Numeric Calculations with DAX

Markus Ehrenmueller-Jensen

DAX

- Data Analysis Expressions (not Data Analytic Expressions)
- Born with “PowerPivot for Excel”
  - Lives in Power BI and Analysis Services Tabular as well
- Influenced by other languages
  - Excel (function based syntax, 100+ functions share same source code)
  - SQL (relational function, row context)
  - MDX (measures, implicit join, filter context)
- Language for both, measures and calculated columns
  - And: calculated tables, row level security, queries
- Easy syntax
- Complex semantic
  - Model based
  - Allowing for powerful computations

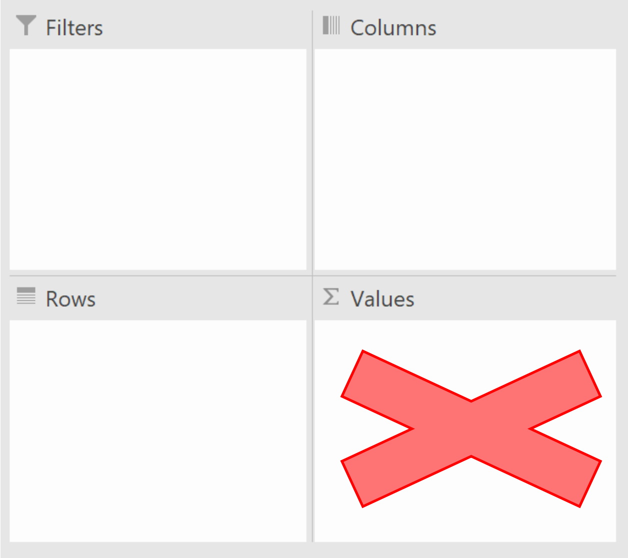
Sample Calculations

$Sales\ Amount$

$Margin$

$Margin\ in\ \%$

Calculated Columns

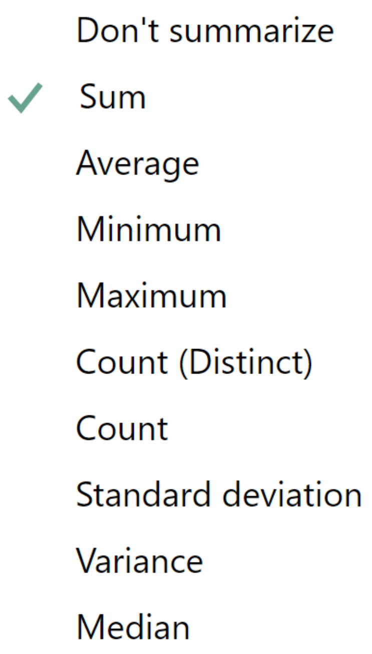


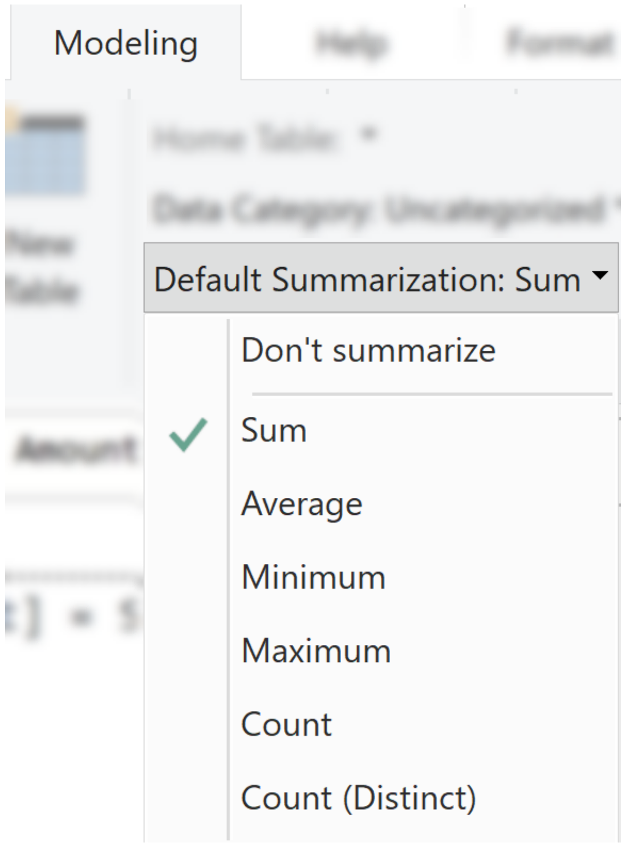
- Convention: `'Table'[Column]`
  - Specify the table name
- Calculation is done row-by-row
  - Every row has its own *row context*
  - And persisted in the model during model refresh
- Data type is determined by formula
  - Can be changed by user
- We will concentrate on **numeric** (calculated) columns
- Excel can use calculated columns as filter only
  - You can put them on *Filters*, *Columns* & *Rows* of a pivot table
  - But you can’t put them in the V*alues* field
- Power BI automatically creates an implicit measure
  - Simple aggregations which are driven by column property *Default Summarization*
  - Measure is invisible
  - Well-intentioned … can be the opposite of well done

Default Summarization

Modelling property of (calculated) column

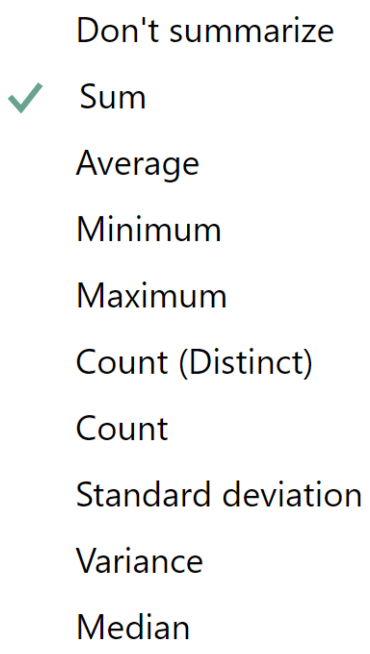
Can be changed per visual





Calculated Columns in Power BI

$$Implicit\ Measure\ = \sum(Calculated\ Column's\ expression)$$



Don't summarize

✓ Sum

Average

Minimum

Maximum

Count (Distinct)

Count

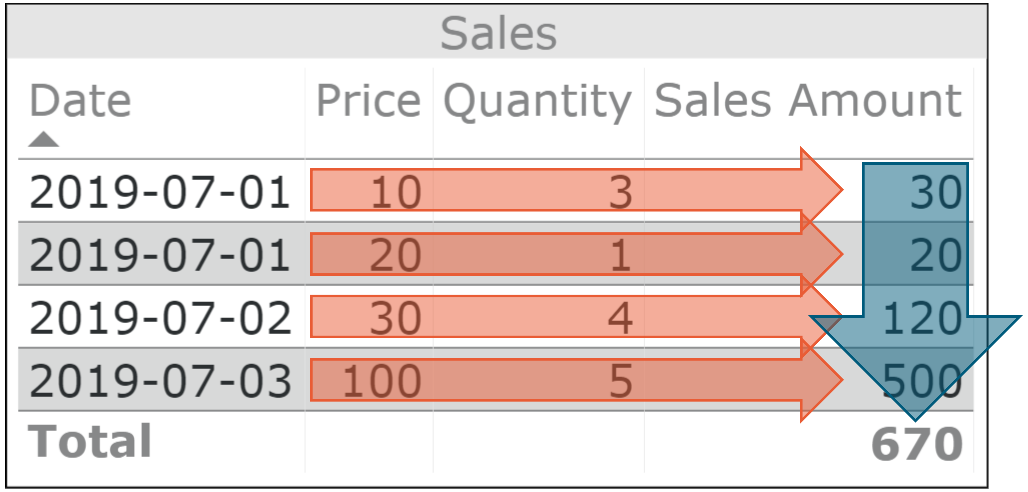
Standard deviation

Variance

Median

'Sales' [Sales Amount]

```
= 'Sales'[Price] * 'Sales'[Quantity]
```

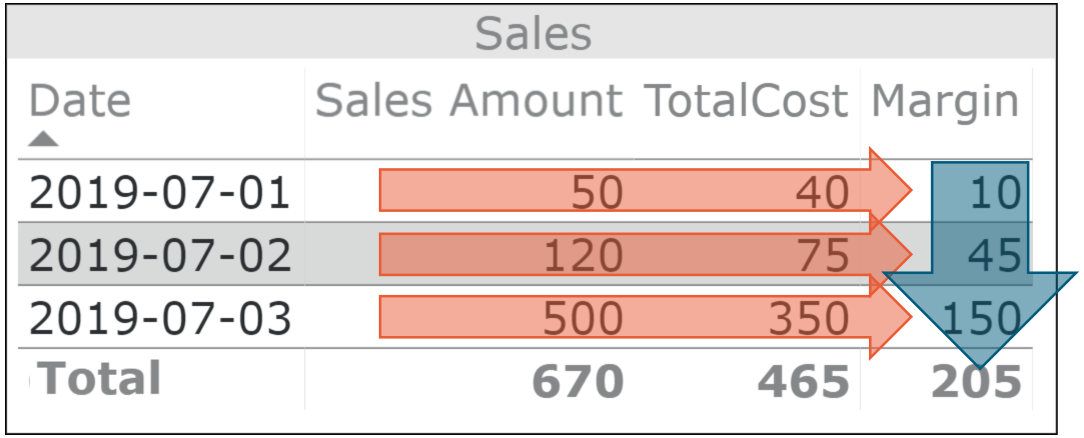


Sales			
Date	Price	Quantity	Sales Amount
2019-07-01	10	3	30
2019-07-01	20	1	20
2019-07-02	30	4	120
2019-07-03	100	5	500
Total			**670**

implicit measure

'Sales'[Margin]

```
= 'Sales'[Sales Amount] - 'Sales'[TotalCost]
```

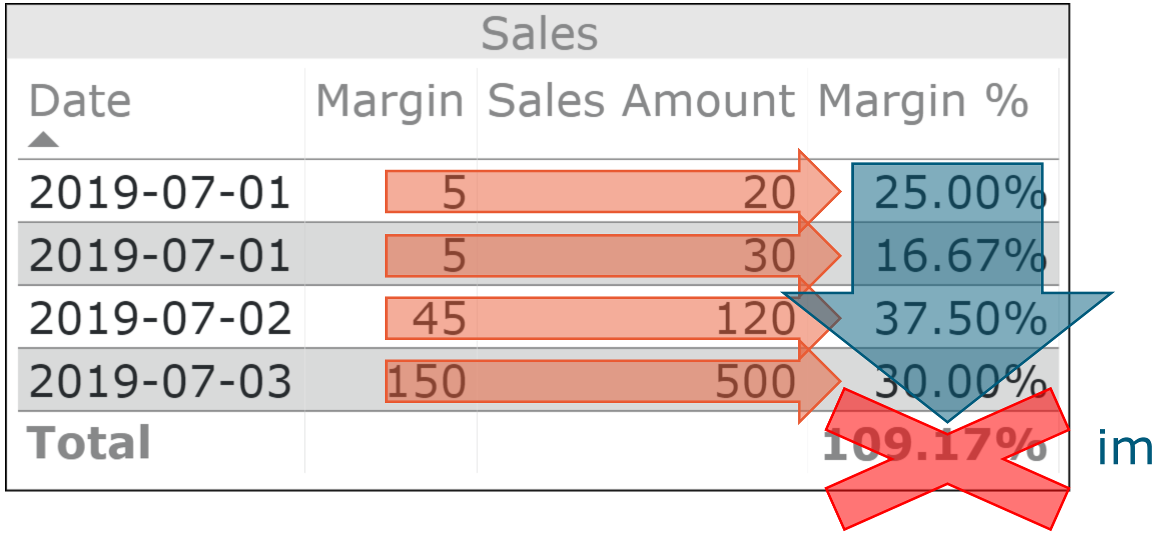


Sales			
Date	Sales Amount	TotalCost	Margin
2019-07-01	50	40	10
2019-07-02	120	75	45
2019-07-03	500	350	150
Total	**670**	**465**	**205**

implicit measure

'Sales'[Margin %]

```
= 'Sales'[Margin] / 'Sales'[Sales Amount]
```



Sales			
Date	Margin	Sales Amount	Margin %
2019-07-01	5	20	25.00%
2019-07-01	5	30	16.67%
2019-07-02	45	120	37.50%
2019-07-03	150	500	30.00%
Total			**109.17%**

implicit measure

Measure

Convention: `[Measure]`

- Omit the table name

Lives detached from any table rows

- Calculation *context* is defined via filters, not via a single row
- Filters: slicer, cross filter, page filter, rows & columns of pivot table, …

Calculation is NOT done row-by-row

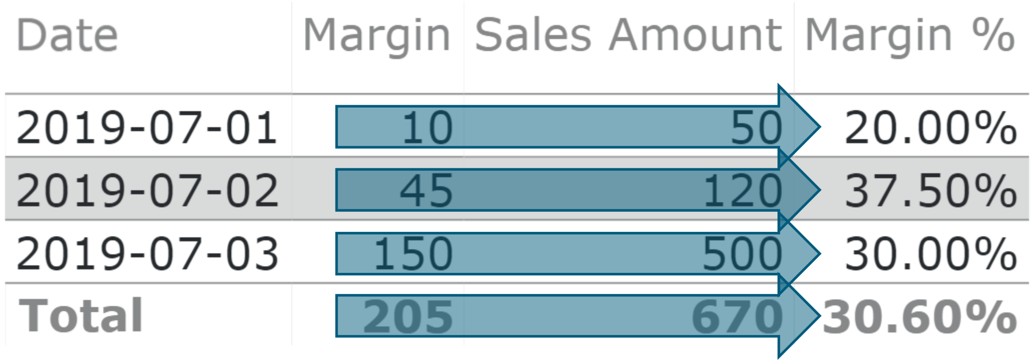
- Unless we explicitly demand so

Calculation result is NOT saved in model

Behave the same in Excel & Power BI

[Margin %]

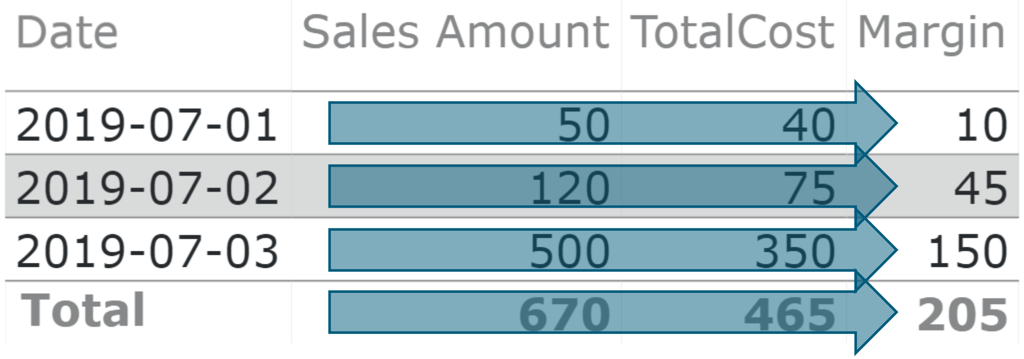
```
:= SUM('Sales'[Margin]) / SUM('Sales'[Sales Amount])
```



Date	Margin	Sales Amount	Margin %
2019-07-01	10	50	20.00%
2019-07-02	45	120	37.50%
2019-07-03	150	500	30.00%
Total	**205**	**670**	**30.60%**

[Margin]

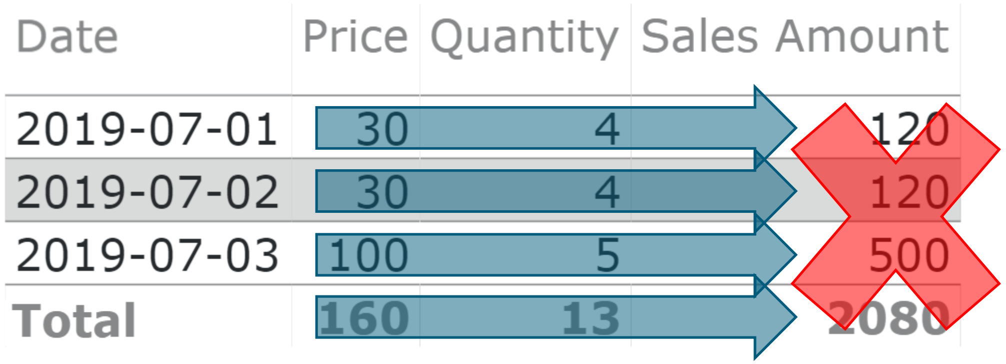
```
:= SUM(Sales[Sales Amount]) - SUM(Sales[TotalCost])
```



Date	Sales Amount	TotalCost	Margin
2019-07-01	50	40	10
2019-07-02	120	75	45
2019-07-03	500	350	150
Total	**670**	**465**	**205**

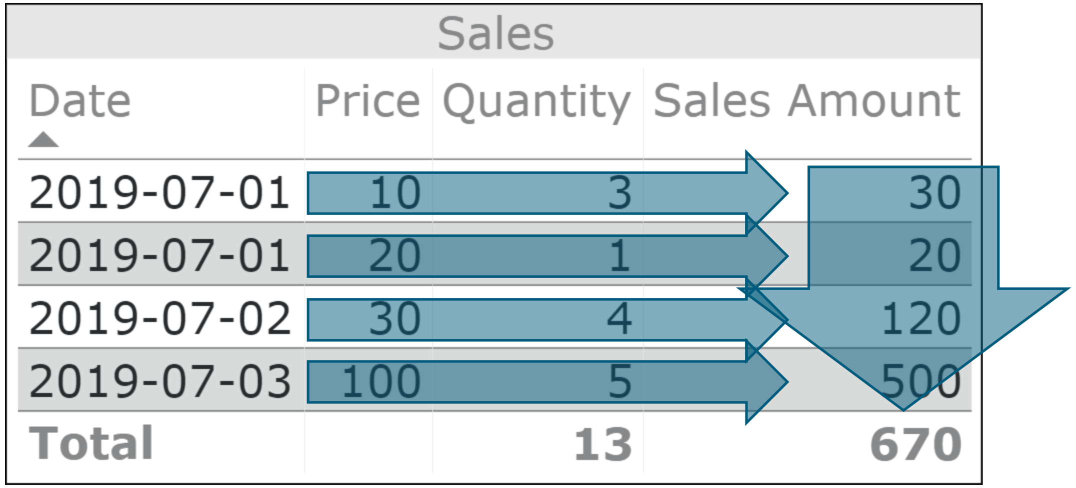
[Sales Amount]

```
:= SUM('Sales'[Price]) * SUM('Sales'[Quantity])
```



Date	Price	Quantity	Sales Amount
2019-07-01	30	4	120
2019-07-02	30	4	120
2019-07-03	100	5	500
Total	**160**	**13**	**2080**

[Sales Amount]



Sales			
Date	Price	Quantity	Sales Amount
2019-07-01	10	3	30
2019-07-01	20	1	20
2019-07-02	30	4	120
2019-07-03	100	5	500
Total		**13**	**670**

```
:=
SUMX (
     'Sales',
     'Sales'[Price] * 'Sales'[Quantity]
)
```

Sales Amount

$$Sales\ Amount = \sum (Price\ * Quantity)$$

$$\cancel{Sales\ Amount = \sum Price * \sum Quantity}$$

Margin

$$Margin = \sum (Sales\ Amount - Total\ Cost)$$

$$Margin = \sum Sales\ Amount - \sum Total\ Cost$$

Margin in %

$$Margin\ in\ \% = \frac{\sum Margin}{\sum Sales\ Amount}$$

$$\xcancel{Margin\ in\ \% = \sum \frac{Margin}{Sales\ Amount}}$$

Take-aways

- Take special care of all your numeric columns
  - Ordinary columns and calculated columns alike
- Change *Default Summarization* to *Don’t Summarize* for all numeric columns, where an aggregation does not make sense
  - e. g. Price, Calendar Year, Month Number, …
- Create a measure for all the others
  - With the correct aggregation
  - Hide the original column
- Don’t create (numeric) calculated columns at all
  - Use Power Query / M if necessary (eg. for certain visuals)

Next level

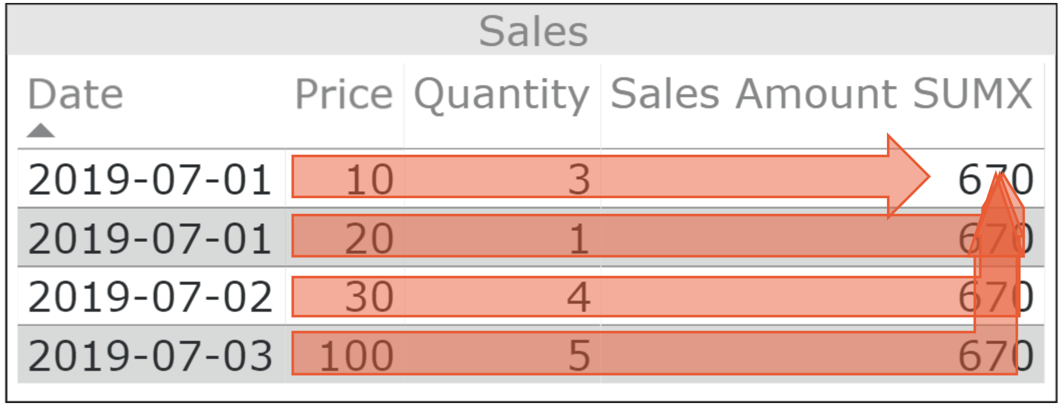
Using an iterator (eg. `SUMX`) in a calculated column

Using a measure inside a calculated column

'Sales'[Sales Amount SUMX]

The calculated column’s *row context* is **not filtering** the `'Sales'` table inside of SUMX.

```
=
SUMX (
    'Sales',
    'Sales'[Price] * 'Sales'[Quantity]
)
```

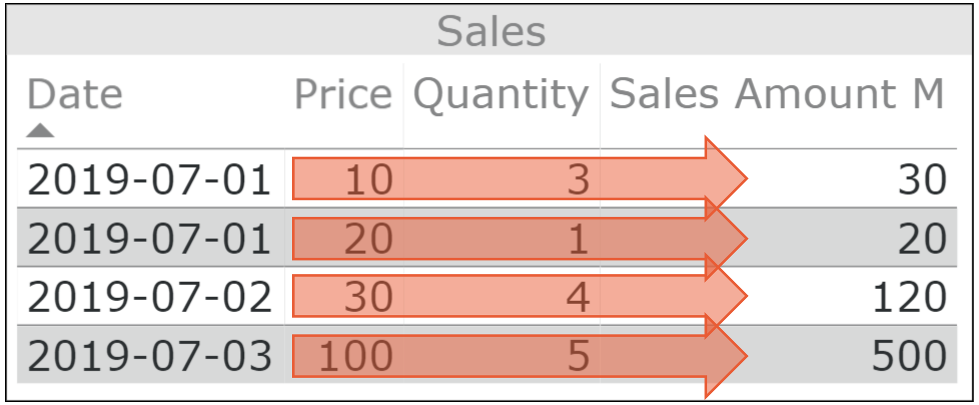


Sales

Date	Price	Quantity	Sales Amount SUMX
2019-07-01	10	3	670
2019-07-01	20	1	670
2019-07-02	30	4	670
2019-07-03	100	5	670

'Sales'[Sales Amount M]

Row context is (implicitly) transitioned into a *filter context* when a measure is referenced inside a *row context*.



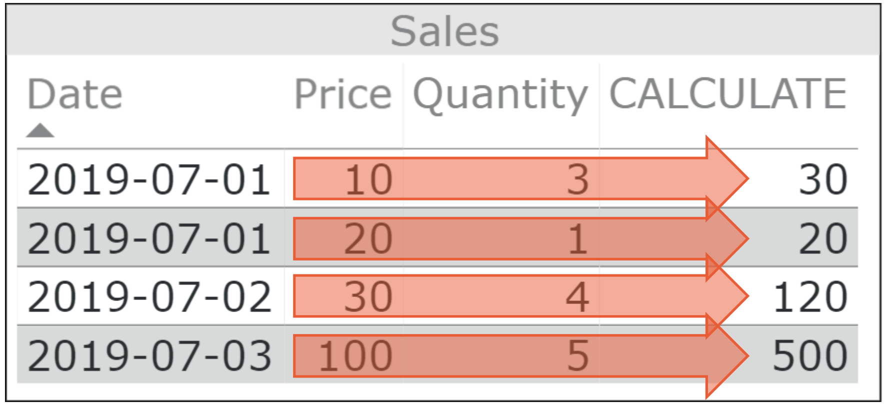
Sales			
Date	Price	Quantity	Sales Amount M
2019-07-01	10	3	30
2019-07-01	20	1	20
2019-07-02	30	4	120
2019-07-03	100	5	500

```
= [Sales Amount]
```

```
[Sales Amount] :=
SUMX (
    'Sales',
    'Sales'[Price] * 'Sales'[Quantity]
)
```

'Sales'[Sales Amount CALCULATE]

`CALCULATE()` performs a transition of a *row context* into a *filter context*.



Sales			
Date	Price	Quantity	CALCULATE
2019-07-01	10	3	30
2019-07-01	20	1	20
2019-07-02	30	4	120
2019-07-03	100	5	500

```
=
CALCULATE(
     SUMX (
         'Sales',
         'Sales'[Price] * 'Sales'[Quantity]
     )
)
```

Take-aways

Calculated columns can reference other columns (row-by-row)

Row context does not filter a table

CALCULATE() explicitly *transitions* a *row context* into a *filter context*

- Measures are implicitly wrapped in a CALCULATE()

Include table name when referencing columns (`'Table'[Column]`)

Omit table name when referencing measures (`[Measure]`)

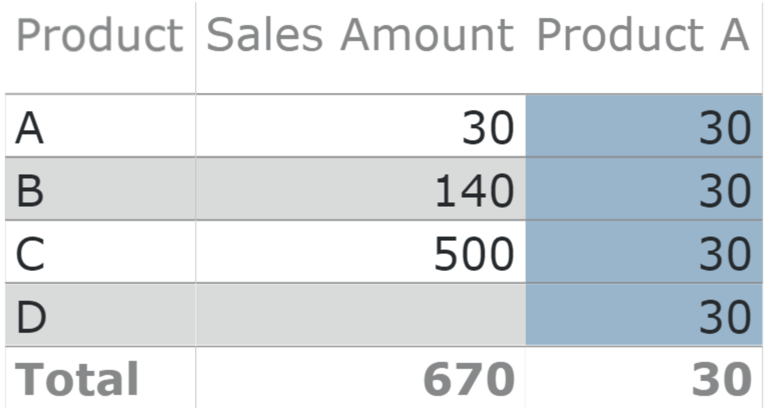
- Makes it clear that *context transition* might happen
- Makes it possible to move a measure to another table

Next level

Manipulate *filter context* with CALCULATE

[Product A]

CALCULATE() can filter the expression.

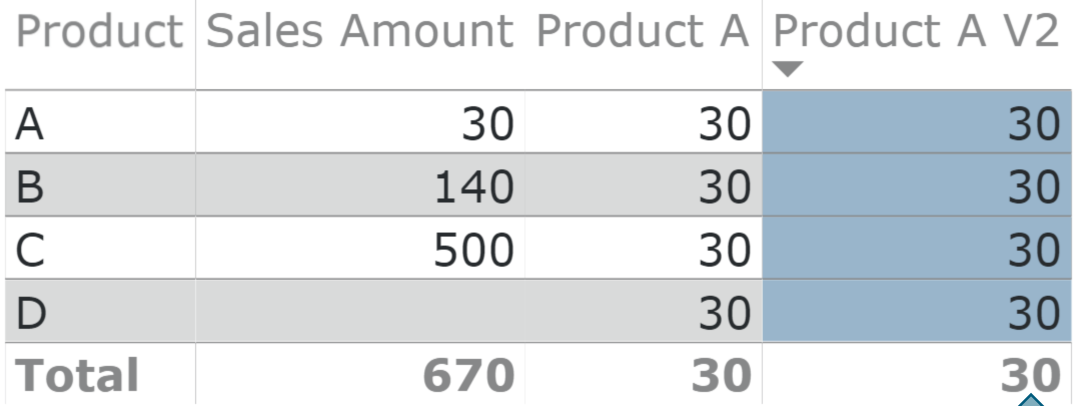


Product	Sales Amount	Product A
A	30	30
B	140	30
C	500	30
D		30
Total	**670**	**30**

```
:=
CALCULATE (
     [Sales Amount],
     'Product'[Product Desc] = "A"
)
```

[Product A V2]

`CALCULATE()` filters the expression via implicit
`FILTER(ALL('Table'[Column]), 'Table'[Column] =` *`<expr>`*`)`



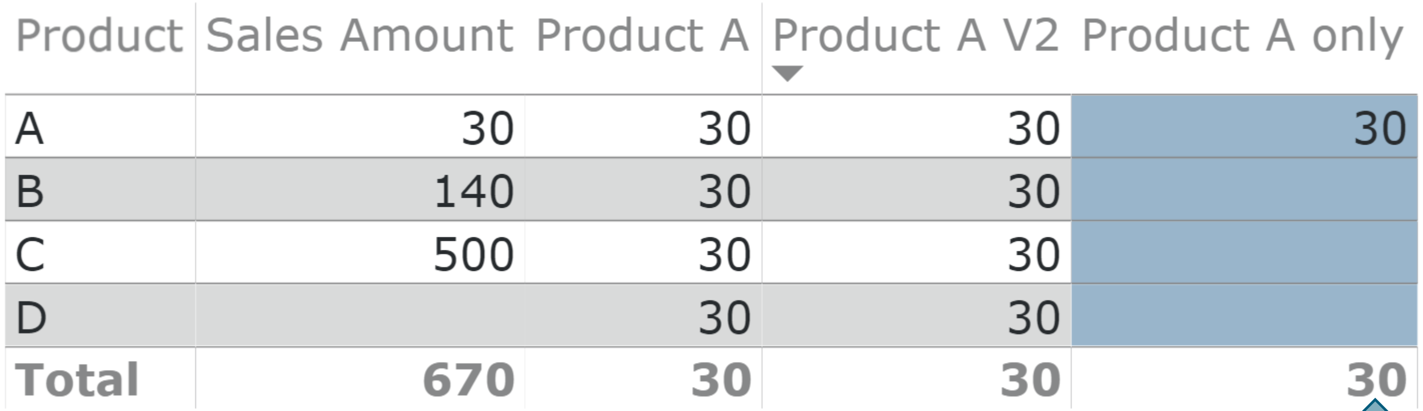
Product	Sales Amount	Product A	Product A V2
A	30	30	30
B	140	30	30
C	500	30	30
D		30	30
Total	**670**	**30**	**30**

```
:=
CALCULATE (
    [Sales Amount],
    FILTER (
        ALL('Product'[Product Desc]),
        'Product'[Product Desc] = "A"
    )
)
```

[Product A only]

Without `ALL()` filters combined via logical AND operator

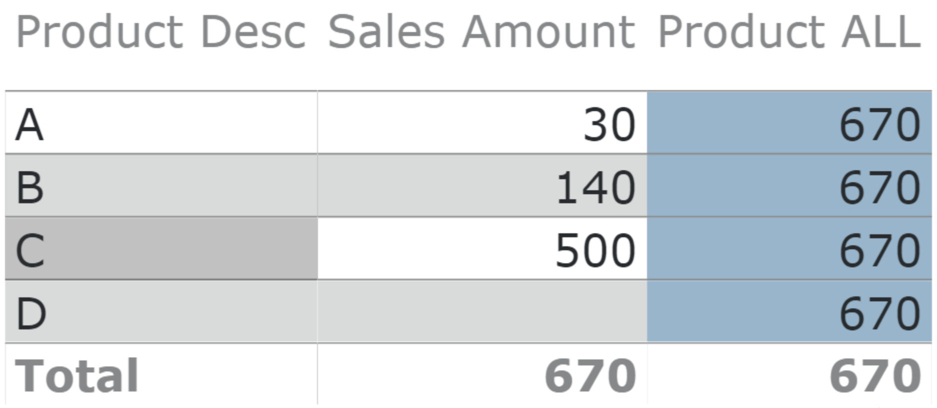
```
:=
CALCULATE (
    [Sales Amount],
    FILTER (
        'Product',
        'Product'[Product Desc] = "A"
    )
)
```



Product	Sales Amount	Product A	Product A V2	Product A only
A	30	30	30	30
B	140	30	30	
C	500	30	30	
D		30	30	
Total	**670**	**30**	**30**	**30**

[Product ALL]

`ALL()` alone removes filters

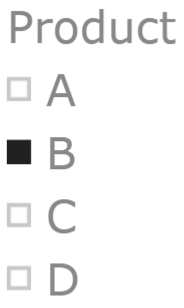


Product Desc	Sales Amount	Product ALL
A	30	670
B	140	670
C	500	670
D		670
Total	**670**	**670**

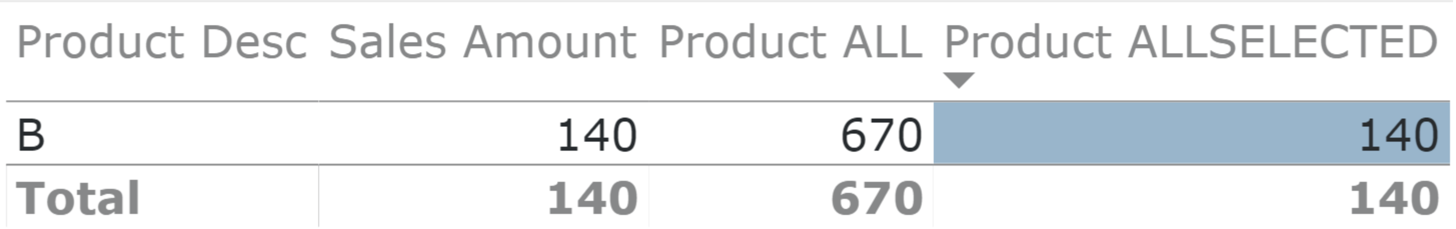
```
:=
CALCULATE (
     [Sales Amount],
     ALL('Product'[Product Desc])
)
```

[Product ALLSELECTED]

`ALLSELECTED()` alone removes filters from “within” the visual only


Product
□ A
■ B
□ C
□ D

```
:=
CALCULATE (
    [Sales Amount],
    ALLSELECTED('Product'[Product Desc])
)
```



Product Desc	Sales Amount	Product ALL	Product ALLSELECTED
B	140	670	140
Total	**140**	**670**	**140**

Take-aways

`CALCULATE()` can both, add and remove filters

Filtering in `CALCULATE()` overrides existing filter on same column

- As it is implemented as `FILTER(ALL('Table'[Column]), 'Table'[Column] = <expr>)`

`ALL()` in `CALCULATE()` removes filter from table or column

`ALLSELECTED() in CALCULATE()` removes filter from table or column, but keeps filter from outside the current visual

Wrap-up

DAX has easy syntax, but complex semantic

Don’t create (numeric) calculated columns

- Use Power Query / M if necessary (eg. for certain visuals)

Create a measure for all calculations

- With the correct aggregation
- Hide the original column

Include table name when referencing columns (`'Table'[Column]`)

Omit table name when referencing measures (`[Measure]`)

Questions?

Savory DATA

Microsoft CERTIFIED Solutions Expert

Business Intelligence

Data Platform

Microsoft CERTIFIED Trainer

Markus Ehrenmüller-Jensen

Founder & BI Architect

markus@savorydata.com

@MEhrenmueller

/markus-ehrenmueller

www.savorydata.com

Microsoft Data Platform

BI Developer

Database Developer

Database Admin

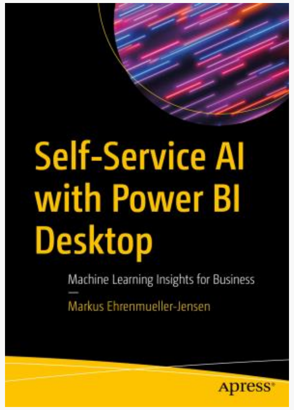
Microsoft MCT

MVP Microsoft® Most Valuable Professional

PowerBIUG POWER BI USER GROUP

PASS AUSTRIA

PASS SQLSATURDAY VIENNA



Self-Service AI with Power BI Desktop

Machine Learning Insights for Business

Markus Ehrenmueller-Jensen

Apress®