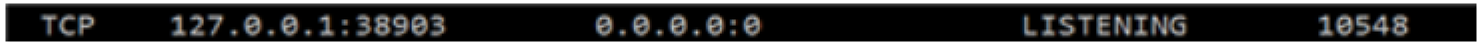
Step #2 - Determine which port the process is running on. For this, I'll use the following netstat command from a command window:



C:\WINDOWS\system32\cmd.exe

```
C:\>Netstat -anop tcp
```

From the results, find the entry that includes the PID discovered in step #1 (10548 in this example). Once we discover this entry, we located the port number following the semicolon after the IP address. Below we see 127.0.0.1:38903, showing us that it is running on port 38903.



```
 TCP    127.0.0.1:38903        0.0.0.0:0              LISTENING       10548
```

Calculations

Prepare

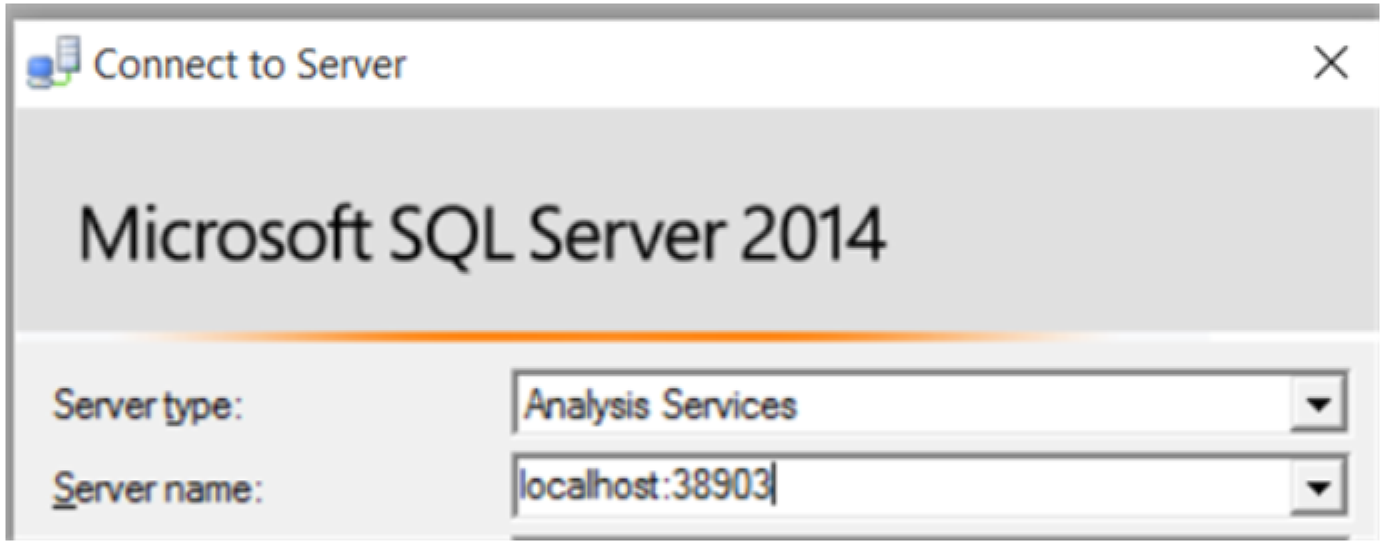
Explore

Report

Share & collaborate

Solving DAX Problems

Step #3 - Connect to the AS instance with SQL Server Profiler - In this case, the server name will be localhost: followed by the port number discovered in step #2.


Connect to Server

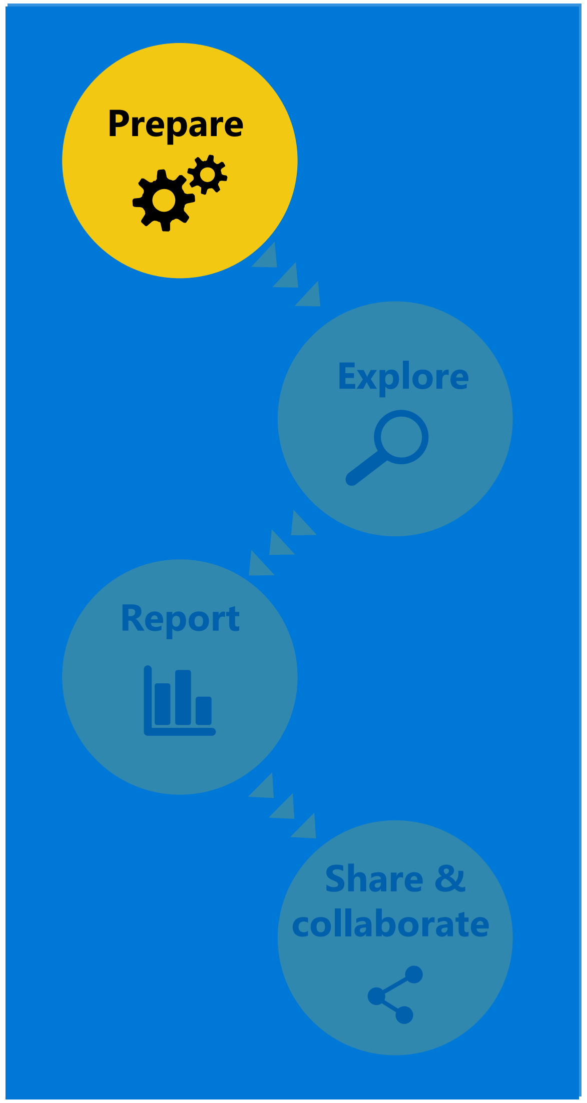
Microsoft SQL Server 2014

Server type: Analysis Services

Server name: localhost:38903

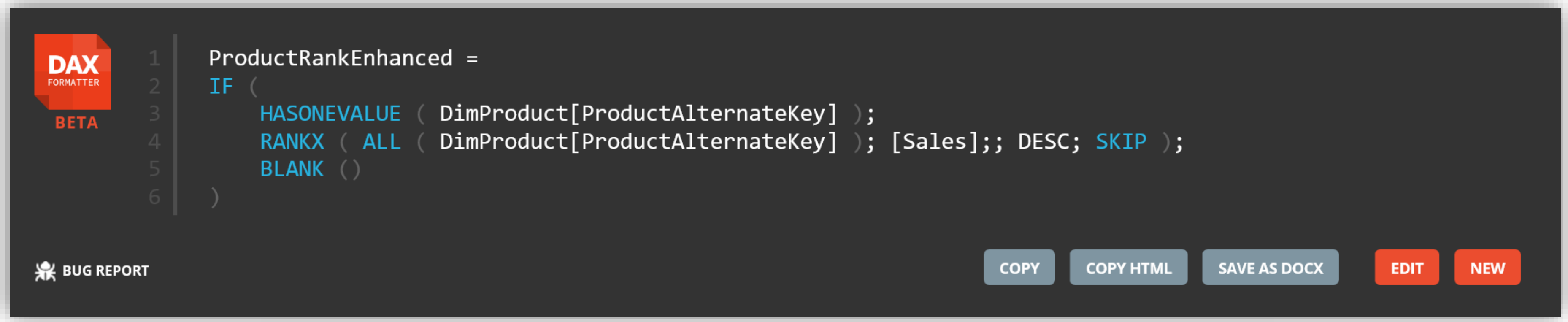
After connecting to our instance, we will begin to see the queries being executed. You can also use this to capture a trace for later investigation.

Calculations

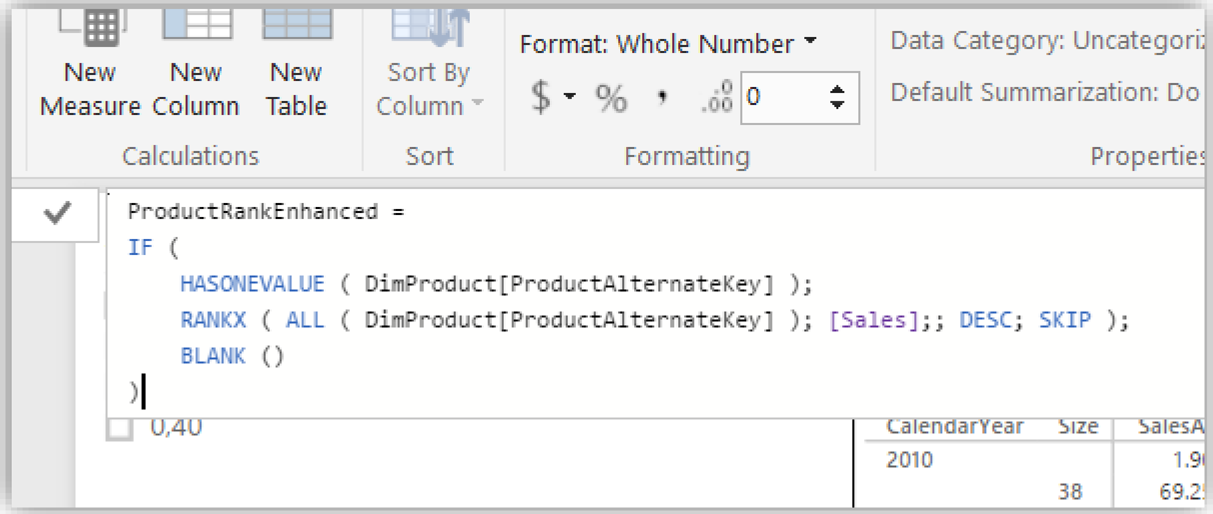


Formatting DAX

- As a good practice, **format** your code. DAXFormatter from SQLBI helps



```
ProductRankEnhanced =
IF (
    HASONEVALUE ( DimProduct[ProductAlternateKey] );
    RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP );
    BLANK ()
)
```



```
ProductRankEnhanced =
IF (
    HASONEVALUE ( DimProduct[ProductAlternateKey] );
    RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP );
    BLANK ()
)
```

CalendarYear	Size	SalesA
2010		1.9
	38	69.2

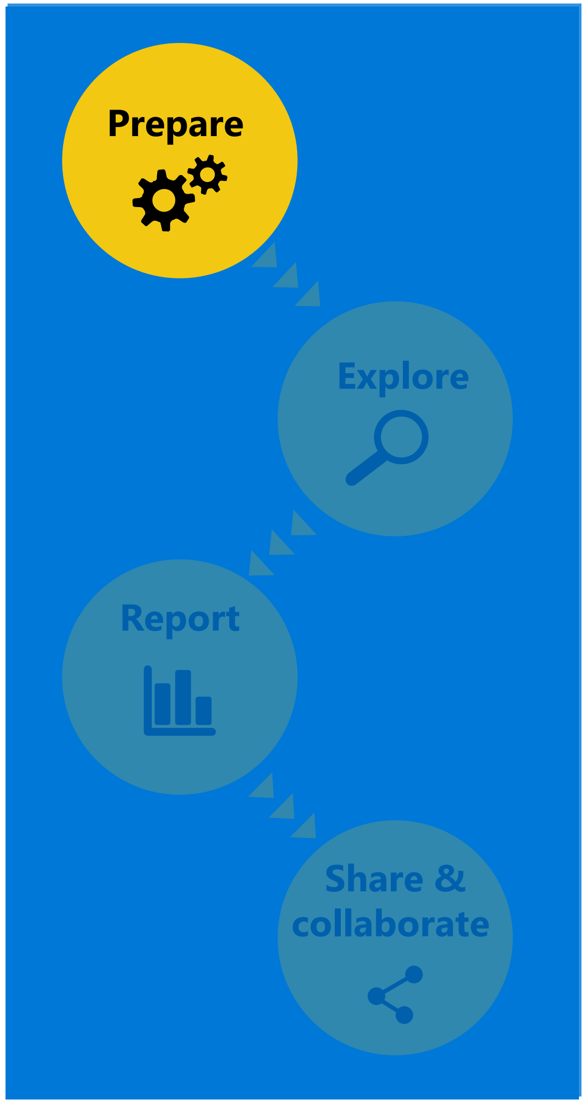
0,40

Demonstration:

Disconnected Tables

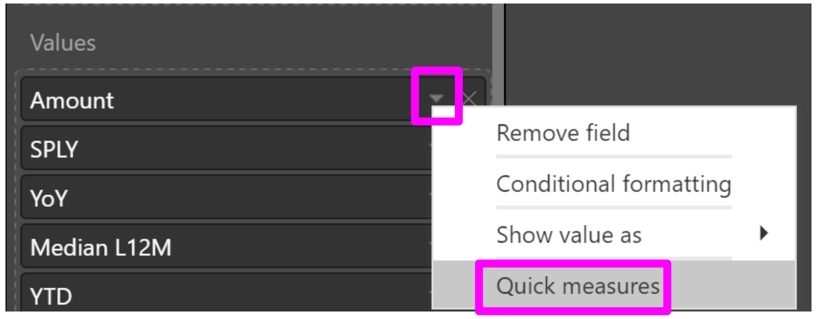
Lab 1 Exercise 3:
Extending the Data Model

Calculations

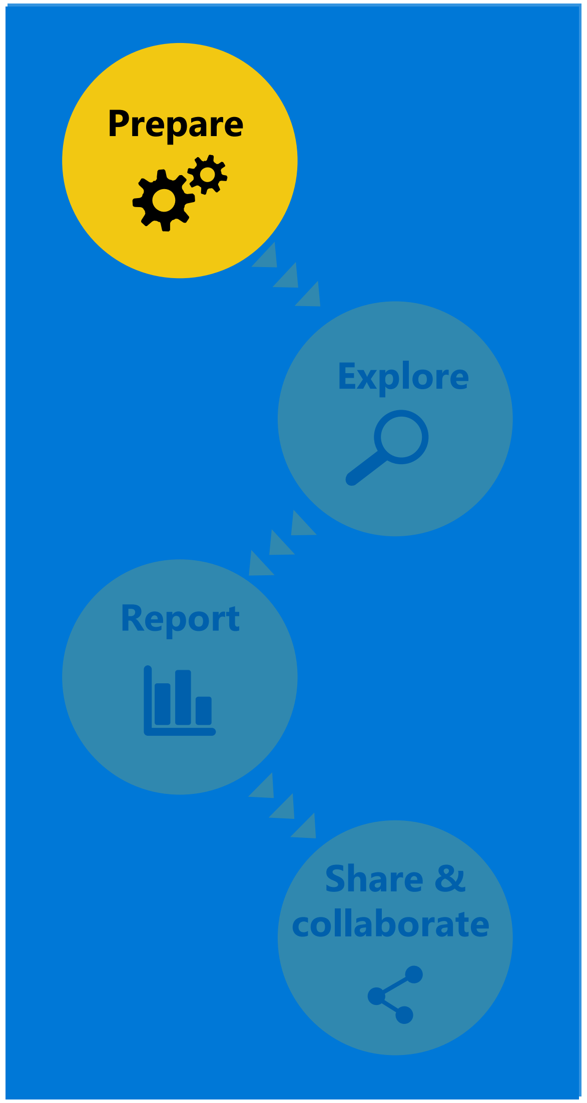


Quick Measures:

- Allow you to **build calculations, without knowing DAX**
- Also **works with** live connection mode against **SSAS models**.
- Available calculations are organized in **6 categories**:
  - Aggregate within category
  - Filters and baselines
  - Time Intelligence
  - Running Total
  - Mathematical Operations
  - Text
- Start with an element on the field well or on the field list and create calculation
- It's great for **learning**!

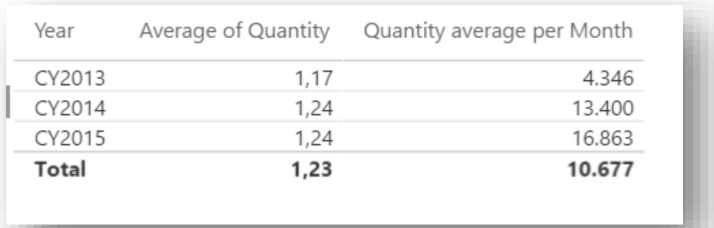


Calculations

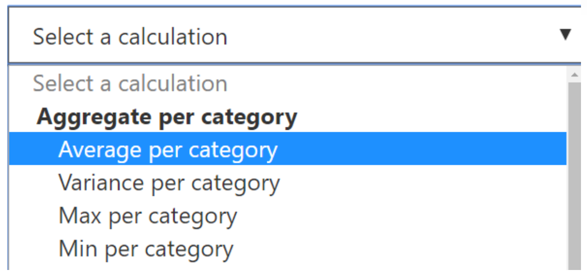


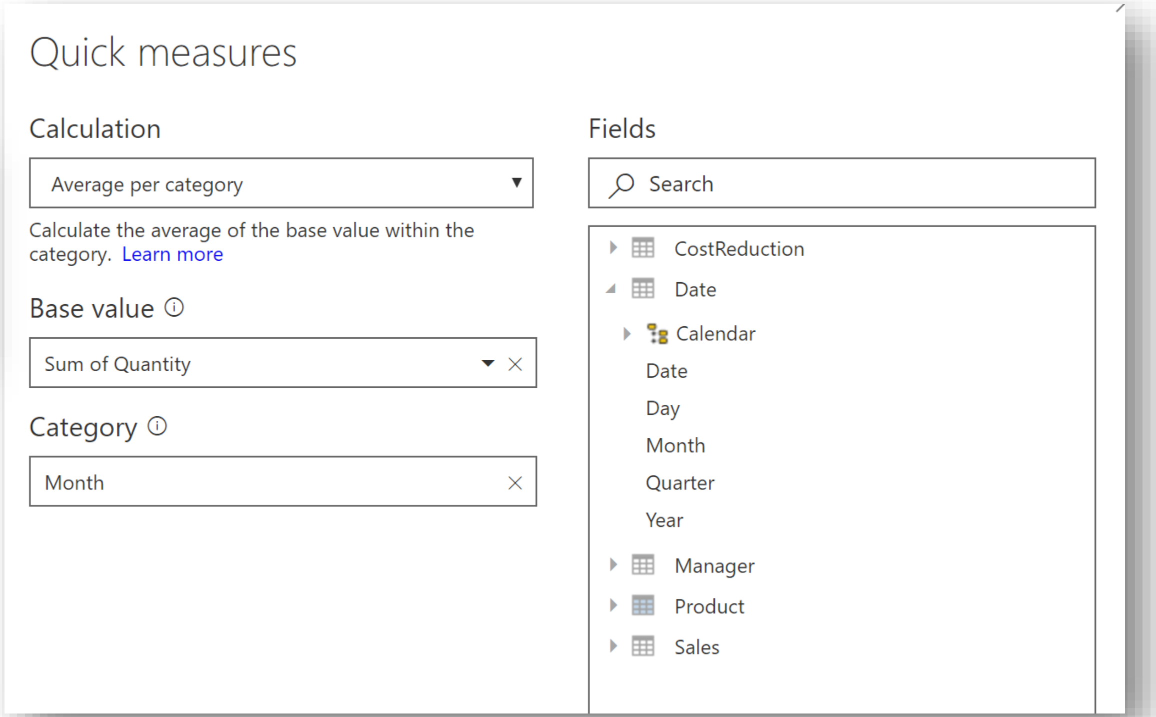
Quick Measures:

- **Aggreagate within category** – Allows applying different aggregates at different levels:



Year	Average of Quantity	Quantity average per Month
CY2013	1,17	4.346
CY2014	1,24	13.400
CY2015	1,24	16.863
Total	**1,23**	**10.677**





Quick measures

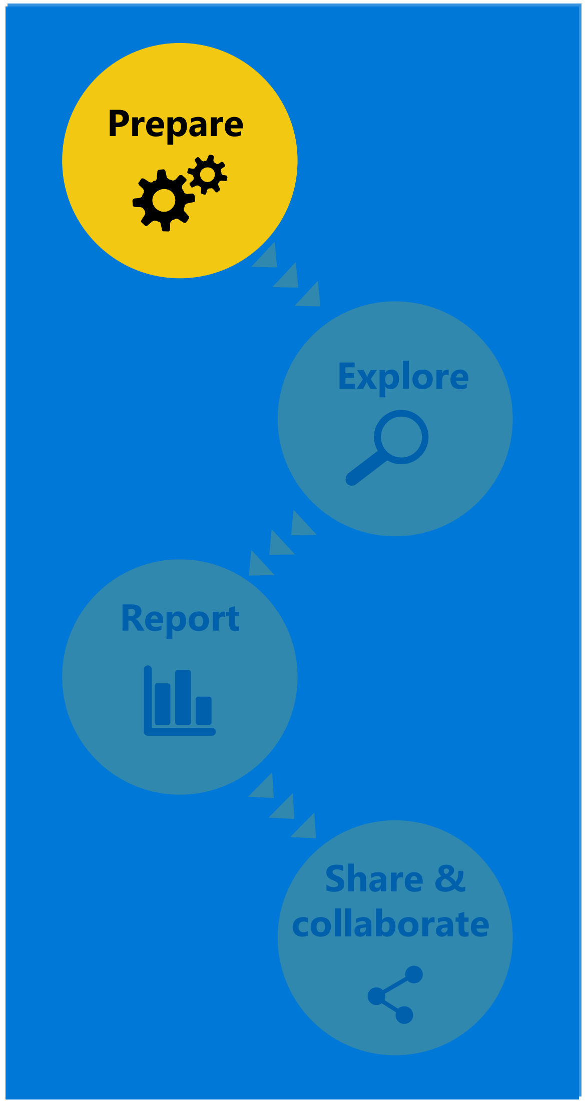
Calculation: Average per category

Calculate the average of the base value within the category. Learn more

Base value: Sum of Quantity

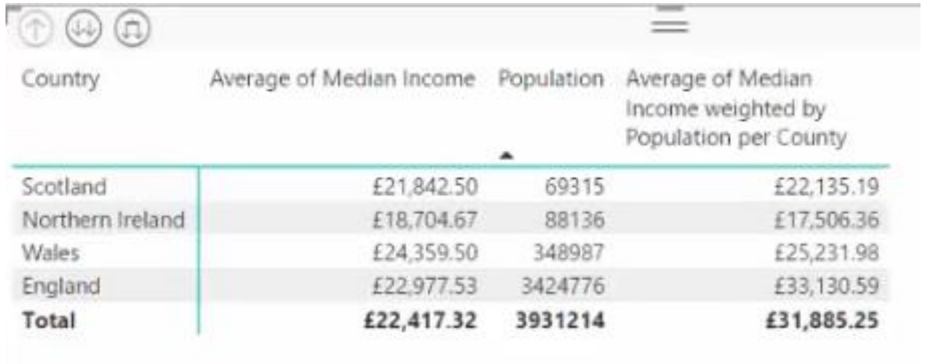
Category: Month

Calculations

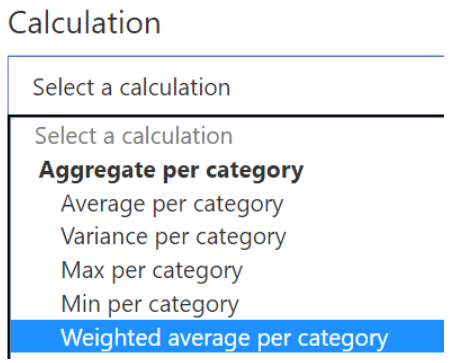


Quick Measures:

- **Weighted Average per Category** – calculates a weighted average of the base value for each category



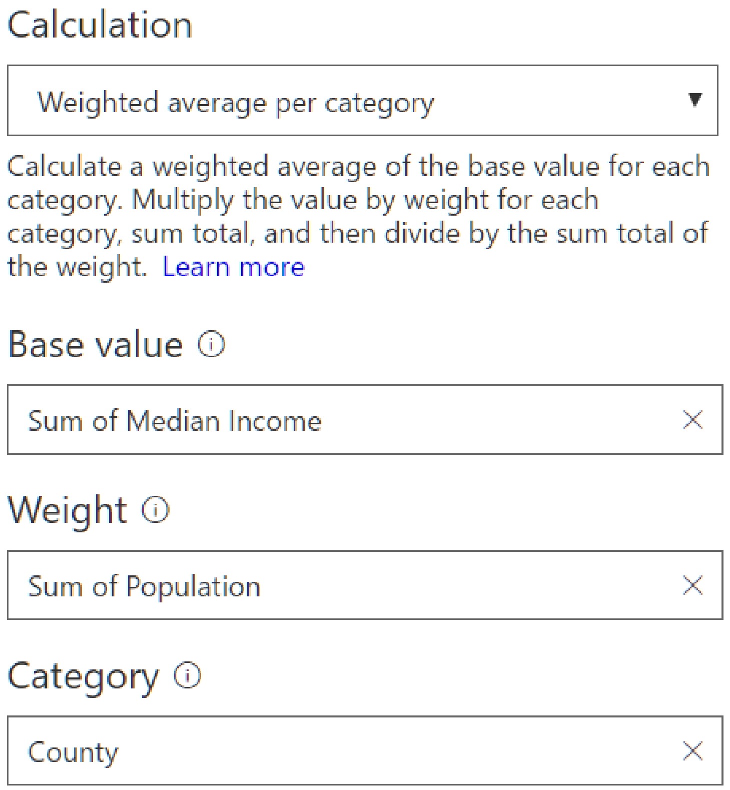
Country	Average of Median Income	Population	Average of Median Income weighted by Population per County
Scotland	£21,842.50	69315	£22,135.19
Northern Ireland	£18,704.67	88136	£17,506.36
Wales	£24,359.50	348987	£25,231.98
England	£22,977.53	3424776	£33,130.59
Total	**£22,417.32**	**3931214**	**£31,885.25**



Calculation

Select a calculation

Select a calculation
Aggregate per category
Average per category
Variance per category
Max per category
Min per category
Weighted average per category



Calculation

Weighted average per category ▾

Calculate a weighted average of the base value for each category. Multiply the value by weight for each category, sum total, and then divide by the sum total of the weight. Learn more

Base value ⓘ

Sum of Median Income ✕

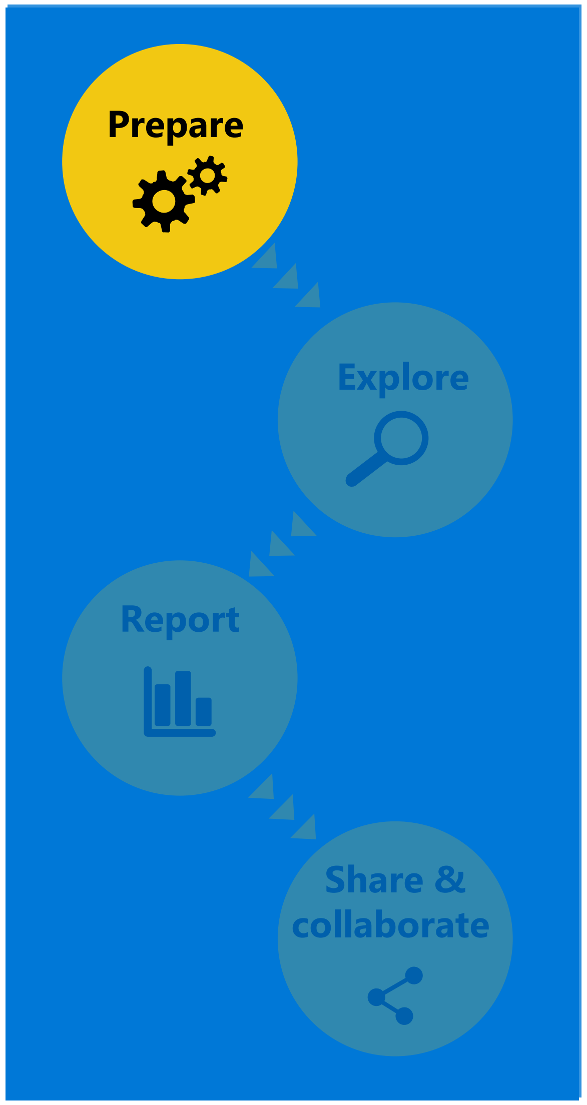
Weight ⓘ

Sum of Population ✕

Category ⓘ

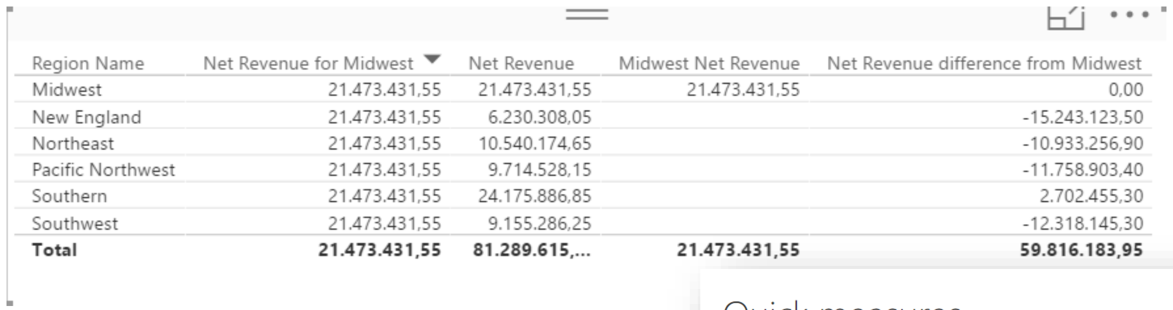
County ✕

Calculations

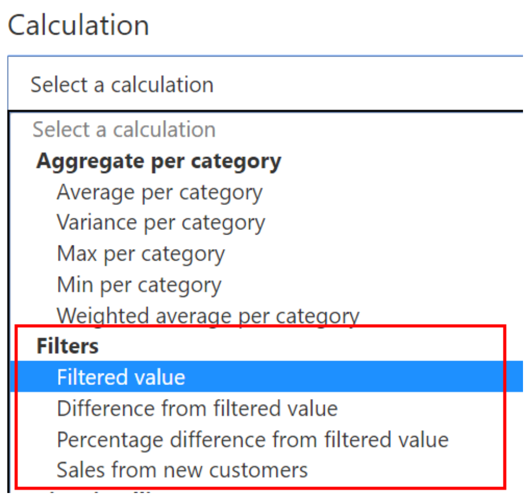


Quick Measures:

- **Filters and baselines** – calculate values for a specific category in a column, or compare values to a specific baseline:



Region Name	Net Revenue for Midwest	Net Revenue	Midwest Net Revenue	Net Revenue difference from Midwest
Midwest	21.473.431,55	21.473.431,55	21.473.431,55	0,00
New England	21.473.431,55	6.230.308,05		-15.243.123,50
Northeast	21.473.431,55	10.540.174,65		-10.933.256,90
Pacific Northwest	21.473.431,55	9.714.528,15		-11.758.903,40
Southern	21.473.431,55	24.175.886,85		2.702.455,30
Southwest	21.473.431,55	9.155.286,25		-12.318.145,30
Total	**21.473.431,55**	**81.289.615,...**	**21.473.431,55**	**59.816.183,95**



Calculation

Select a calculation

Select a calculation

Aggregate per category

Average per category

Variance per category

Max per category

Min per category

Weighted average per category

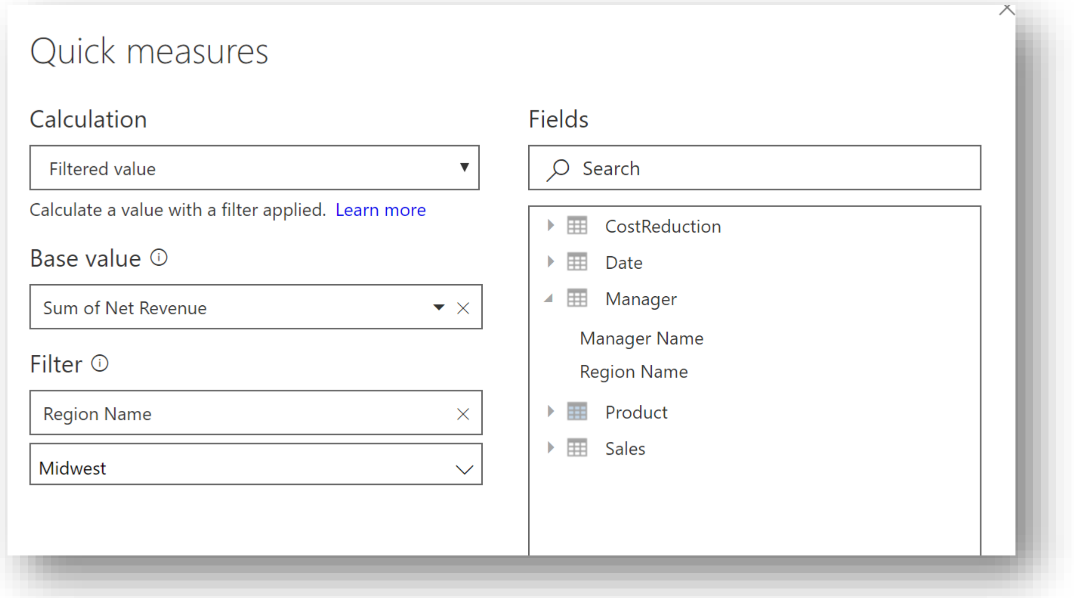
Filters

Filtered value

Difference from filtered value

Percentage difference from filtered value

Sales from new customers



Quick measures

Calculation: Filtered value

Calculate a value with a filter applied. Learn more

Base value: Sum of Net Revenue

Filter: Region Name

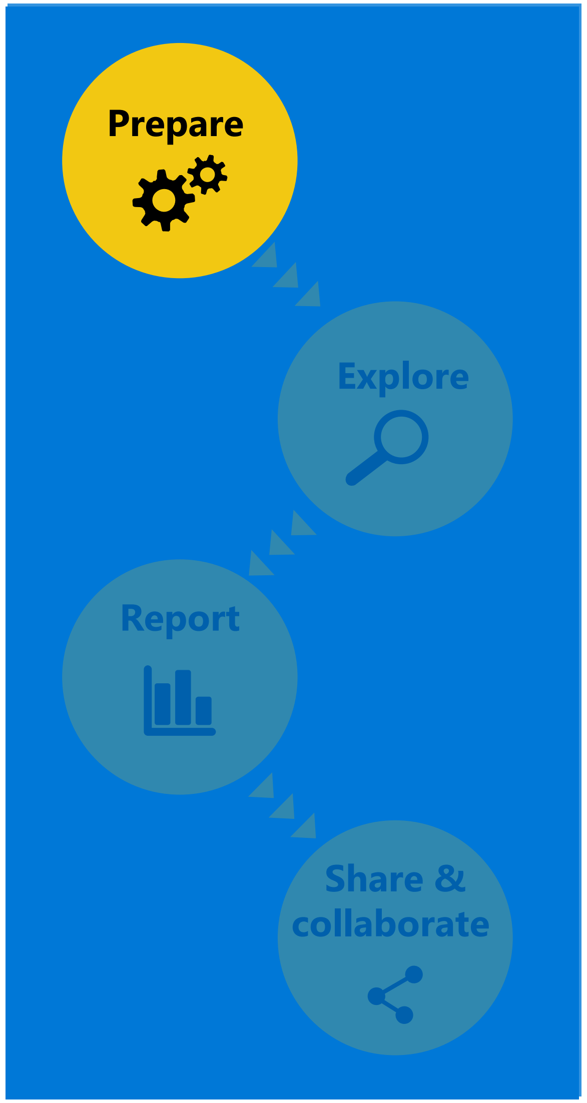
Midwest

Fields

Search

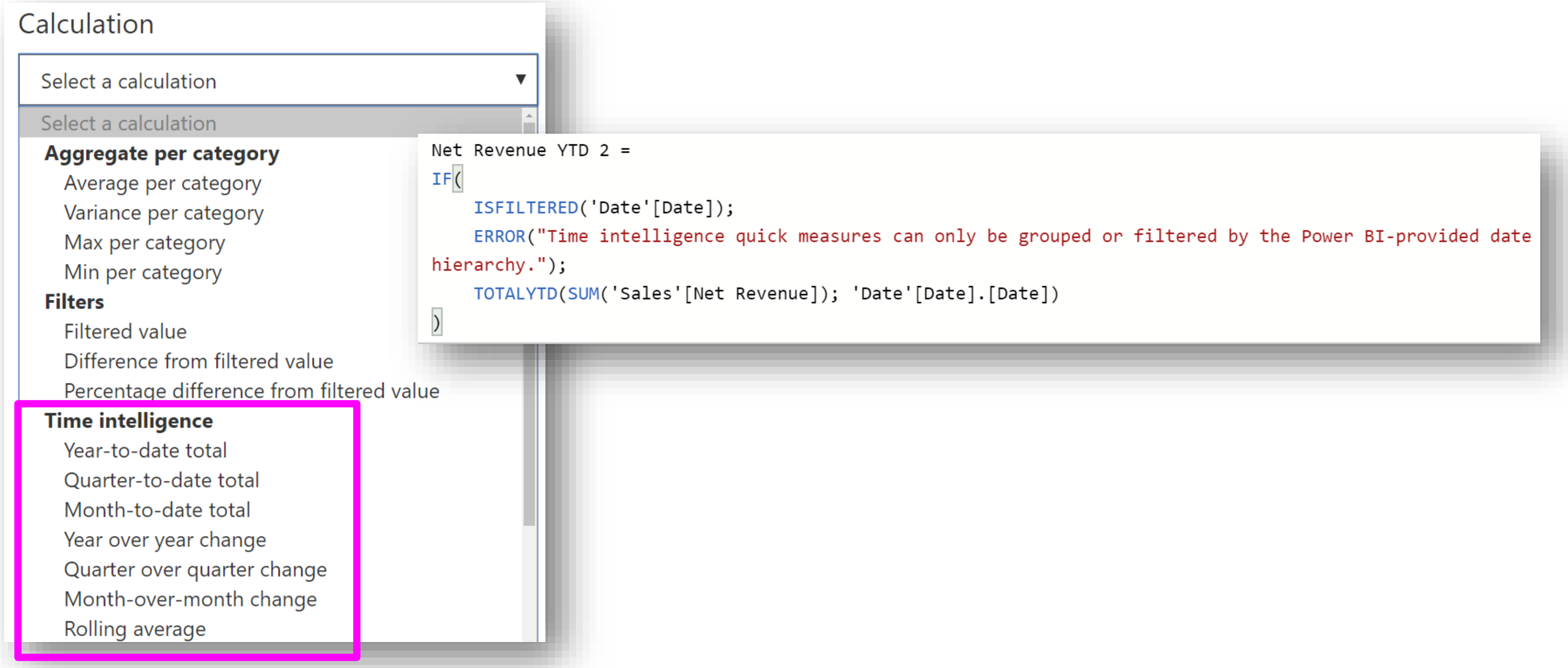
CostReduction
Date
Manager
Manager Name
Region Name
Product
Sales

Calculations



Quick Measures:

- **Time Intelligence** – Time-based calculations (currently only for native date tables):



Calculation

Select a calculation

Select a calculation

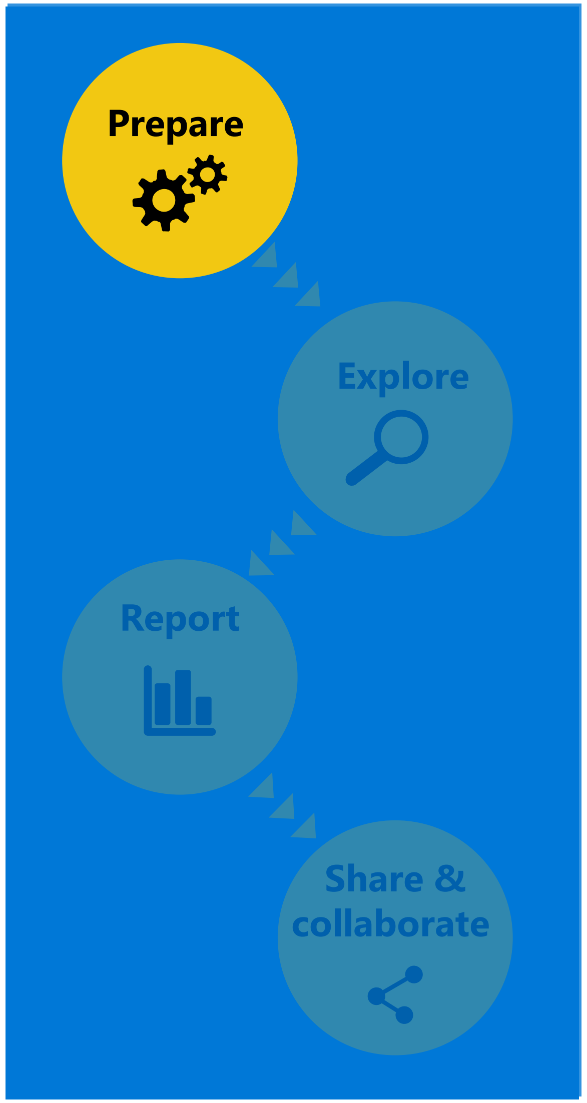
Aggregate per category
- Average per category
- Variance per category
- Max per category
- Min per category

Filters
- Filtered value
- Difference from filtered value
- Percentage difference from filtered value

Time intelligence
- Year-to-date total
- Quarter-to-date total
- Month-to-date total
- Year over year change
- Quarter over quarter change
- Month-over-month change
- Rolling average

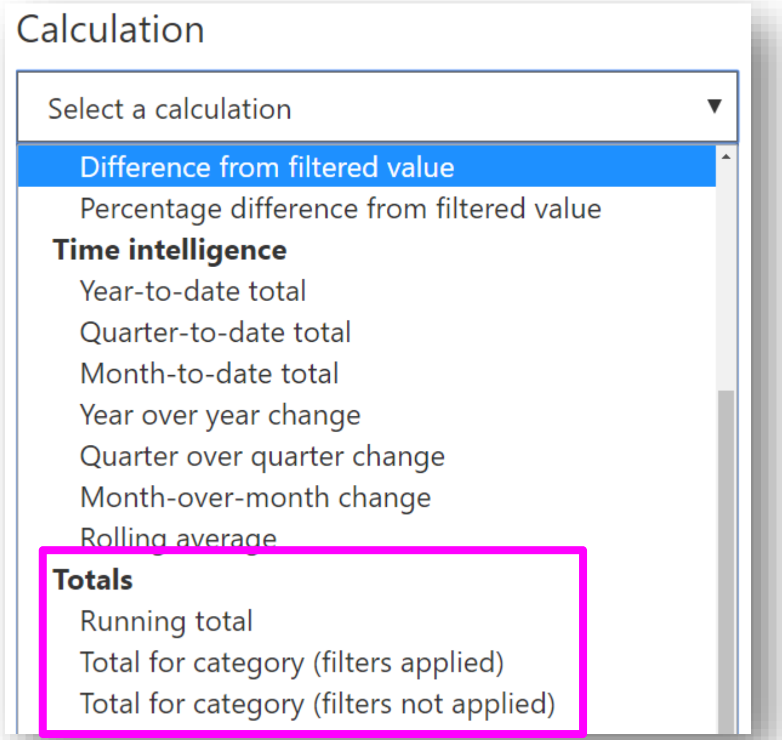
```
Net Revenue YTD 2 =
IF(
    ISFILTERED('Date'[Date]);
    ERROR("Time intelligence quick measures can only be grouped or filtered by the Power BI-provided date
hierarchy.");
    TOTALYTD(SUM('Sales'[Net Revenue]); 'Date'[Date].[Date])
)
```

Calculations



Quick Measures:

- **Totals** – Calculations meant to utilize totals in different ways



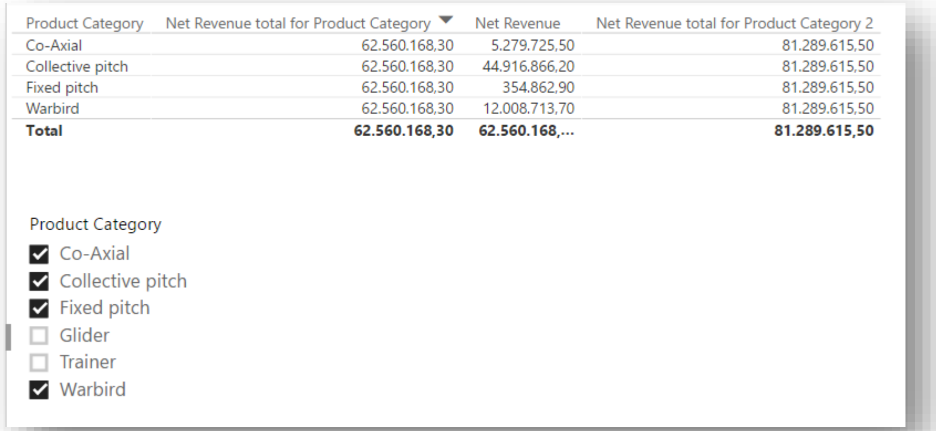
Calculation

Select a calculation ▼

- Difference from filtered value
- Percentage difference from filtered value

Time intelligence
- Year-to-date total
- Quarter-to-date total
- Month-to-date total
- Year over year change
- Quarter over quarter change
- Month-over-month change
- Rolling average

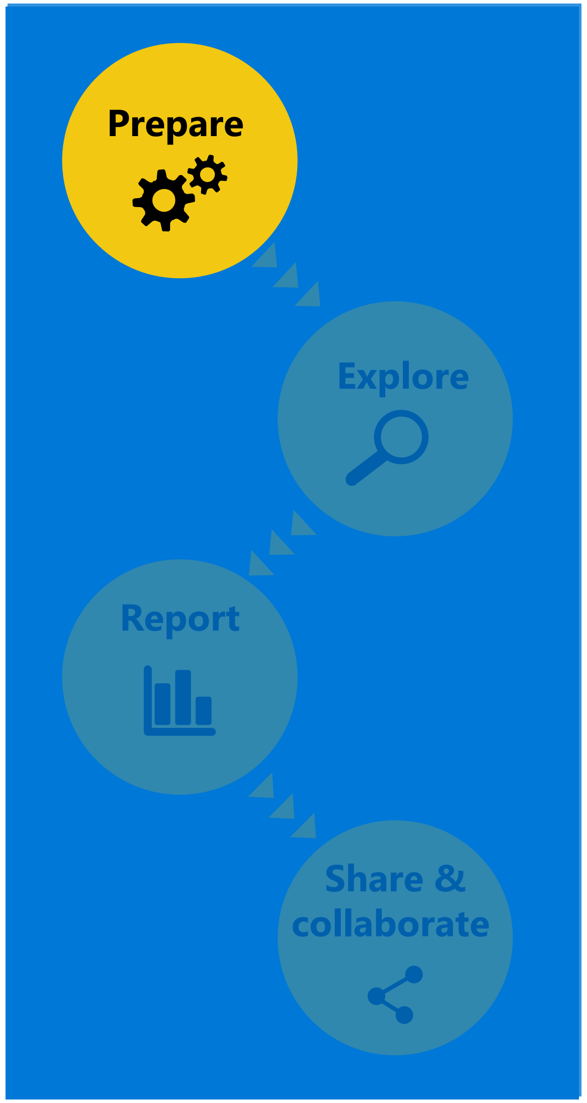
Totals
- Running total
- Total for category (filters applied)
- Total for category (filters not applied)



Product Category	Net Revenue total for Product Category	Net Revenue	Net Revenue total for Product Category 2
Co-Axial	62.560.168,30	5.279.725,50	81.289.615,50
Collective pitch	62.560.168,30	44.916.866,20	81.289.615,50
Fixed pitch	62.560.168,30	354.862,90	81.289.615,50
Warbird	62.560.168,30	12.008.713,70	81.289.615,50
Total	**62.560.168,30**	**62.560.168,...**	**81.289.615,50**

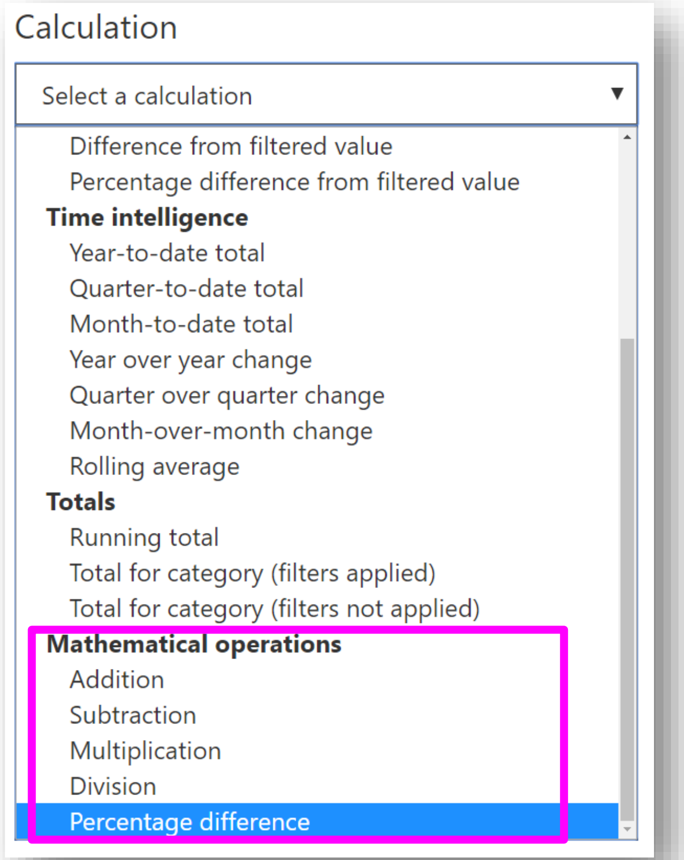
Product Category
- ✓ Co-Axial
- ✓ Collective pitch
- ✓ Fixed pitch
- ☐ Glider
- ☐ Trainer
- ✓ Warbird

Calculations



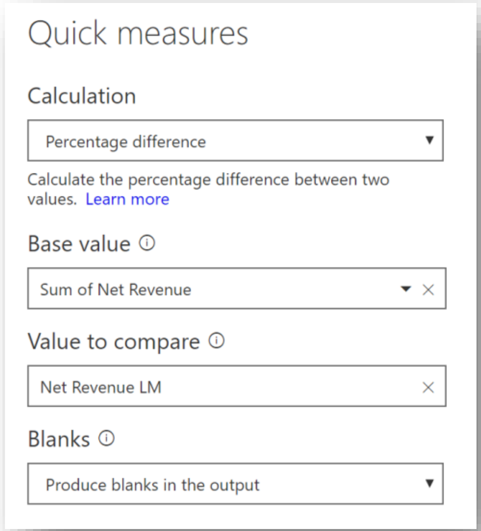
Quick Measures:

- **Mathematical Operations** – Simplifying implementation of easy calculations


Calculation

Select a calculation

Difference from filtered value
Percentage difference from filtered value
Time intelligence
Year-to-date total
Quarter-to-date total
Month-to-date total
Year over year change
Quarter over quarter change
Month-over-month change
Rolling average
Totals
Running total
Total for category (filters applied)
Total for category (filters not applied)
Mathematical operations
Addition
Subtraction
Multiplication
Division
Percentage difference


Quick measures

Calculation: Percentage difference

Calculate the percentage difference between two values. Learn more

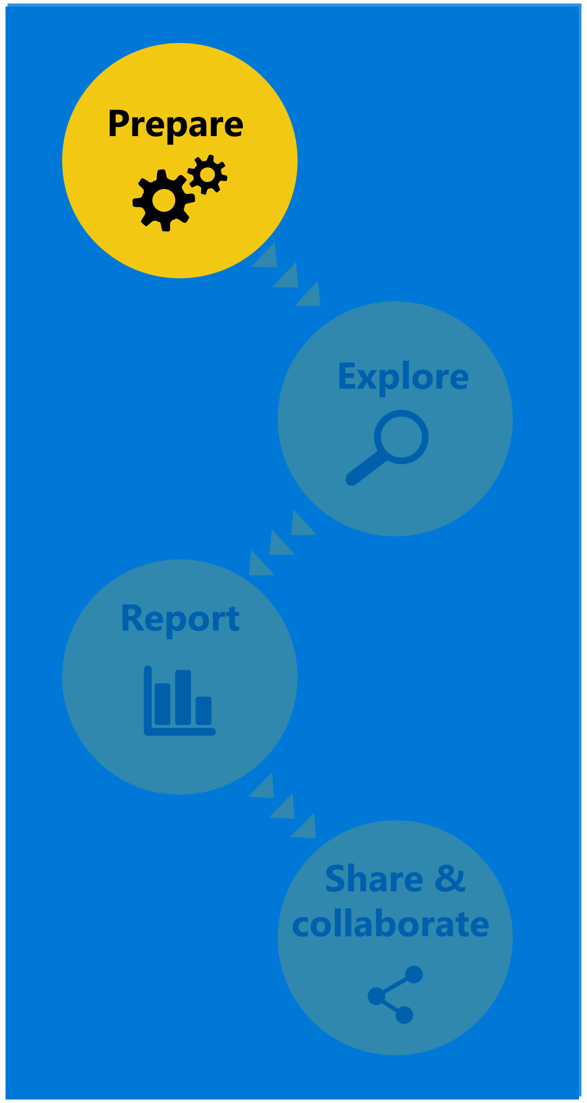
Base value: Sum of Net Revenue

Value to compare: Net Revenue LM

Blanks: Produce blanks in the output

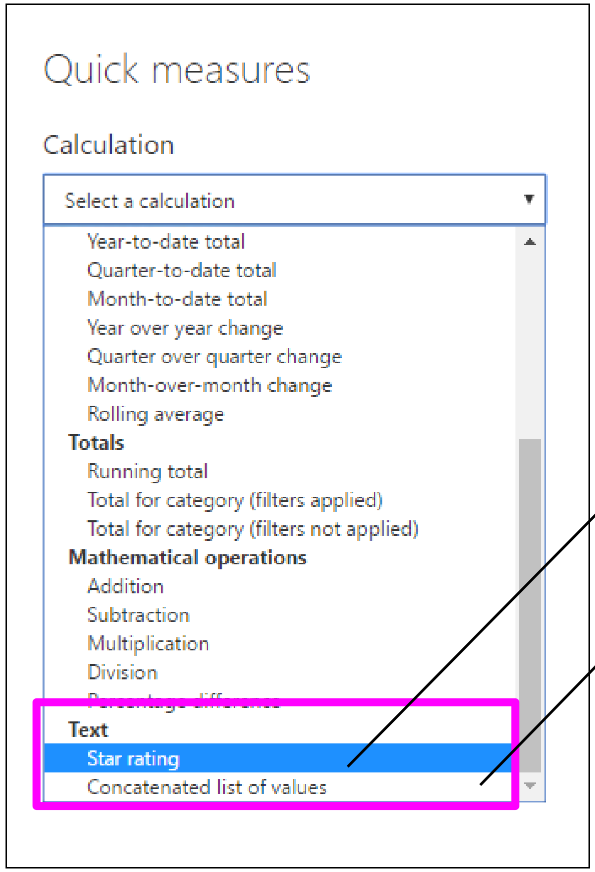
```
Net Revenue LM % difference from Net Revenue =
VAR __BASELINE_VALUE =
    SUM('Sales'[Net Revenue])
VAR __VALUE_TO_COMPARE =
    'Sales'[Net Revenue LM]
RETURN
    IF(
        NOT ISBLANK(__VALUE_TO_COMPARE);
        DIVIDE(__VALUE_TO_COMPARE - __BASELINE_VALUE; __BASELINE_VALUE)
    )
```

Calculations



Quick Measures:

- **Text** – Analytics and modeling



Quick measures

Calculation

Select a calculation

- Year-to-date total
- Quarter-to-date total
- Month-to-date total
- Year over year change
- Quarter over quarter change
- Month-over-month change
- Rolling average

Totals

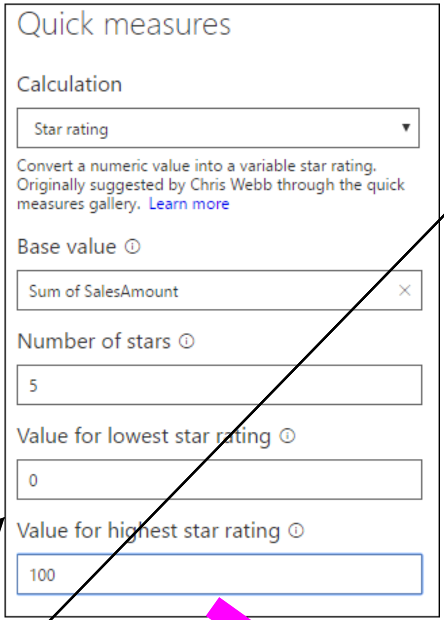
- Running total
- Total for category (filters applied)
- Total for category (filters not applied)

Mathematical operations

- Addition
- Subtraction
- Multiplication
- Division
- Percentage difference

Text

- Star rating
- Concatenated list of values



Quick measures

Calculation: Star rating

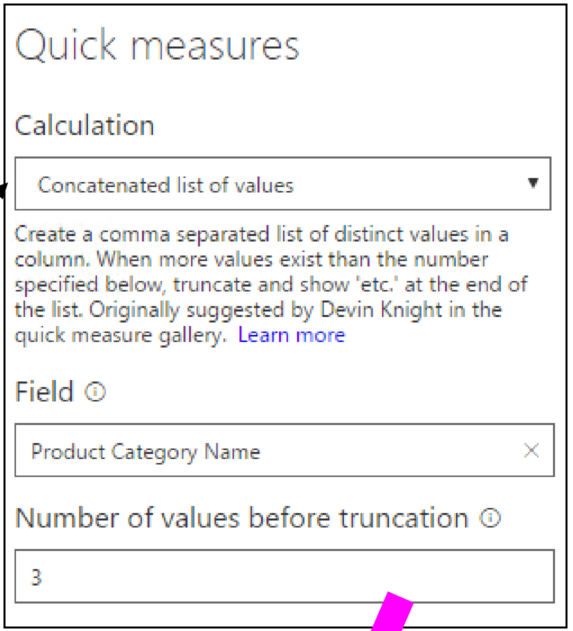
Convert a numeric value into a variable star rating. Originally suggested by Chris Webb through the quick measures gallery. Learn more

Base value: Sum of SalesAmount

Number of stars: 5

Value for lowest star rating: 0

Value for highest star rating: 100



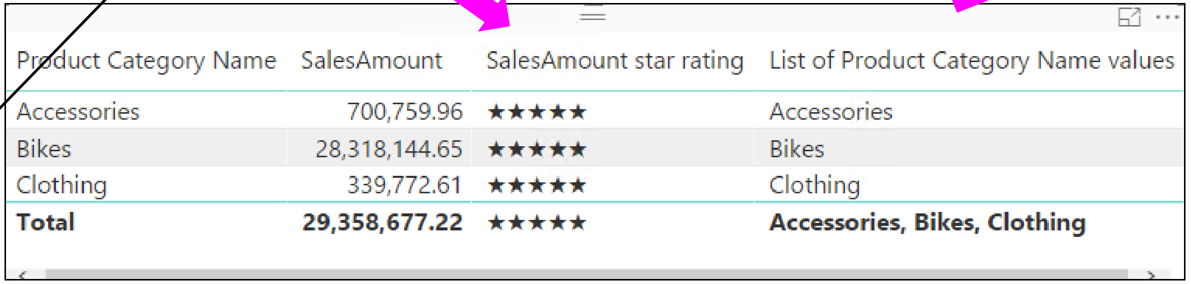
Quick measures

Calculation: Concatenated list of values

Create a comma separated list of distinct values in a column. When more values exist than the number specified below, truncate and show 'etc.' at the end of the list. Originally suggested by Devin Knight in the quick measure gallery. Learn more

Field: Product Category Name

Number of values before truncation: 3



Product Category Name	SalesAmount	SalesAmount star rating	List of Product Category Name values
Accessories	700,759.96	★★★★★	Accessories
Bikes	28,318,144.65	★★★★★	Bikes
Clothing	339,772.61	★★★★★	Clothing
Total	**29,358,677.22**	★★★★★	**Accessories, Bikes, Clothing**