Manipulation with hierarchies in Power BI

„Make your data shine!“

2023

SPEAKER

LINKEDIN

TWITTER

ŠTĚPÁN REŠL

Data Brothers

Microsoft® Most Valuable Professional MVP

JAK NA POWER BI

Power BI kafíčko

Data Meerkat

b3

REBTECH

SQLSERVICE
A PART OF SYSCO

24

solarwinds

dbWatch

redgate

devart

Snabbfoting

AGENDA

- Type of hierarchies

- How visuals handles hierarchies

- Hierarchies and RLS

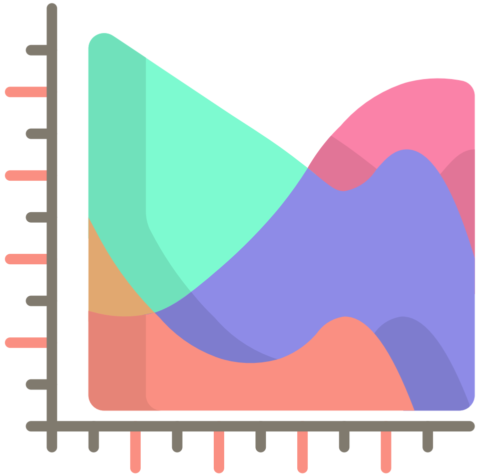
- Field parameter as a hierarchy

- Specific RLS + Hierarchy scenario

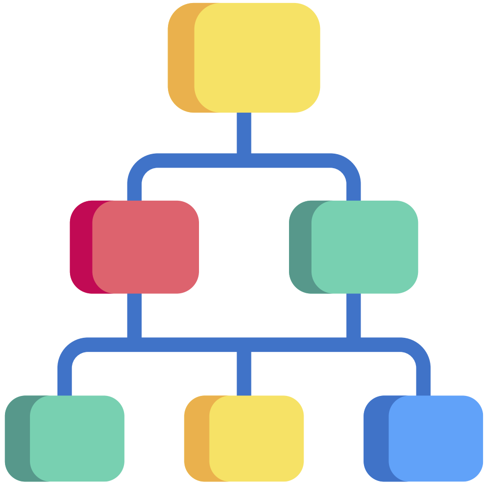
TYPE OF HIERARCHIES

Division by purpose

Different requirements call for different approaches



Hierarchy for visuals

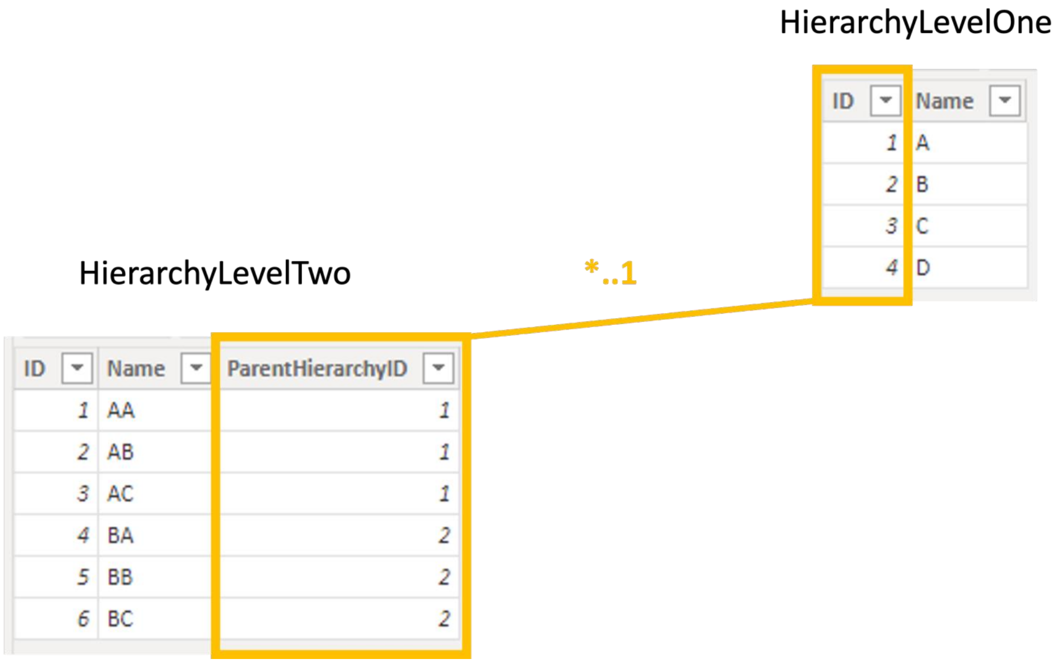


Hierarchy for RLS

For Visuals

Divided by count of tables

One table, two tables… should we care?

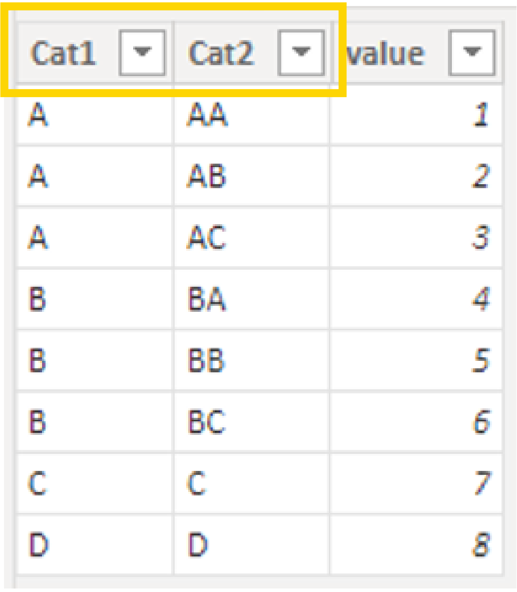


HierarchyLevelOne

ID	Name
1	A
2	B
3	C
4	D

HierarchyLevelTwo

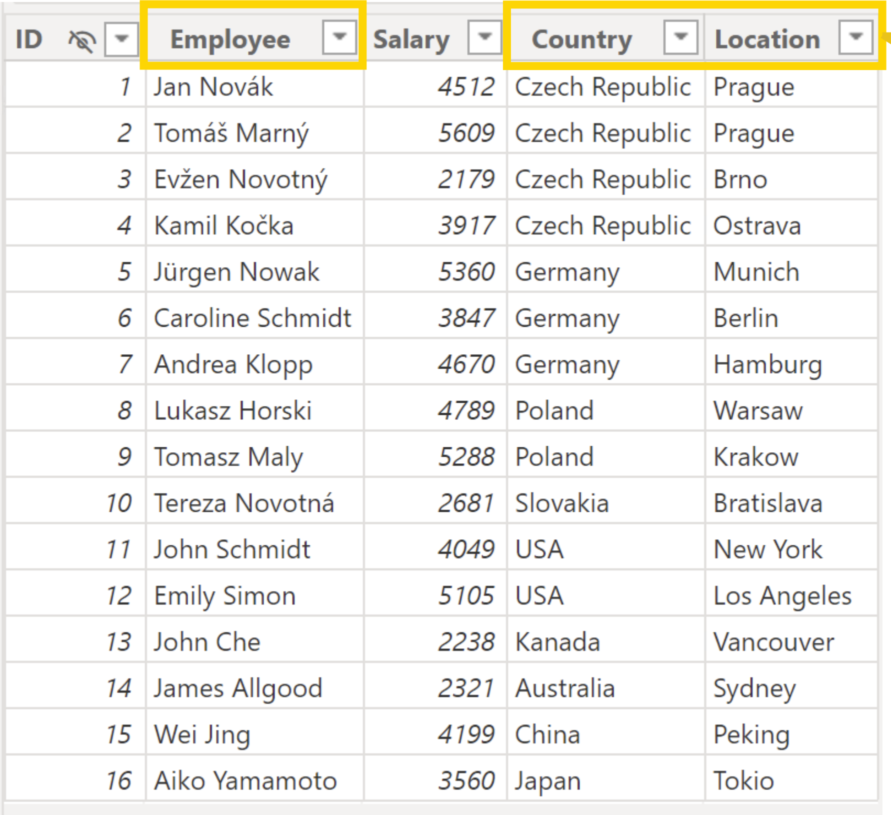
ID	Name	ParentHierarchyID
1	AA	1
2	AB	1
3	AC	1
4	BA	2
5	BB	2
6	BC	2



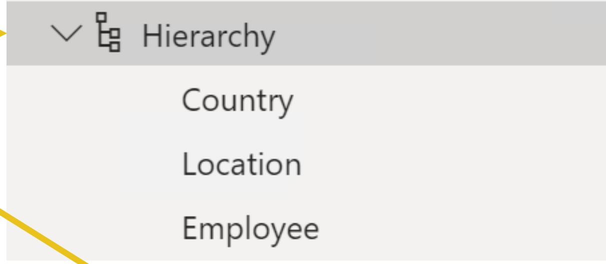
Cat1	Cat2	value
A	AA	1
A	AB	2
A	AC	3
B	BA	4
B	BB	5
B	BC	6
C	C	7
D	D	8

Hierarchy inside one table

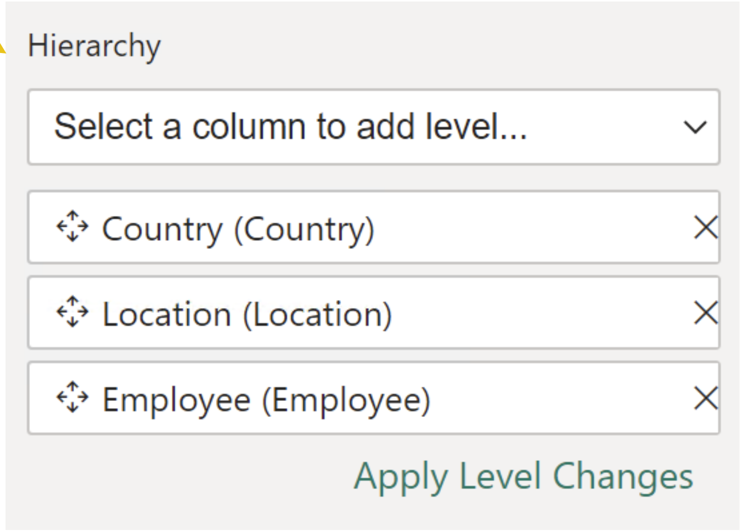
When everything is at one place



ID	Employee	Salary	Country	Location
1	Jan Novák	4512	Czech Republic	Prague
2	Tomáš Marný	5609	Czech Republic	Prague
3	Evžen Novotný	2179	Czech Republic	Brno
4	Kamil Kočka	3917	Czech Republic	Ostrava
5	Jürgen Nowak	5360	Germany	Munich
6	Caroline Schmidt	3847	Germany	Berlin
7	Andrea Klopp	4670	Germany	Hamburg
8	Lukasz Horski	4789	Poland	Warsaw
9	Tomasz Maly	5288	Poland	Krakow
10	Tereza Novotná	2681	Slovakia	Bratislava
11	John Schmidt	4049	USA	New York
12	Emily Simon	5105	USA	Los Angeles
13	John Che	2238	Kanada	Vancouver
14	James Allgood	2321	Australia	Sydney
15	Wei Jing	4199	China	Peking
16	Aiko Yamamoto	3560	Japan	Tokio



Hierarchy
- Country
- Location
- Employee



Hierarchy

Select a column to add level...

Country (Country)

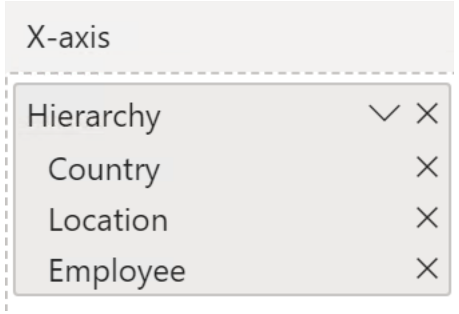
Location (Location)

Employee (Employee)

Apply Level Changes

It is easily applicable

Quick and easy to use



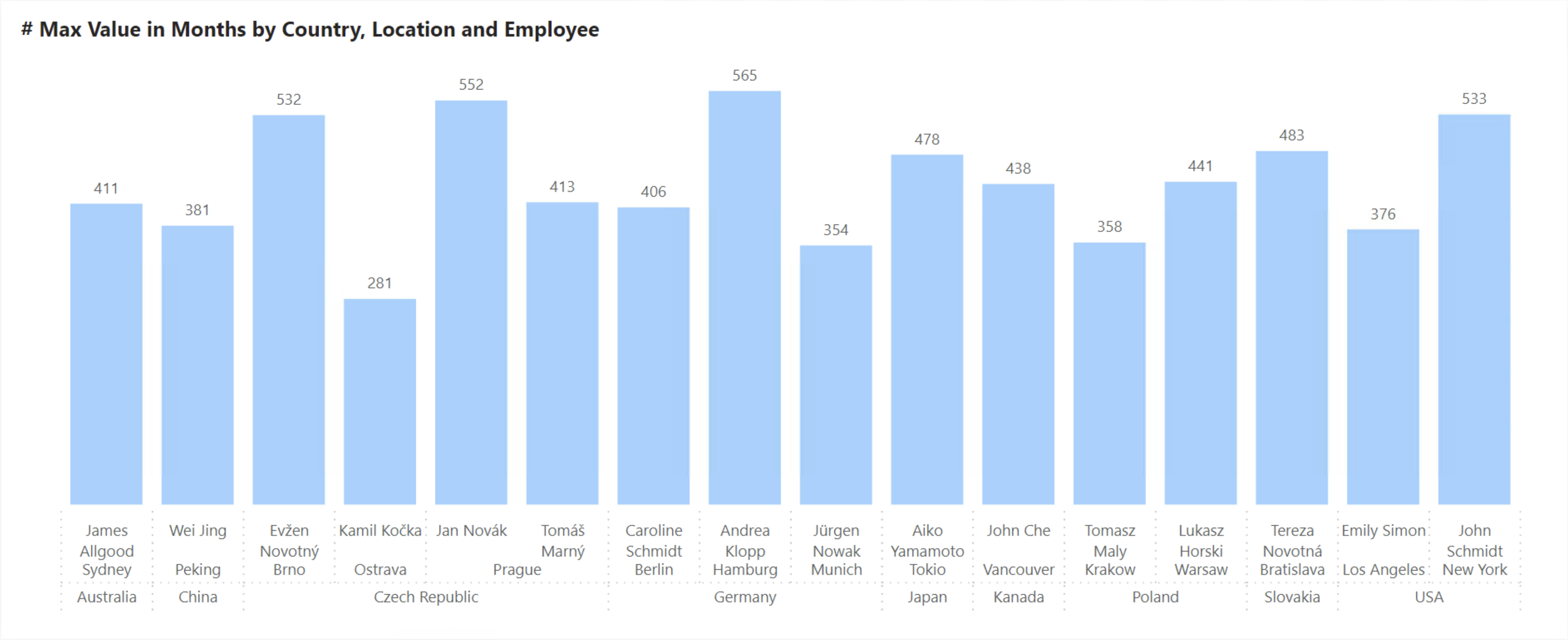
X-axis

Hierarchy

Country

Location

Employee

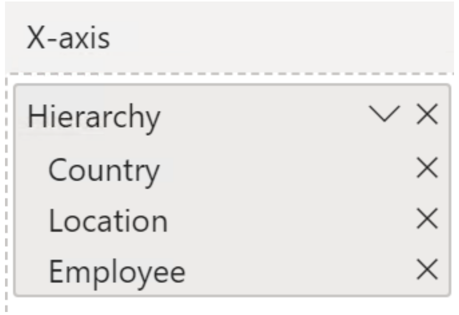


Max Value in Months by Country, Location and Employee

Is it supported in Custom visuals?

Not in many…

X*


X-axis

Hierarchy

Country

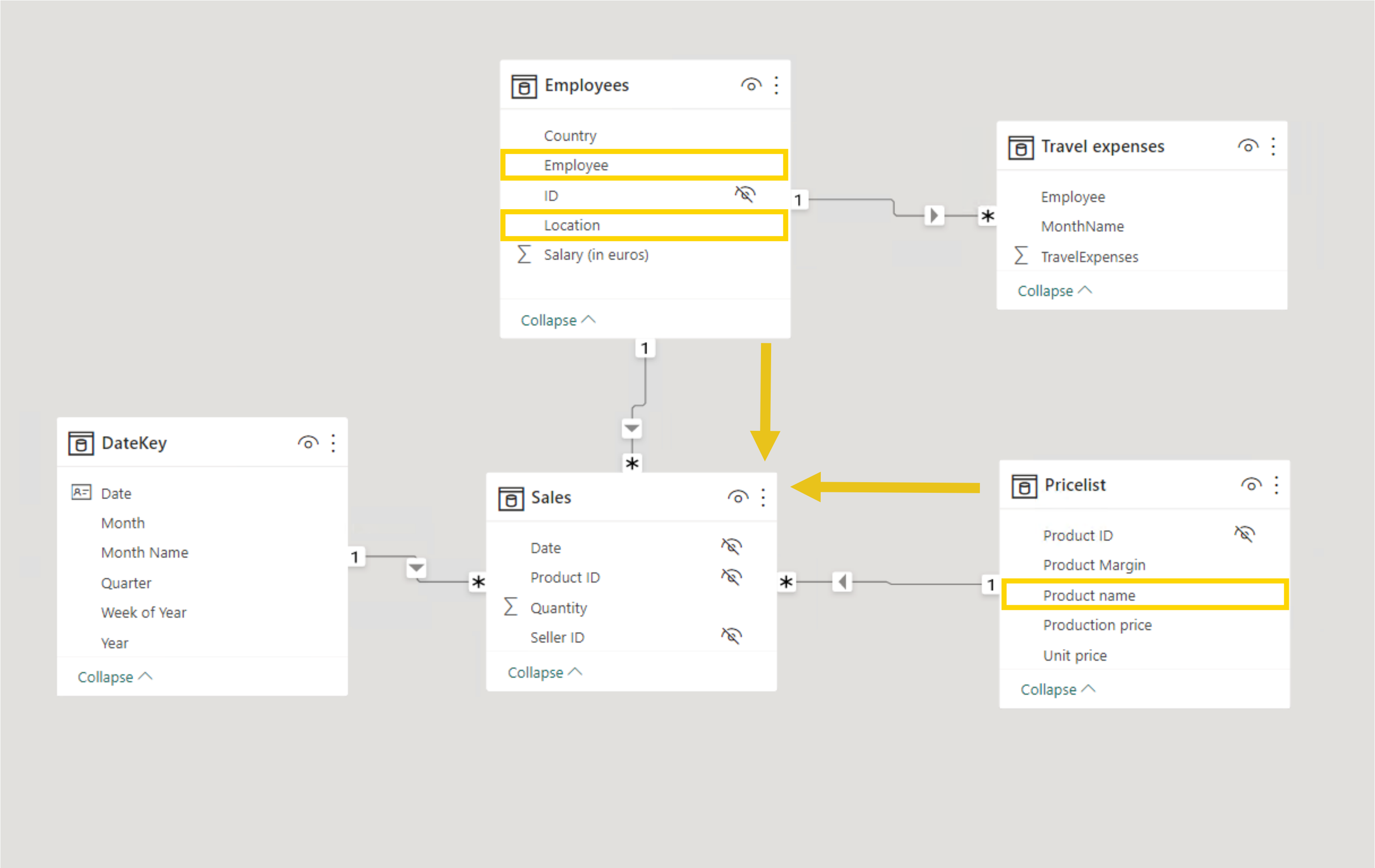
Location

Employee

= ID not NAMEOF()

Between tables hierarchies

Visuals can handle even data without a natural hierarchy.



Employees
Country
Employee
ID
Location
∑ Salary (in euros)
Collapse

Travel expenses
Employee
MonthName
∑ TravelExpenses
Collapse

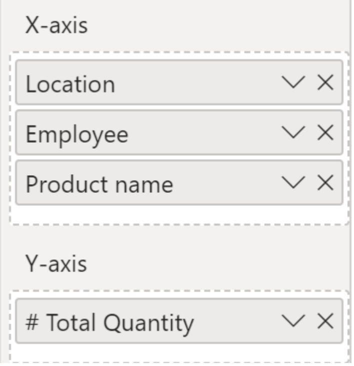
DateKey
Date
Month
Month Name
Quarter
Week of Year
Year
Collapse

Sales
Date
Product ID
∑ Quantity
Seller ID
Collapse

Pricelist
Product ID
Product Margin
Product name
Production price
Unit price
Collapse

Between tables hierarchies

Visuals can handle even data without a natural hierarchy.



X-axis

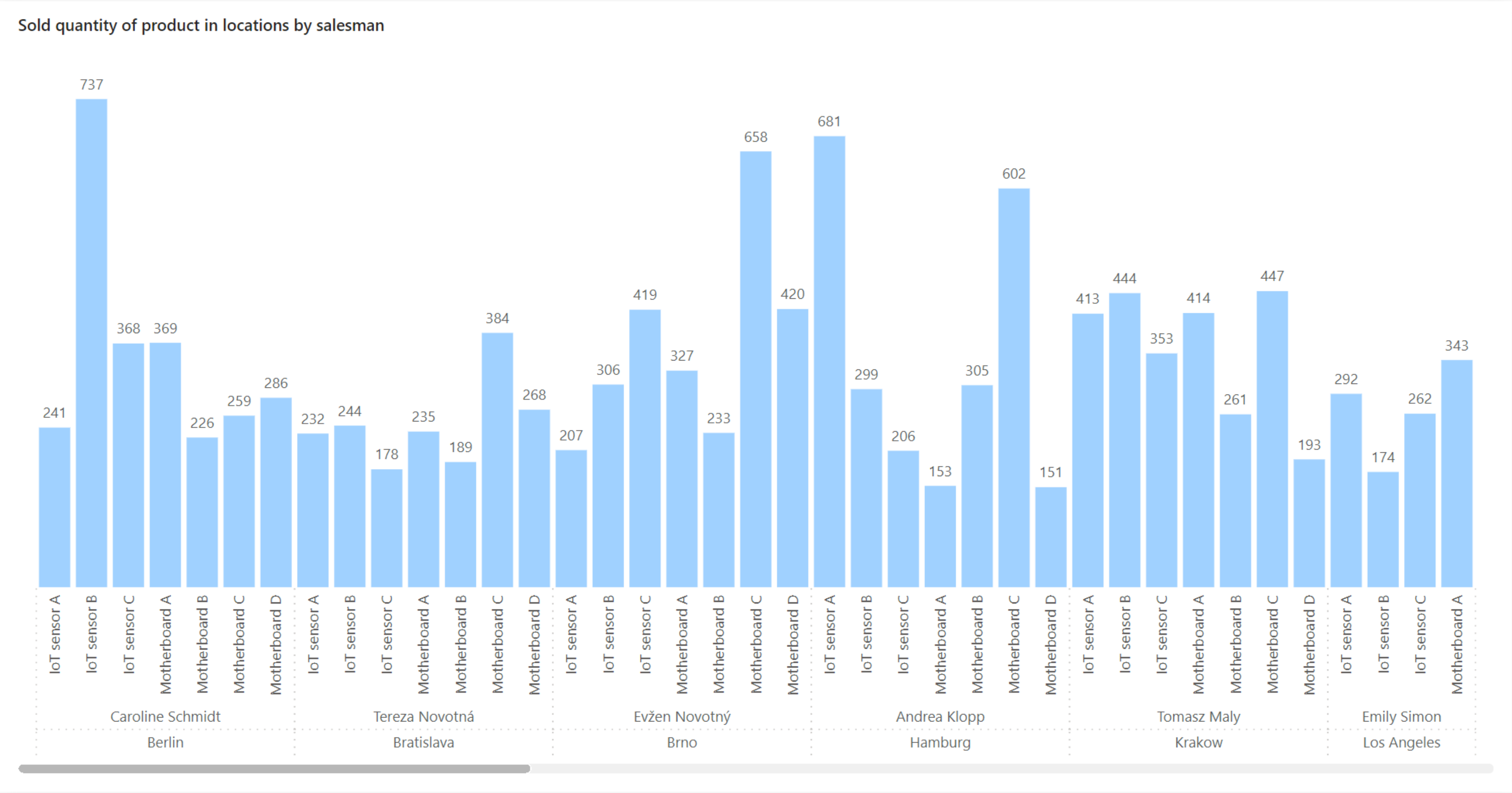
Location

Employee

Product name

Y-axis

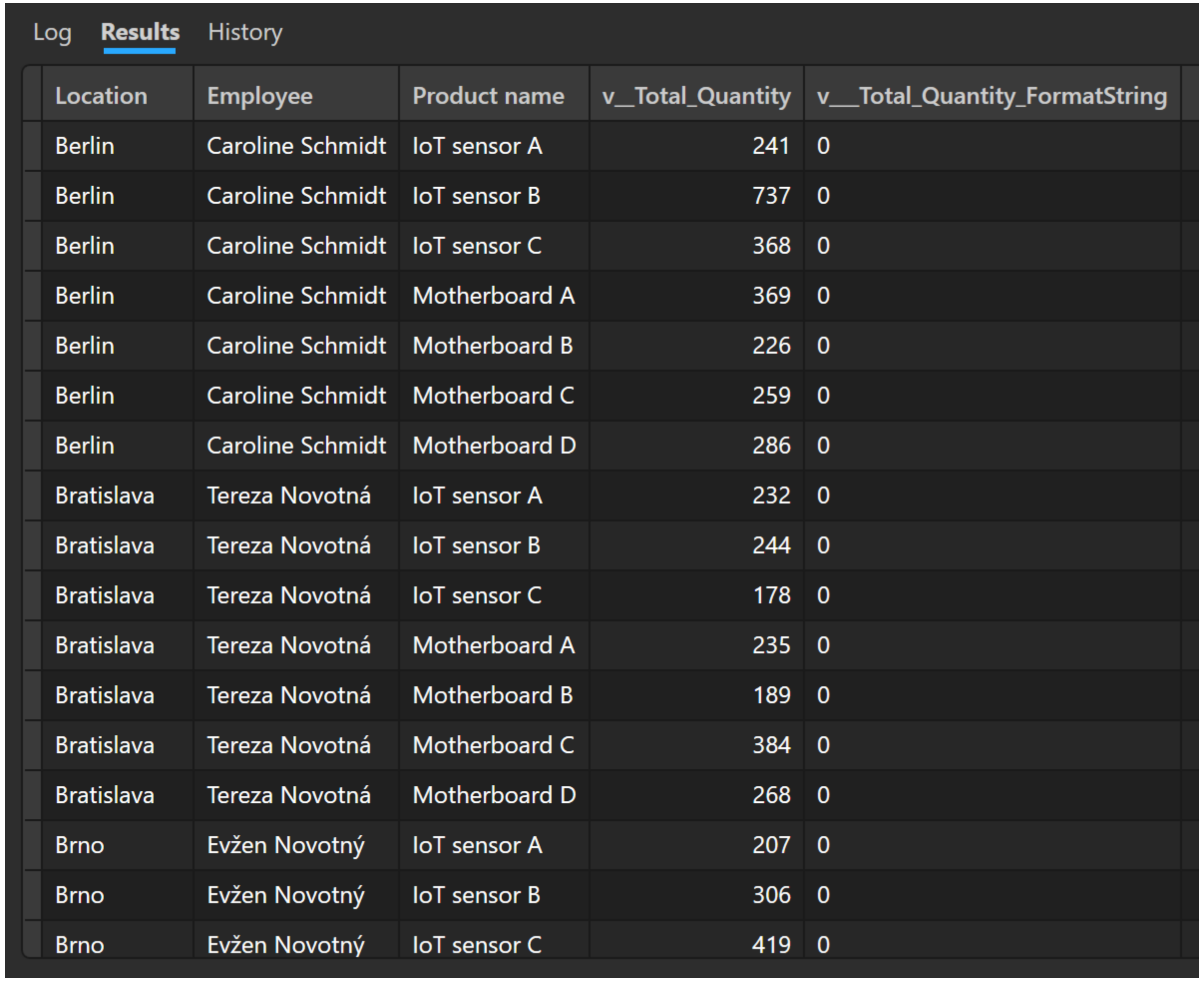
Total Quantity



Sold quantity of product in locations by salesman

Data used for creating our chart

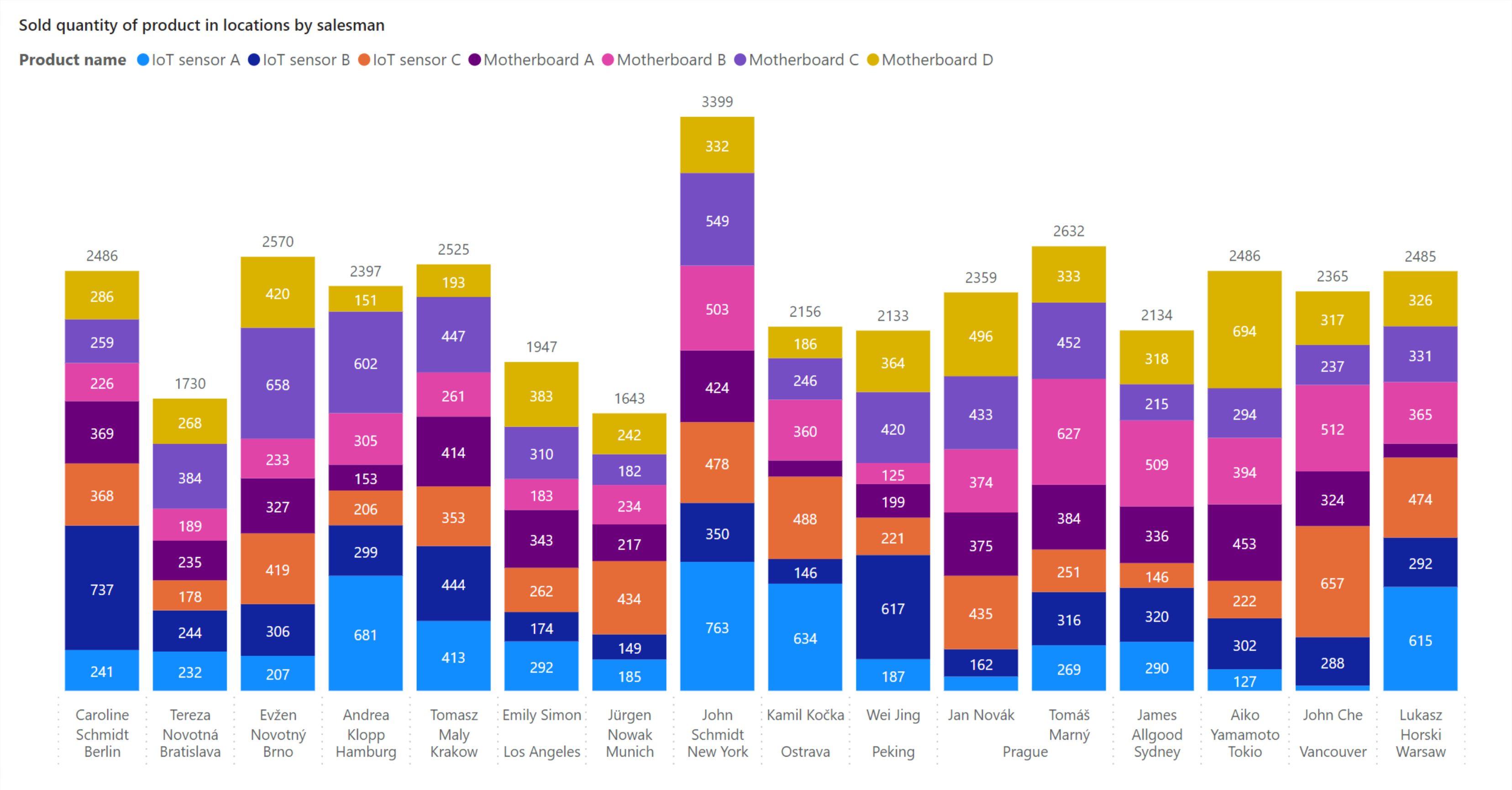
They seem standard, but will it look like this in every visual? NO!!



Location	Employee	Product name	v__Total_Quantity	v__Total_Quantity_FormatString
Berlin	Caroline Schmidt	IoT sensor A	241	0
Berlin	Caroline Schmidt	IoT sensor B	737	0
Berlin	Caroline Schmidt	IoT sensor C	368	0
Berlin	Caroline Schmidt	Motherboard A	369	0
Berlin	Caroline Schmidt	Motherboard B	226	0
Berlin	Caroline Schmidt	Motherboard C	259	0
Berlin	Caroline Schmidt	Motherboard D	286	0
Bratislava	Tereza Novotná	IoT sensor A	232	0
Bratislava	Tereza Novotná	IoT sensor B	244	0
Bratislava	Tereza Novotná	IoT sensor C	178	0
Bratislava	Tereza Novotná	Motherboard A	235	0
Bratislava	Tereza Novotná	Motherboard B	189	0
Bratislava	Tereza Novotná	Motherboard C	384	0
Bratislava	Tereza Novotná	Motherboard D	268	0
Brno	Evžen Novotný	IoT sensor A	207	0
Brno	Evžen Novotný	IoT sensor B	306	0
Brno	Evžen Novotný	IoT sensor C	419	0

Just a little change…

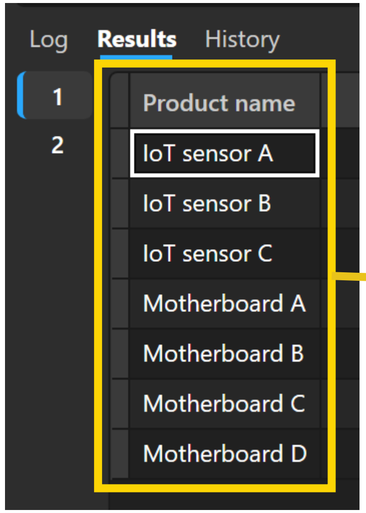
And even data needs to look different!



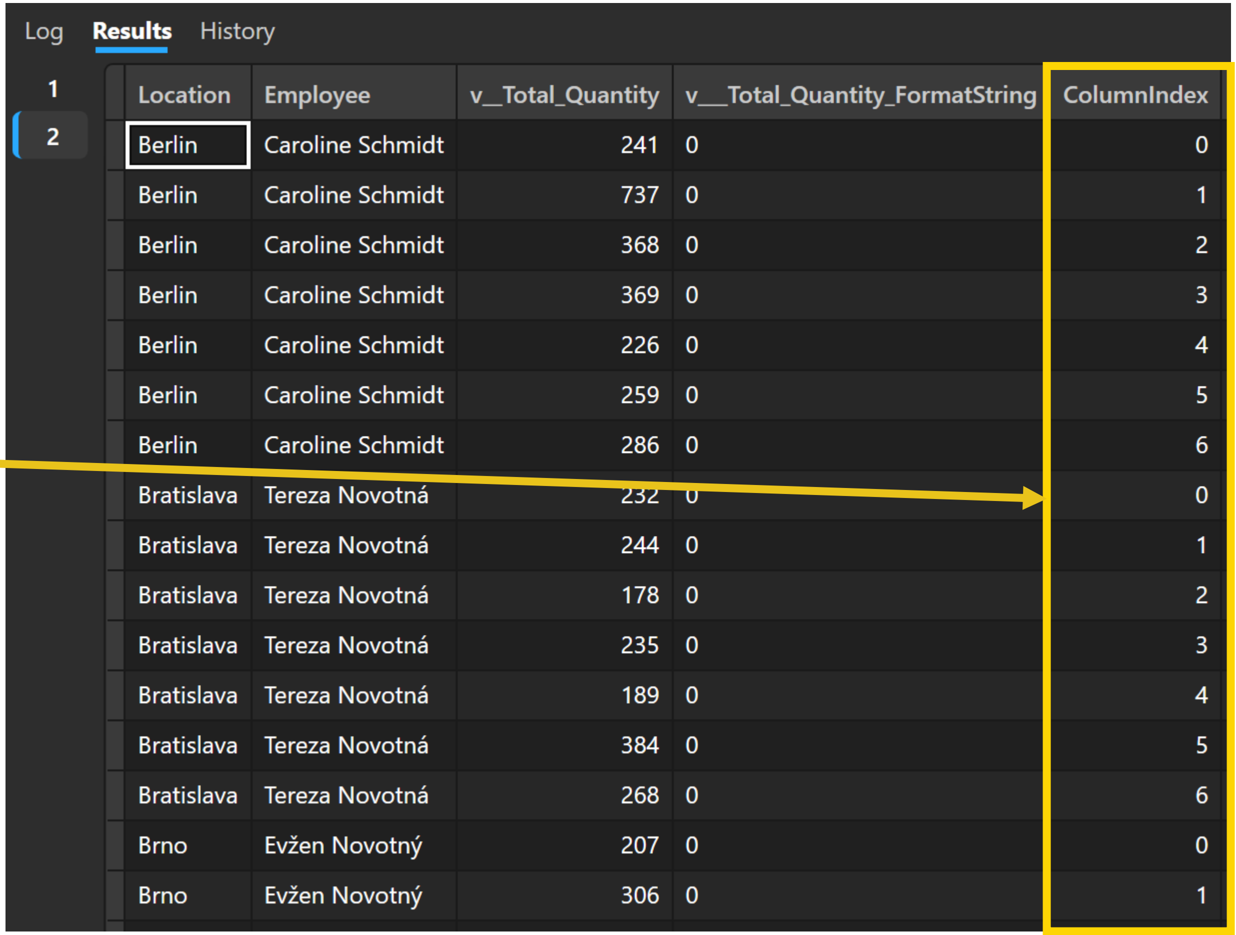
Sold quantity of product in locations by salesman

ColumnIndex appeared

This would also happen with data from just ONE TABLE



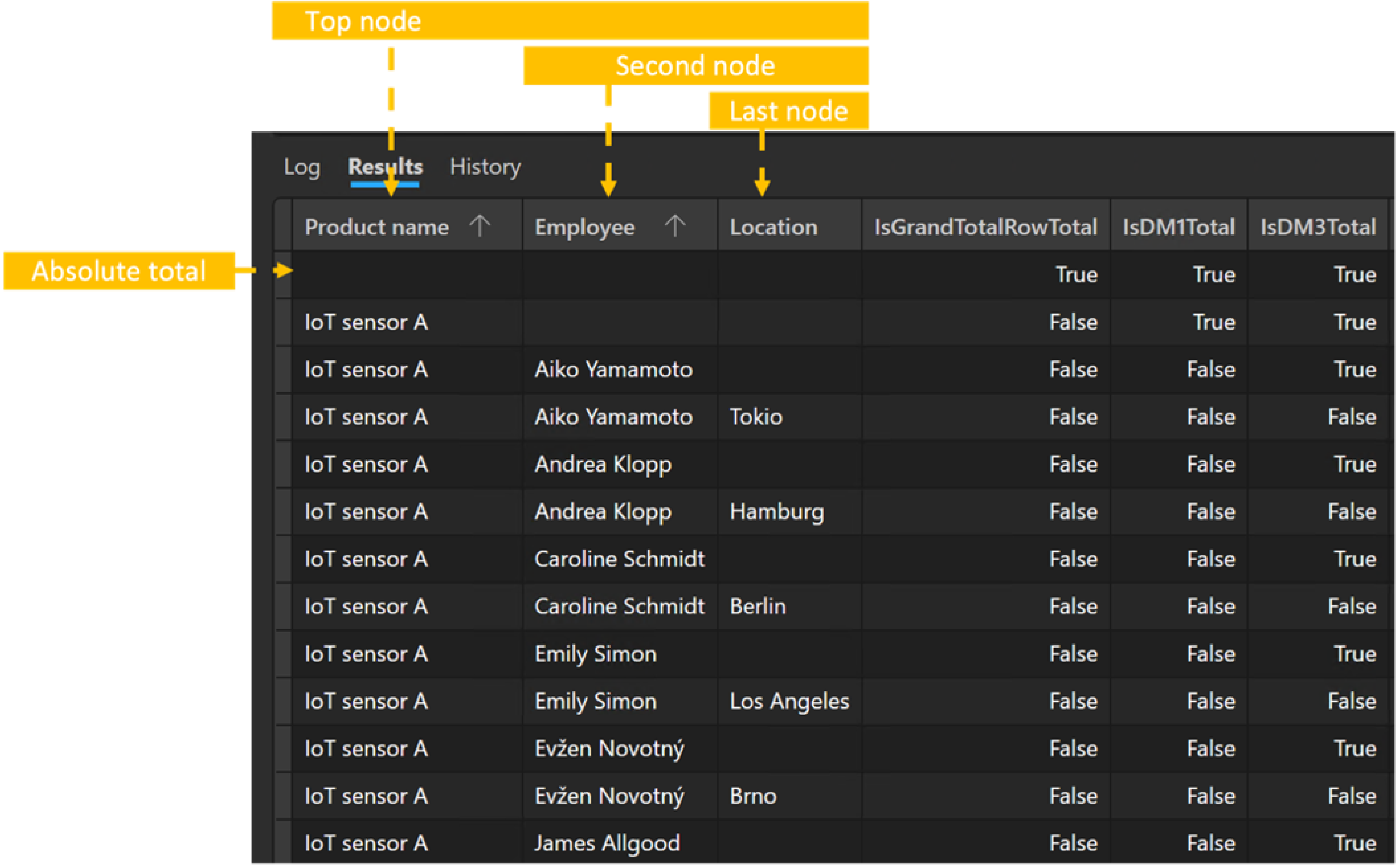
Product name
IoT sensor A
IoT sensor B
IoT sensor C
Motherboard A
Motherboard B
Motherboard C
Motherboard D



Location	Employee	v_Total_Quantity	v__Total_Quantity_FormatString	ColumnIndex
Berlin	Caroline Schmidt	241	0	0
Berlin	Caroline Schmidt	737	0	1
Berlin	Caroline Schmidt	368	0	2
Berlin	Caroline Schmidt	369	0	3
Berlin	Caroline Schmidt	226	0	4
Berlin	Caroline Schmidt	259	0	5
Berlin	Caroline Schmidt	286	0	6
Bratislava	Tereza Novotná	232	0	0
Bratislava	Tereza Novotná	244	0	1
Bratislava	Tereza Novotná	178	0	2
Bratislava	Tereza Novotná	235	0	3
Bratislava	Tereza Novotná	189	0	4
Bratislava	Tereza Novotná	384	0	5
Bratislava	Tereza Novotná	268	0	6
Brno	Evžen Novotný	207	0	0
Brno	Evžen Novotný	306	0	1

Visuals using node system

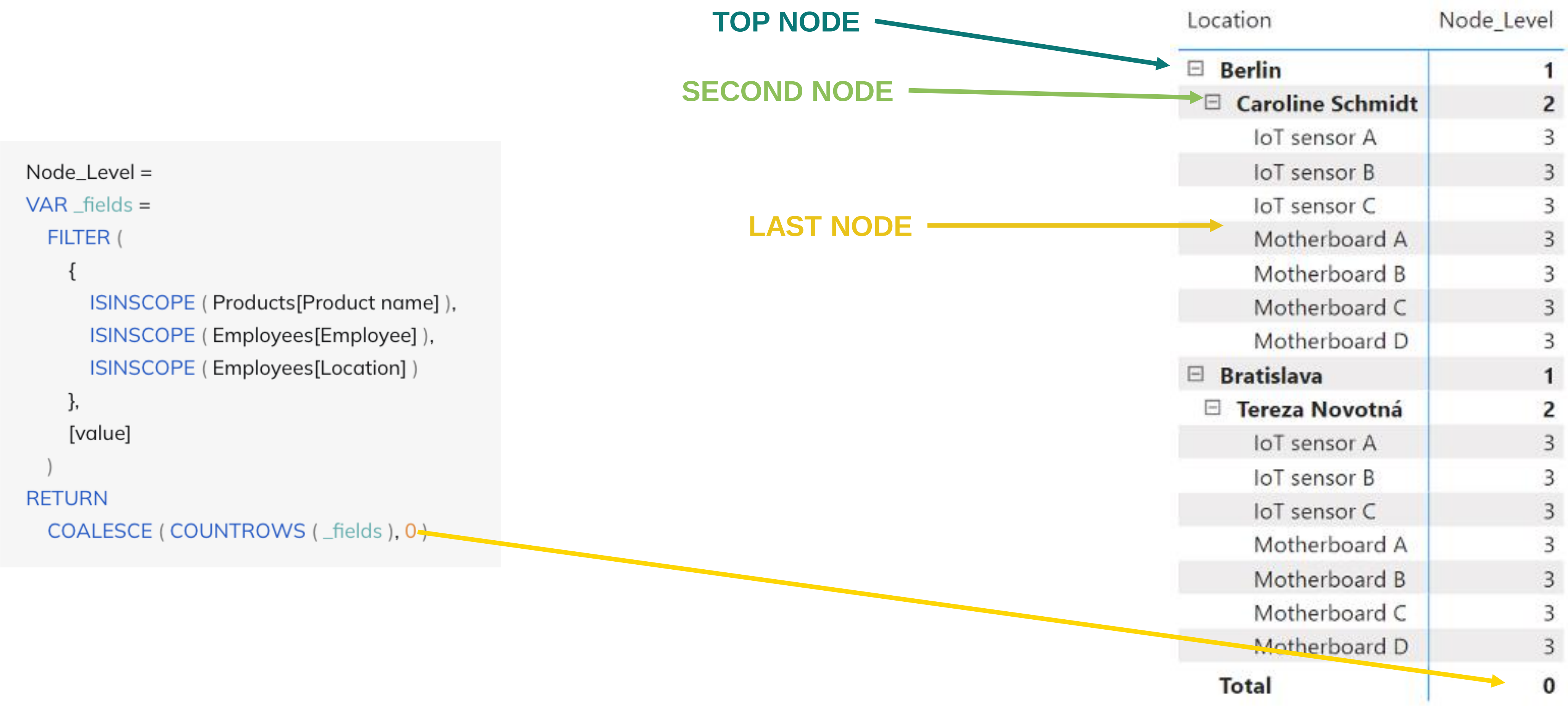
The dimensional node system can be beneficial but needs to be understood. Some of them even use total additions.



Product name	Employee	Location	IsGrandTotalRowTotal	IsDM1Total	IsDM3Total
			True	True	True
IoT sensor A			False	True	True
IoT sensor A	Aiko Yamamoto		False	False	True
IoT sensor A	Aiko Yamamoto	Tokio	False	False	False
IoT sensor A	Andrea Klopp		False	False	True
IoT sensor A	Andrea Klopp	Hamburg	False	False	False
IoT sensor A	Caroline Schmidt		False	False	True
IoT sensor A	Caroline Schmidt	Berlin	False	False	False
IoT sensor A	Emily Simon		False	False	True
IoT sensor A	Emily Simon	Los Angeles	False	False	False
IoT sensor A	Evžen Novotný		False	False	True
IoT sensor A	Evžen Novotný	Brno	False	False	False
IoT sensor A	James Allgood		False	False	True

ISINSCOPE helps to navigate

So we can do different types of calculations at different levels



TOP NODE

SECOND NODE

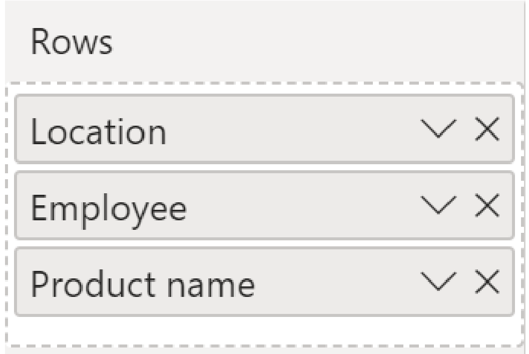
LAST NODE

```
Node_Level =
VAR _fields =
    FILTER (
        {
            ISINSCOPE ( Products[Product name] ),
            ISINSCOPE ( Employees[Employee] ),
            ISINSCOPE ( Employees[Location] )
        },
        [value]
    )
RETURN
    COALESCE ( COUNTROWS ( _fields ), 0 )
```

Location	Node_Level
⊟ **Berlin**	**1**
⊟ **Caroline Schmidt**	**2**
IoT sensor A	3
IoT sensor B	3
IoT sensor C	3
Motherboard A	3
Motherboard B	3
Motherboard C	3
Motherboard D	3
⊟ **Bratislava**	**1**
⊟ **Tereza Novotná**	**2**
IoT sensor A	3
IoT sensor B	3
IoT sensor C	3
Motherboard A	3
Motherboard B	3
Motherboard C	3
Motherboard D	3
Total	**0**

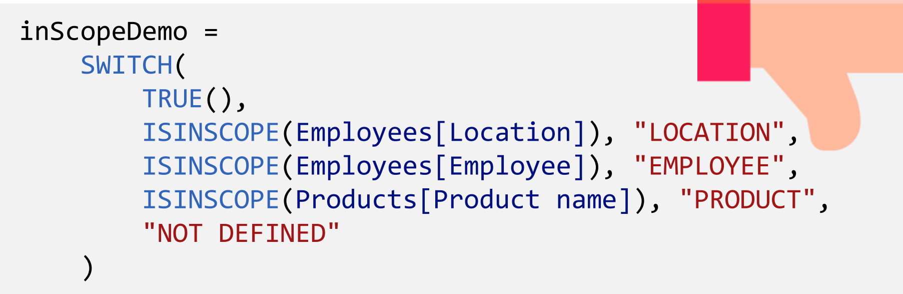
Don’t forget about the falling rule

Switch tries to approve one rule after another, so if you start with TOP NODE…

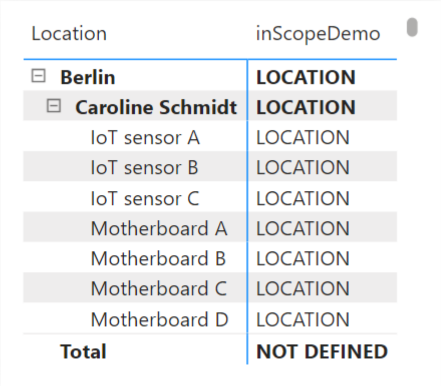


Rows

- Location
- Employee
- Product name



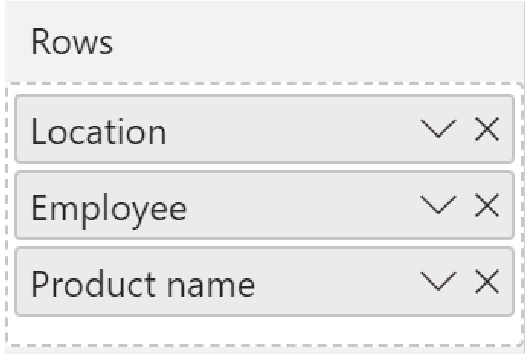
```
inScopeDemo =
    SWITCH(
        TRUE(),
        ISINSCOPE(Employees[Location]), "LOCATION",
        ISINSCOPE(Employees[Employee]), "EMPLOYEE",
        ISINSCOPE(Products[Product name]), "PRODUCT",
        "NOT DEFINED"
    )
```



Location	inScopeDemo
⊟ **Berlin**	**LOCATION**
⊟ **Caroline Schmidt**	**LOCATION**
IoT sensor A	LOCATION
IoT sensor B	LOCATION
IoT sensor C	LOCATION
Motherboard A	LOCATION
Motherboard B	LOCATION
Motherboard C	LOCATION
Motherboard D	LOCATION
Total	**NOT DEFINED**

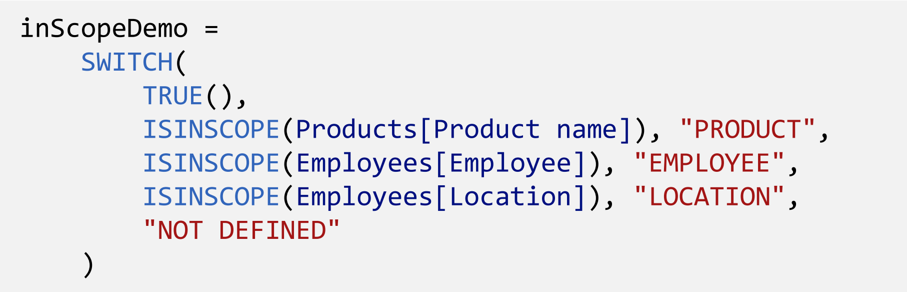
Start from the END

The most far node is where we should start

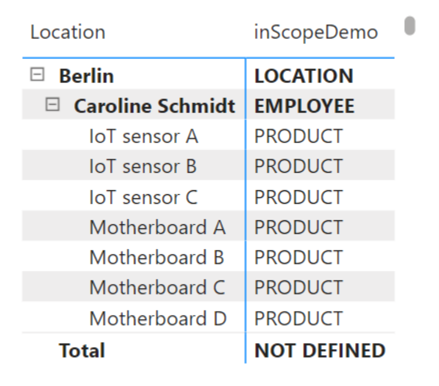


Rows

- Location
- Employee
- Product name



```
inScopeDemo =
    SWITCH(
        TRUE(),
        ISINSCOPE(Products[Product name]), "PRODUCT",
        ISINSCOPE(Employees[Employee]), "EMPLOYEE",
        ISINSCOPE(Employees[Location]), "LOCATION",
        "NOT DEFINED"
    )
```



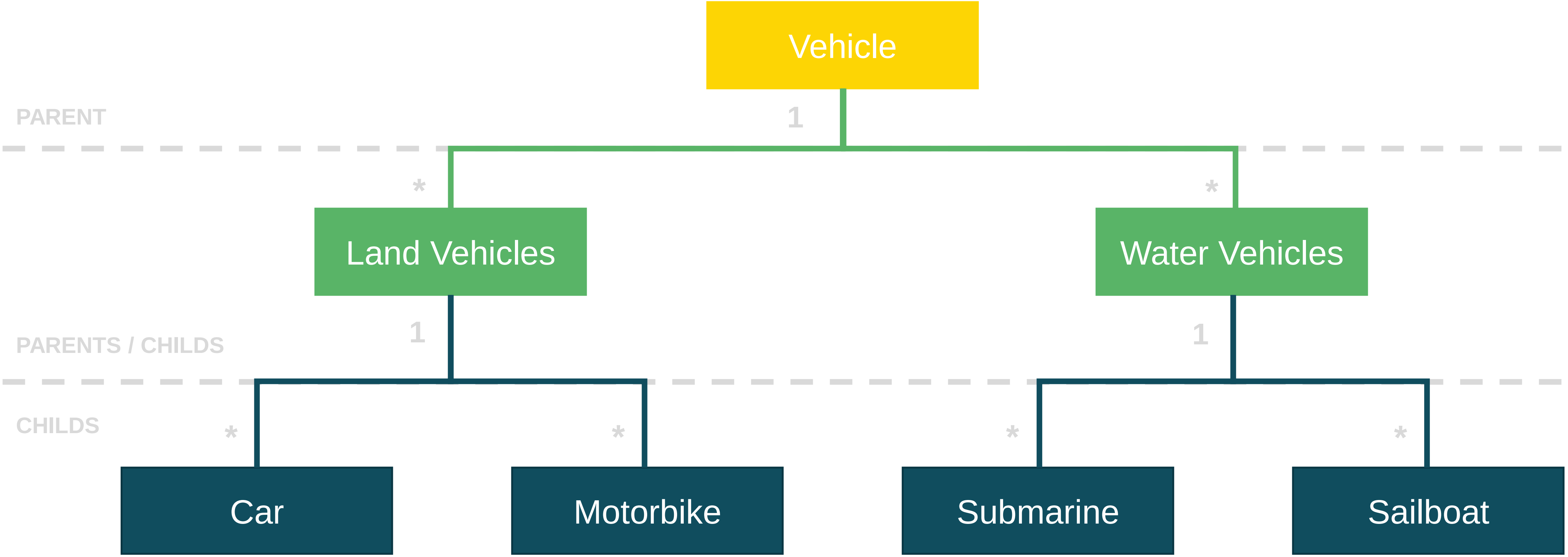
Location	inScopeDemo
⊟ **Berlin**	**LOCATION**
⊟ **Caroline Schmidt**	**EMPLOYEE**
IoT sensor A	PRODUCT
IoT sensor B	PRODUCT
IoT sensor C	PRODUCT
Motherboard A	PRODUCT
Motherboard B	PRODUCT
Motherboard C	PRODUCT
Motherboard D	PRODUCT
Total	**NOT DEFINED**

For RLS

(ROW LEVEL SECURITY)

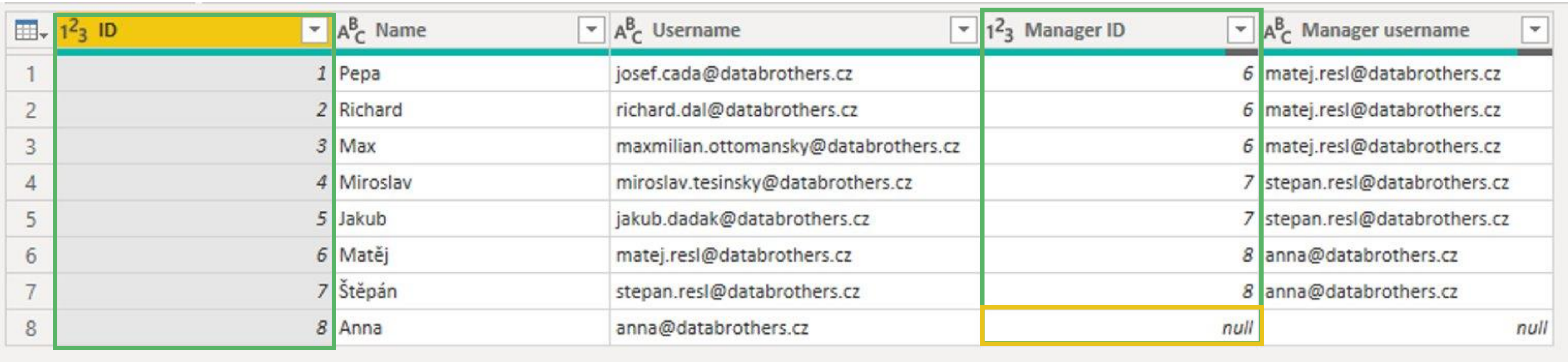
Typical Parent-Child hierarchy

Every child have exactly one parent.



Whats need to be remembered

A blank string is not the same as a Null.

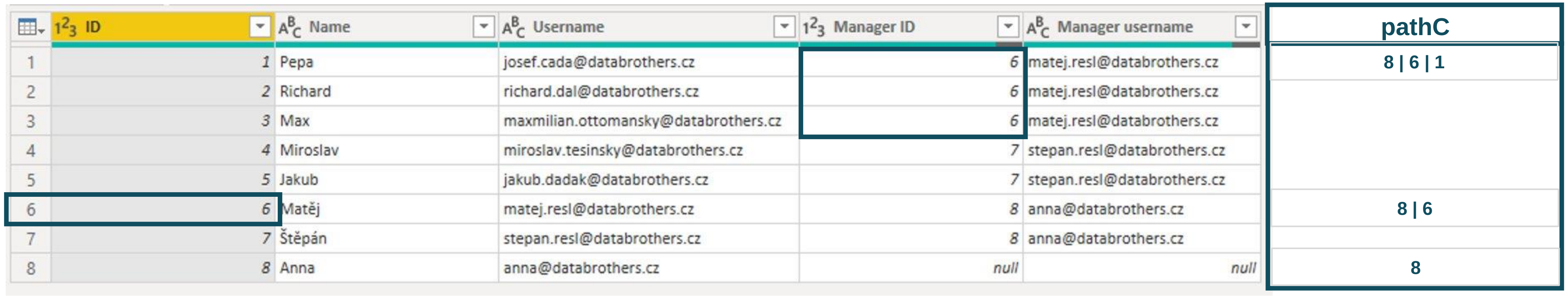


	ID	Name	Username	Manager ID	Manager username
1	1	Pepa	josef.cada@databrothers.cz	6	matej.resl@databrothers.cz
2	2	Richard	richard.dal@databrothers.cz	6	matej.resl@databrothers.cz
3	3	Max	maxmilian.ottomansky@databrothers.cz	6	matej.resl@databrothers.cz
4	4	Miroslav	miroslav.tesinsky@databrothers.cz	7	stepan.resl@databrothers.cz
5	5	Jakub	jakub.dadak@databrothers.cz	7	stepan.resl@databrothers.cz
6	6	Matěj	matej.resl@databrothers.cz	8	anna@databrothers.cz
7	7	Štěpán	stepan.resl@databrothers.cz	8	anna@databrothers.cz
8	8	Anna	anna@databrothers.cz	null	null

DAX for easy Parent-Child

If we have parents and a child prepared, it's a piece of cake!

pathC := PATH([ID], [Manager ID])

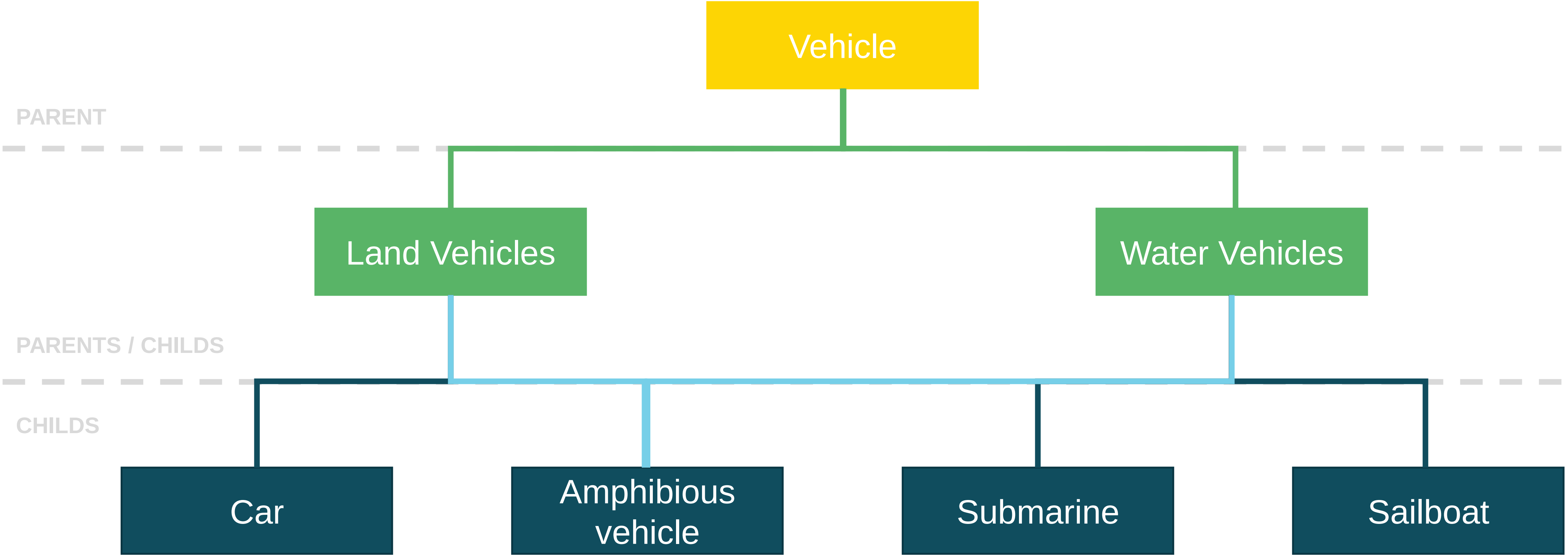


	ID	Name	Username	Manager ID	Manager username	pathC
1	1	Pepa	josef.cada@databrothers.cz	6	matej.resl@databrothers.cz	8 | 6 | 1
2	2	Richard	richard.dal@databrothers.cz	6	matej.resl@databrothers.cz	
3	3	Max	maxmilian.ottomansky@databrothers.cz	6	matej.resl@databrothers.cz	
4	4	Miroslav	miroslav.tesinsky@databrothers.cz	7	stepan.resl@databrothers.cz	
5	5	Jakub	jakub.dadak@databrothers.cz	7	stepan.resl@databrothers.cz	
6	6	Matěj	matej.resl@databrothers.cz	8	anna@databrothers.cz	8 | 6
7	7	Štěpán	stepan.resl@databrothers.cz	8	anna@databrothers.cz	
8	8	Anna	anna@databrothers.cz	null	null	8

PATHCONTAINS(<path> /* [pathC] */, <value> /* 6 */)

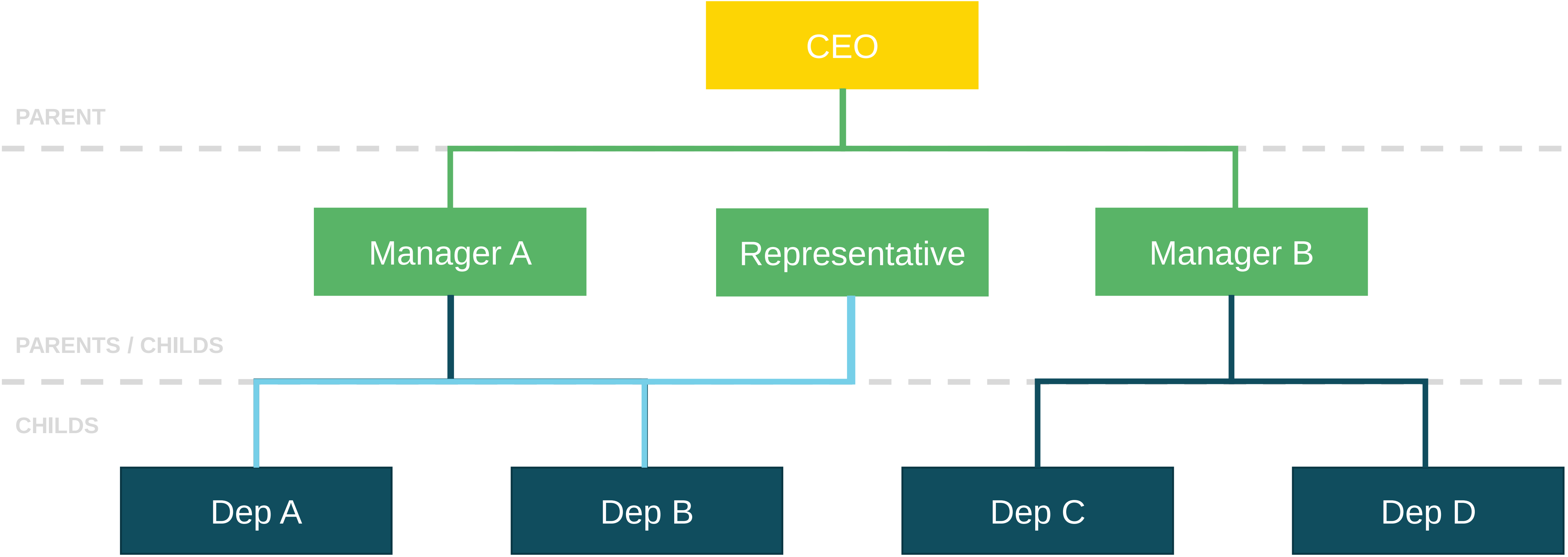
Multiple inheritance hierarchy

But what if the child has more parents



Let's look at that closer

But we will not be speaking about vehicles



Can we do it Power BI?

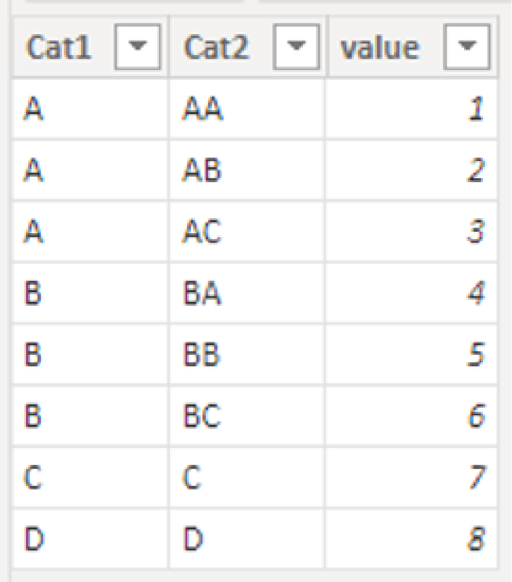
Of course, we can! But supporting it in RLS isn’t that easy.

- To be able to handle this, we need to consider a few main perspectives:
  - Creating hierarchy inside just **ONE** table will lead to multiplying primary IDs
    - Because of that, we will need to handle **M:N** or **M:1** relationships. That isn’t good…
  - Creating it by TWO tables have two points that can make it entirely fall down
    - Relationship between these tables can not exist!
    - We need to set RLS policy on both of these tables.

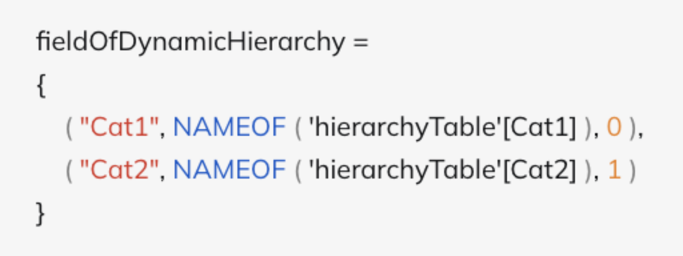
FIELD PARAMETER AS A HIERARCHY

Field parameters as a hierarchy

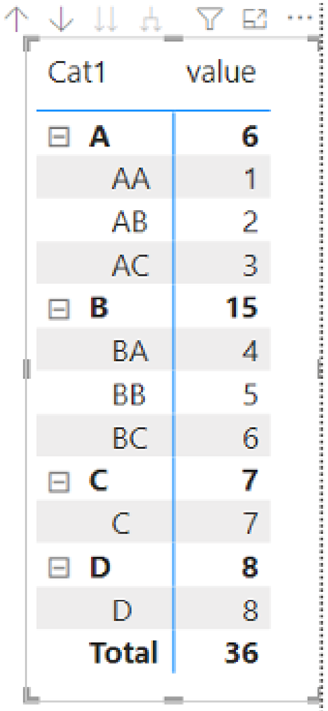
Additional grouping can be added very simply



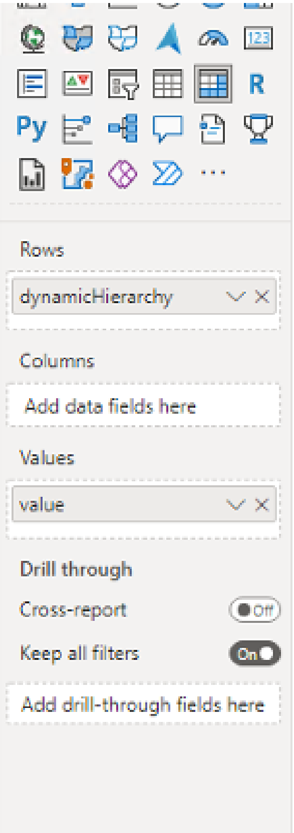
Cat1	Cat2	value
A	AA	1
A	AB	2
A	AC	3
B	BA	4
B	BB	5
B	BC	6
C	C	7
D	D	8



```
fieldOfDynamicHierarchy =
{
    ( "Cat1", NAMEOF ( 'hierarchyTable'[Cat1] ), 0 ),
    ( "Cat2", NAMEOF ( 'hierarchyTable'[Cat2] ), 1 )
}
```



Cat1	value
⊟ **A**	**6**
AA	1
AB	2
AC	3
⊟ **B**	**15**
BA	4
BB	5
BC	6
⊟ **C**	**7**
C	7
⊟ **D**	**8**
D	8
Total	**36**



Rows

dynamicHierarchy

Columns

Add data fields here

Values

value

Drill through

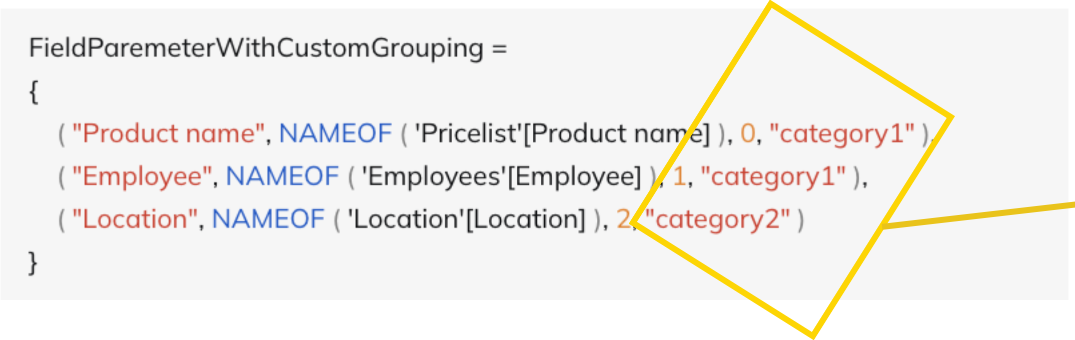
Cross-report Off

Keep all filters On

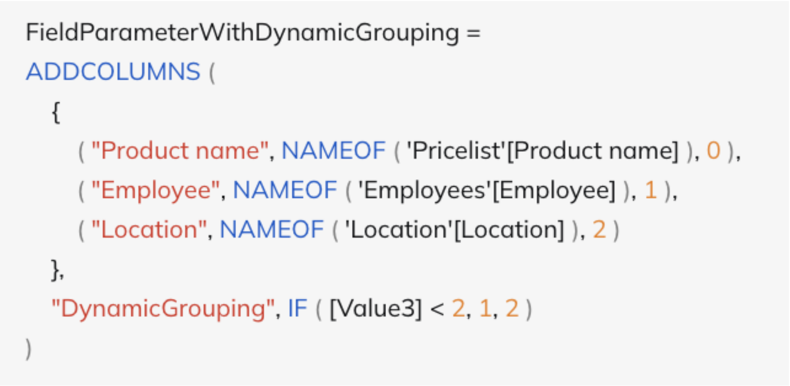
Add drill-through fields here

Field parameter with addition

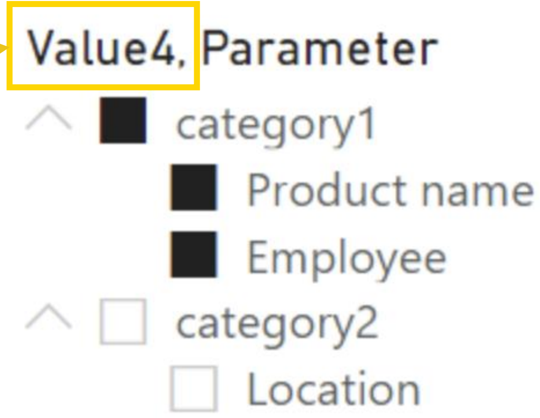
Additional grouping can be added very simply



```
FieldParemeterWithCustomGrouping =
{
    ( "Product name", NAMEOF ( 'Pricelist'[Product name] ), 0, "category1" ),
    ( "Employee", NAMEOF ( 'Employees'[Employee] ), 1, "category1" ),
    ( "Location", NAMEOF ( 'Location'[Location] ), 2, "category2" )
}
```



```
FieldParameterWithDynamicGrouping =
ADDCOLUMNS (
    {
        ( "Product name", NAMEOF ( 'Pricelist'[Product name] ), 0 ),
        ( "Employee", NAMEOF ( 'Employees'[Employee] ), 1 ),
        ( "Location", NAMEOF ( 'Location'[Location] ), 2 )
    },
    "DynamicGrouping", IF ( [Value3] < 2, 1, 2 )
)
```

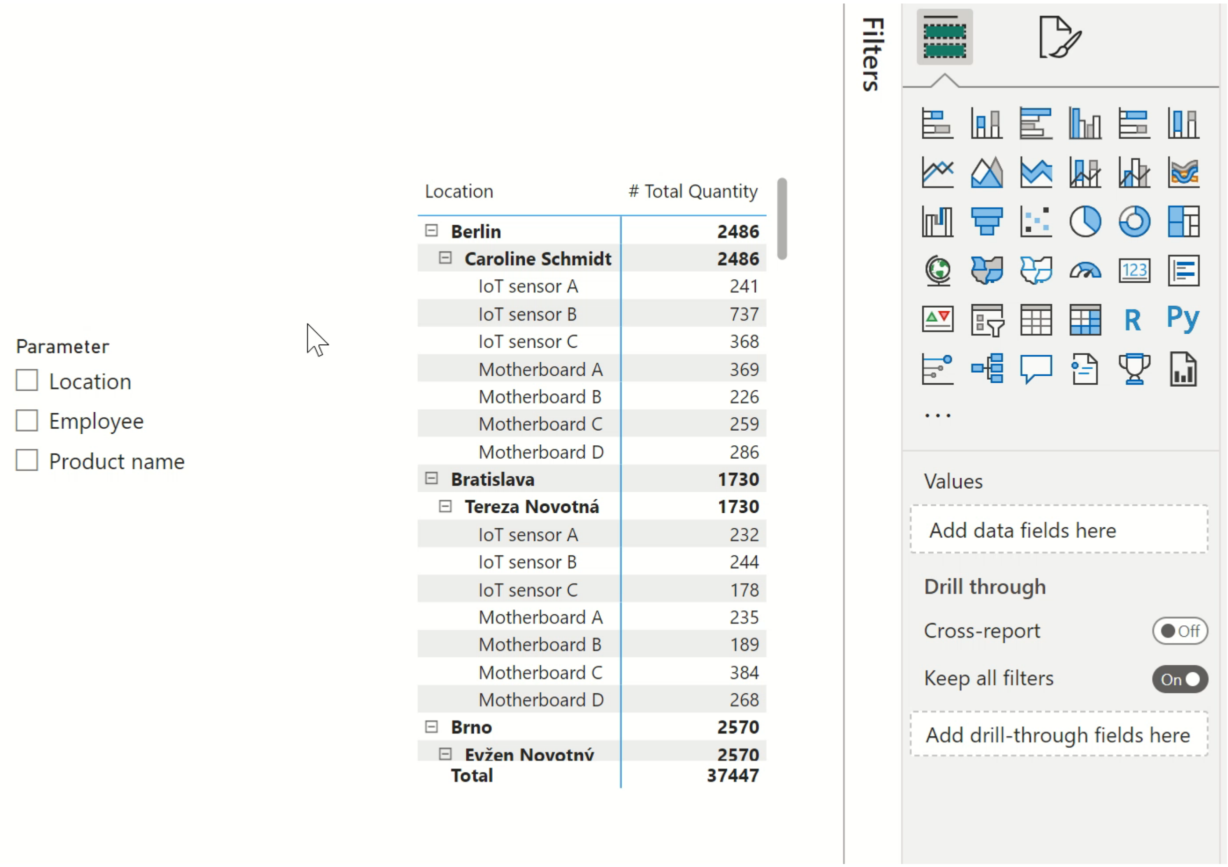


Value4, Parameter

- ■ category1
  - ■ Product name
  - ■ Employee
- □ category2
  - □ Location

Please be aware of them!

Field parameters can confuse end-users more then Calculation Groups



Parameter

□ Location

□ Employee

□ Product name

Location	# Total Quantity
⊟ **Berlin**	**2486**
⊟ **Caroline Schmidt**	**2486**
IoT sensor A	241
IoT sensor B	737
IoT sensor C	368
Motherboard A	369
Motherboard B	226
Motherboard C	259
Motherboard D	286
⊟ **Bratislava**	**1730**
⊟ **Tereza Novotná**	**1730**
IoT sensor A	232
IoT sensor B	244
IoT sensor C	178
Motherboard A	235
Motherboard B	189
Motherboard C	384
Motherboard D	268
⊟ **Brno**	**2570**
⊟ **Evžen Novotný**	**2570**
Total	**37447**

Filters

Values

Add data fields here

Drill through

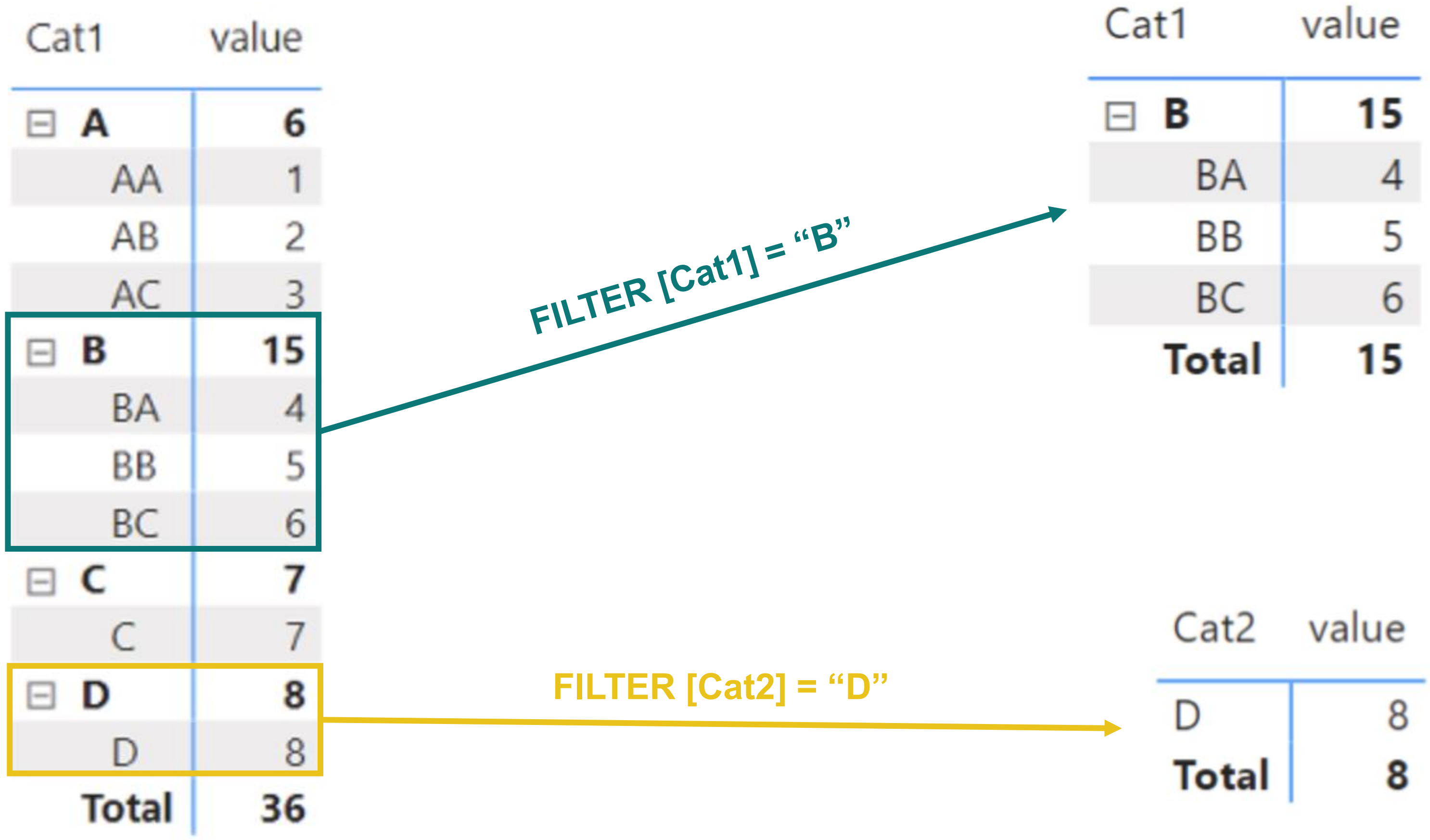
Cross-report: Off

Keep all filters: On

Add drill-through fields here

Dynamic display "+/-"

Additional grouping can be added very simply



Cat1	value
⊟ **A**	**6**
AA	1
AB	2
AC	3
⊟ **B**	**15**
BA	4
BB	5
BC	6
⊟ **C**	**7**
C	7
⊟ **D**	**8**
D	8
Total	**36**

FILTER [Cat1] = “B”

Cat1	value
⊟ **B**	**15**
BA	4
BB	5
BC	6
Total	**15**

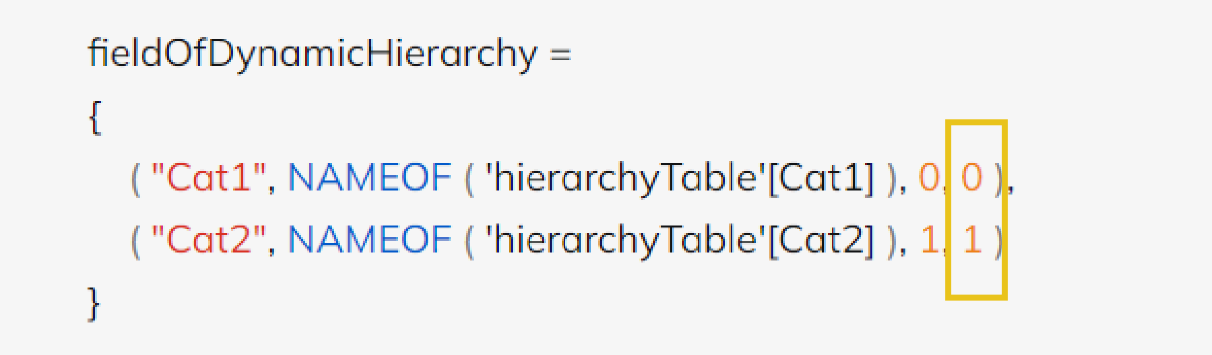
FILTER [Cat2] = “D”

Cat2	value
D	8
Total	**8**

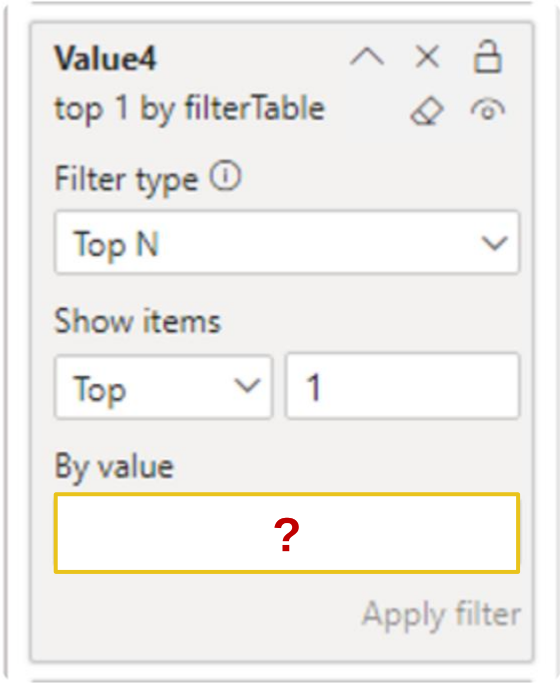
HOW?

One more colum is needed

Because native Order wouldn’t work itself



```
fieldOfDynamicHierarchy =
{
    ( "Cat1", NAMEOF ( 'hierarchyTable'[Cat1] ), 0, 0 ),
    ( "Cat2", NAMEOF ( 'hierarchyTable'[Cat2] ), 1, 1 )
}
```



Value4

top 1 by filterTable

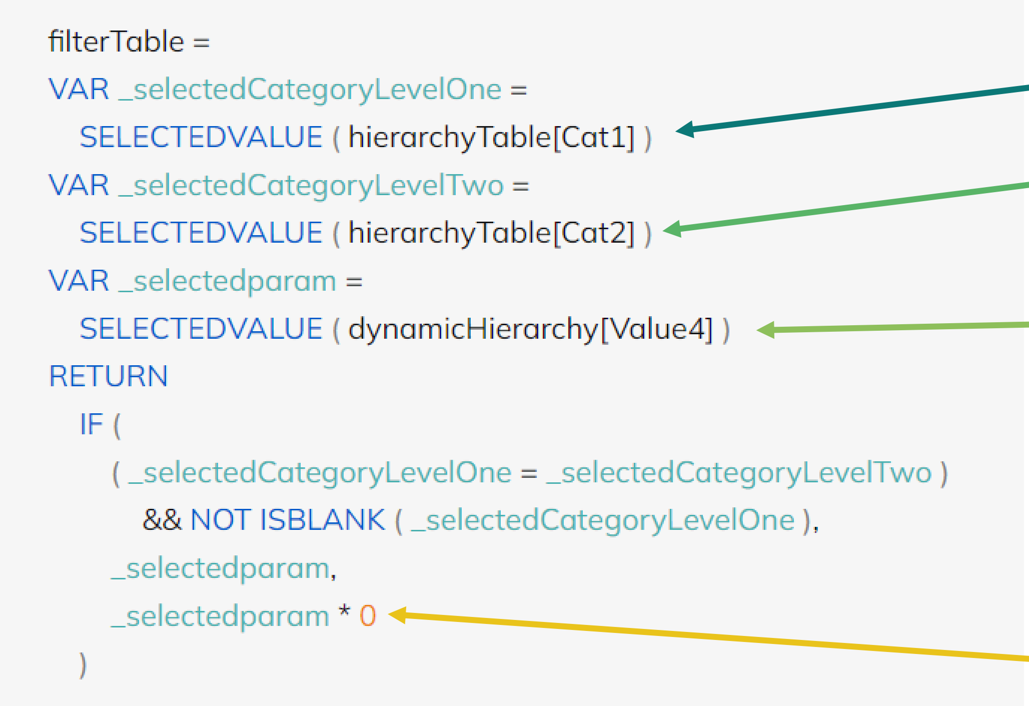
Filter type: Top N

Show items: Top 1

By value: ?

We need to use math to solve it

And because we are in school… what is X*0?



```
filterTable =
VAR _selectedCategoryLevelOne =
    SELECTEDVALUE ( hierarchyTable[Cat1] )
VAR _selectedCategoryLevelTwo =
    SELECTEDVALUE ( hierarchyTable[Cat2] )
VAR _selectedparam =
    SELECTEDVALUE ( dynamicHierarchy[Value4] )
RETURN
    IF (
        ( _selectedCategoryLevelOne = _selectedCategoryLevelTwo )
            && NOT ISBLANK ( _selectedCategoryLevelOne ),
        _selectedparam,
        _selectedparam * 0
    )
```

IF VALUE IS ACTUALY FROM FIRST COLUMN

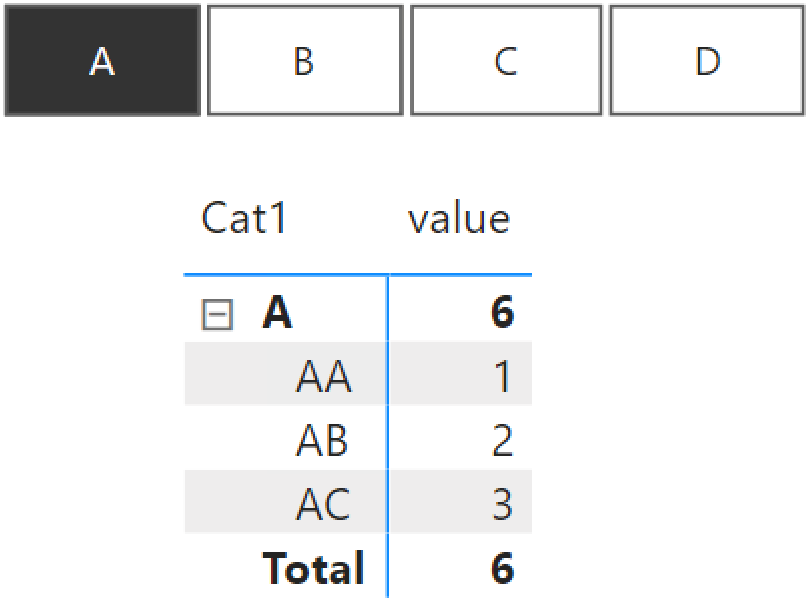
FROM A SECOND COLUMN

RECEIVING OUR “ORDER”

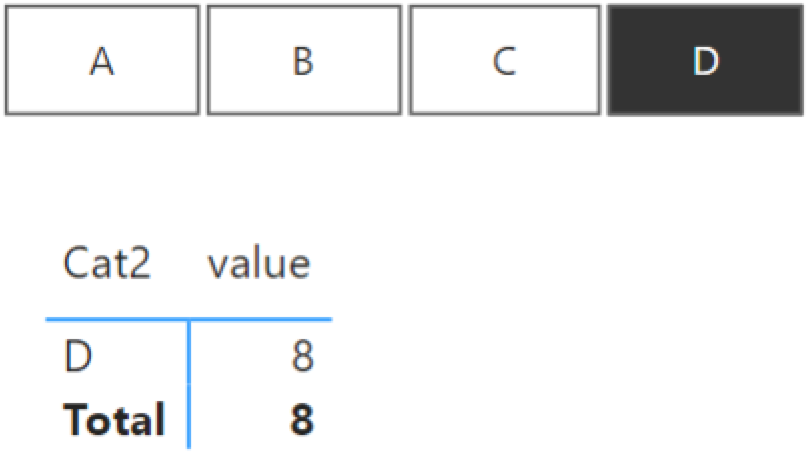
IF RECEIVED VALUES ARE THE SAME WE WILL SHOW THEM, OTHERWISE WE WILL MODIFY ALL VALUES TO ZERO

Active result

Isn’t that cool?



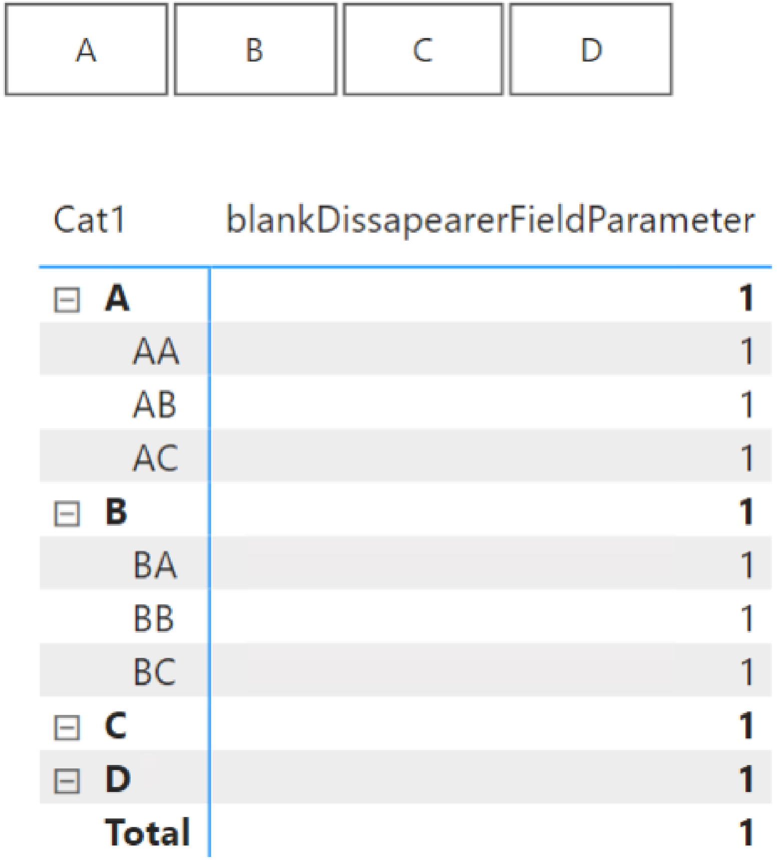
Cat1	value
⊟ **A**	**6**
AA	1
AB	2
AC	3
Total	**6**



Cat2	value
D	8
Total	**8**

Of course.. Without selection

We need to show everything



A	B	C	D

Cat1	blankDissapearerFieldParameter
⊟ **A**	**1**
AA	1
AB	1
AC	1
⊟ **B**	**1**
BA	1
BB	1
BC	1
⊟ **C**	**1**
⊟ **D**	**1**
Total	**1**

b3

REBTECH

SQLSERVICE
A PART OF SYSCO

24

solarwinds

dbWatch

redgate

devart

Snabbfoting

THANK YOU FOR THE ATTENTION

ŠTĚPÁN REŠL

LINKEDIN

TWITTER