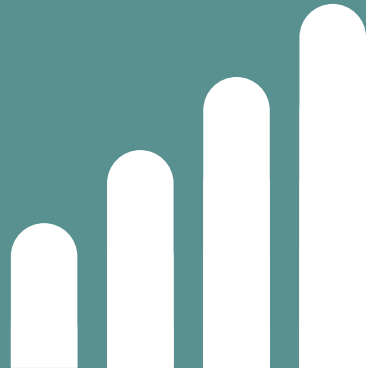


Power BI - DAX





DAX (Data Analysis Expressions)

- Power Pivot (Excel), Tabular, Power BI
- Počítá hodnoty na základě sloupců
- Velice podobný kalkulacím v excelu
- Je vyhodnocován zevnitř ven
- Je možné pomocí něj napočítat metriky, tabulky, sloupce i psát query



Calculated Columns vs Measures

Calculated Column

- Vyhodnocuje se v kontextu aktuálního řádku a náleží k dané tabulce
- vytěžuje RAM
- Vyhodnocuje se při aktualizaci dat
- `Table[Column]`
`Sales[Units]`
- Hodí se při kategorizaci, řádkových výpočtech ($\text{Price} * \text{Quantity}$)

Measure

- Nevyhodnocuje se v kontextu řádku, ale agreguje hodnoty, nenáleží tabulce, ale celému datovému modelu
- vytěžuje CPU
- Vyhodnocuje se vždy, když se použije
- `[Measure]`
`[Revenue]`
- Procenta, vážené průměry, specifické filtrování z různých tabulek

*home table



Evaluation Context

Row Context

- Iteruje řádek po řádku
- automaticky se objevuje u počítaných sloupců
- Obsahují ho iterační funkce (SUMX, AVERAGEX, MINX, ...)
- Vždy se první aplikuje filtrovací kontext a poté řádkový kontext

Filter Context

- Filtruje tabulku
- Vyskytuje se v kontextu řádku, sloupce, filtru, sliceru a dalších vizualizací nebo může být vytvořený pomocí DAX kalkulace.



Variables

- Hodí se pro zpřehlednění kódu, pro změnu pořadí vyhodnocení, případně pro testování
- umožní uložit část výpočtu do proměnné se kterou se dál může pracovat

```
- RevenueLD = CALCULATE([Revenue], Sales[Date] = MAX(Sales[Date]))
```

```
RevenueLD =
```

```
var v1 = MAX(Sales[Date])
```

```
var v2 = CALCULATE([Revenue], Sales[Date] = v1)
```

```
return v2
```

- Pro komentování se dá využít "--" případně "/*" což zakomentuje vše, co je za tímto znakem v daném řádku. pro komentování více řádků naráz se používá "/*" na začátku a "*/" na konci komentáře



Agregační funkce

SUM, AVERAGE, MIN, MAX, COUNT

- Agregují vždy jeden sloupec
- `Revenue = SUM(Sales[Revenue])`
`Revenue* = SUM(Sales[Price]*Sales[Units])`
`Revenue* = SUM(Sales[Price])*SUM(Sales[Units])`

SUMX, AVERAGEX, MINX, MAXX, COUNTX

- Agregují celé formule a formule se vyhodnotí pro každý řádek a poté se sečte výsledek
- `Revenue = SUMX(Sales, Sales[Revenue])`
`Revenue* = SUMX(Sales, Sales[Price]*Sales[Units])`
`Revenue* = Sales[Price]*Sales[Units]` **napočítaný sloupec**

* při psaní metrik, se musí vždy použít agregační funkce



Funkce FILTER a ALL

- Obě funkce vrací tabulku
 - **FILTER** nemění filtrovací kontext, ale slouží k přidání filtru na určitou tabulku
 - **ALL** ignoruje filtrovací kontext tabulky a vrátí celou tabulku, může být aplikovaná jak na tabulku, tak na sloupec
- ```
factSalesUSA = FILTER(Sales, Sales[Country] = "USA") tabulka
RevenueUSA = SUMX(FILTER(Sales, Sales[Country] = "USA"),
 Sales[Price]*Sales[Units])
RevenueALL = SUMX(ALL(Sales), Sales[Price]*Sales[Units])
RevenueUSA_ALL = SUMX(FILTER(ALL(Sales), Sales[Country] = "USA"),
 Sales[Price]*Sales[Units])
```



## Funkce FILTER a ALL

| Country      | Revenue              | RevenueUSA         | RevenueALL           | RevenueUSA_ALL     |
|--------------|----------------------|--------------------|----------------------|--------------------|
| USA          | 629,860,767          | 629,860,767        | 1,119,385,512        | 629,860,767        |
| Australia    | 306,689,348          |                    | 1,119,385,512        | 629,860,767        |
| Japan        | 92,432,112           |                    | 1,119,385,512        | 629,860,767        |
| Nigeria      | 34,244,259           |                    | 1,119,385,512        | 629,860,767        |
| Germany      | 28,666,708           |                    | 1,119,385,512        | 629,860,767        |
| Mexico       | 20,297,110           |                    | 1,119,385,512        | 629,860,767        |
| Canada       | 7,195,208            |                    | 1,119,385,512        | 629,860,767        |
| <b>Total</b> | <b>1,119,385,512</b> | <b>629,860,767</b> | <b>1,119,385,512</b> | <b>629,860,767</b> |





# Funkce CALCULATE

- Používáme pokud chceme nahradit daný filtrovací kontext za jiný
- Každá metrika je automaticky zabalená do calculate - chová se, jakoby tam funkce calculate byla použita.
- Může obsahovat více filtrů
  
- ```
RevenueUSA_C = CALCULATE ([Revenue], Sales[Country] = "USA")
```

```
RevenueUSA_C = CALCULATE ([Revenue],  
                           FILTER (ALL (Sales[Country]),  
                                   Sales[Country] = "USA"))
```

```
RevenueUSA_CF = CALCULATE ([Revenue],  
                           FILTER (Sales[Country] = "USA"))
```
- ```
RevenueUSA2014 = CALCULATE ([Revenue],
 Sales[Country] = "USA",
 'Date'[Year]= 2014)
```

\* Calculate při použití ve sloupcích



# Funkce CALCULATE

| Country      | Revenue              | RevenueUSA_C          | RevenueALL_C            | RevenueUSA_CF         |
|--------------|----------------------|-----------------------|-------------------------|-----------------------|
| USA          | 629,860,767          | 629,860,767.32        | 1,119,385,511.55        | 629,860,767.32        |
| Australia    | 306,689,348          | 629,860,767.32        | 1,119,385,511.55        |                       |
| Japan        | 92,432,112           | 629,860,767.32        | 1,119,385,511.55        |                       |
| Nigeria      | 34,244,259           | 629,860,767.32        | 1,119,385,511.55        |                       |
| Germany      | 28,666,708           | 629,860,767.32        | 1,119,385,511.55        |                       |
| Mexico       | 20,297,110           | 629,860,767.32        | 1,119,385,511.55        |                       |
| Canada       | 7,195,208            | 629,860,767.32        | 1,119,385,511.55        |                       |
| <b>Total</b> | <b>1,119,385,512</b> | <b>629,860,767.32</b> | <b>1,119,385,511.55</b> | <b>629,860,767.32</b> |



# Time Intelligence functions

- Funkce, které pracují s datem
- Pro správné fungování těchto funkcí je potřeba mít v datovém modelu sloupec, který obsahuje spojitě datové období.

## CALENDAR, CALENDARAUTO, DATEADD, SAMEPERIODLASTYEAR, TOTALYTD, DATE

- Funkce, CALENDAR potřebuje stanovit horní a dolní hranici datumů, funkce CALENDARAUTO skenuje celý dataset a pro minimální a maximální nalezené datum vybere vždy 1. den minimálního roku a poslední den maximálního roku a vygeneruje řadu automaticky
- `DateC = CALENDAR(MIN(Sales[Date]), MAX(Sales[Date]))`  
`DateC = CALENDAR( DATE(2013,1,1), DATE(2015,12,31) )`  
`DateAuto = CALENDARAUTO()`
- `RevenuePY = CALCULATE([Revenue], SAMEPERIODLASTYEAR('Date'[Date]))`  
`RevenuePM = CALCULATE([Revenue], DATEADD('Date'[Date], -1, MONTH))`  
`RevenueYTD = TOTALYTD([Revenue], 'Date'[Date])`



# Funkce IF

- Funkce, která se používá pro vytváření podmínek
- Podmínky se můžou kombinovat a být komplexní
  
- Revenue CY/PY =  $([Revenue] - [RevenuePY]) / [RevenuePY]$   
Revenue CY/PY = IF([RevenuePY] > 0,  
                          ([Revenue] - [RevenuePY]) / [RevenuePY],  
                          BLANK())  
Revenue CY/PY = DIVIDE([Revenue] - [RevenuePY], [RevenuePY])
  
- Revenue Share = DIVIDE([Revenue],  
                          CALCULATE([Revenue], ALL(Sales[Country])))
  
- vnořené podmínky: IF(condition1, true, IF(condition2, true, false))
- složené podmínky: IF(condition1 && condition2, true, false)   && je AND operátor  
                          IF(condition1 || condition2, true, false)   || je OR operátor



# Querying

- Psaní komplexních dotazů v jazyce dax
- Nové okno DAX Query View
- Vždy se musí začínat funkcí EVALUATE

- EVALUATE

```
SUMMARIZE (Sales,
 'Date' [Year],
 'Date' [Quarter],
 "Revenue", [Revenue])
```