

PWBI2

Pokročilá analýza dat pomocí jazyka DAX

Obsah

Syntaxe jazyka DAX	3
Slovníček pojmů	8
Názvy parametrů	9
Operátory.....	10
Aritmetické operátory	10
Operátory porovnávání	11
Operátor zřetězení textů	12
Logické operátory	12
Funkce (základní přehled)	13
Statistické funkce	13
Funkce pro práce s tabulkami	14
Relační funkce	15
Informační funkce	15
Funkce CALCULATE	16
Funkce časového měřítka	19
Zdrojové kódy – výpočty	21
Počítané tabulky	21
Počítané sloupce	26
Kalkulované míry	29
Kalkulované míry – úpravy kontextu výpočtu	31

Syntaxe jazyka DAX

Požadavky na syntaxi

Vzorec DAX vždy začíná znaménkem rovná se (=). Po znaménku rovná se můžete zadat libovolný výraz, který se vyhodnotí jako skalár, nebo výraz, který lze převést na skalár. Patří mezi ně:

- Skalární konstanta nebo výraz, který používá skalární operátor (+, -, *, /, >=, <=, &&, ...).
- Odkazy na sloupce nebo tabulky. Jazyk DAX jako vstupy pro funkce vždy používá tabulky a sloupce, nikdy pole nebo libovolnou sadu hodnot.
- Operátory, konstanty a hodnoty zadané jako součást výrazu.
- Výsledek funkce a jejích povinných argumentů. Některé funkce DAX vrátí místo skaláru tabulku a musí být zabaleny do funkce, která tuto tabulku vyhodnotí a vrátí skalár; kromě případů, kdy tabulka obsahuje jediný sloupec nebo jediný řádek. V takovém případě je tabulka považována za skalární hodnotu.

Většina funkcí DAX vyžaduje jeden nebo více argumentů, které mohou obsahovat tabulky, sloupce, výrazy a hodnoty. Některé funkce, jako například PI, nevyžadují žádné argumenty, ale vždy vyžadují závorky, které označují argument null. Například musíte vždy zadat PI(), nikoli PI. Funkce můžete také vnořit do jiných funkcí.

- Výrazy. Výraz může obsahovat cokoli nebo vše z následujícího: operátory, konstanty nebo odkazy na sloupce.

.....

.....

.....

.....

Například vše níže uvedené jsou platné vzorce.

POŽADAVKY NA SYNTAXI

Vzorec	Výsledek
= 3	3
= "Sales"	Sales
=	Pokud použijete tento vzorec v tabulce Prodej, získáte hodnotu sloupce
'Sales'[Částka]	Částka v tabulce Prodej pro aktuální řádek.
= (0.03	Tři procenta z hodnoty ve sloupci Částka v aktuální tabulce.
*[Částka])	I když se tento vzorec dá použít k výpočtu procenta, výsledek se nezobrazí
=0.03 * [Částka]	jako procento, pokud v tabulce nepoužijete formátování.
= PI()	Hodnota konstanty pí.

Poznámka

Vzorce se mohou chovat odlišně v závislosti na tom, zda jsou použity v počítaném sloupci nebo v míře v kontingenční tabulce. Musíte vždy znát kontext a vztah dat, která používáte ve vzorci, k ostatním datům, která mohou být použita při výpočtu.

.....

.....

.....

.....

Požadavky na pojmenování

Datový model často obsahuje více tabulek. Tabulky a jejich sloupce dohromady tvoří databázi uloženou v analytickém modulu (VertiPaq). V rámci této databáze musí mít všechny tabulky jedinečné názvy. Názvy sloupců musí být také jedinečné v každé tabulce. U všech názvů objektů se *nerozlišují velká a malá písmena*, například názvy **PRODEJ** a **Prodej** by představovaly stejnou tabulku.

Jednotlivé sloupce a míry, které přidáváte do existujícího datového modelu, musí patřit do konkrétní tabulky. Tabulku, která obsahuje sloupec, určíte buď implicitně, když vytvoříte počítaný sloupec v tabulce, nebo explicitně, když vytvoříte míru a zadáte název tabulky, do které má být definice míry uložena.

Když použijete tabulku nebo sloupec jako vstup pro funkci, musíte obecně *kvalifikovat* název sloupce. *Plně kvalifikovaný* název sloupce je název tabulky následovaný názvem sloupce v hranatých závorkách, například 'Prodej v USA'[Produkty]. Plně kvalifikovaný název je vždy vyžadován, když odkazujete na sloupec v následujících kontextech:

- Jako argument pro funkci, VALUES
- Jako argument pro funkce, ALL nebo ALLEXCEPT
- V argumentu filtru pro funkce, CALCULATE nebo CALCULATETABLE
- Jako argument pro funkci, RELATEDTABLE
- Jako argument pro libovolnou funkci časového měřítka

.....

.....

.....

.....

Nekvalifikovaný název sloupce je pouze název sloupce, který je uzavřený v hranatých závorkách, například [Částka prodeje]. Pokud například odkazujete na skalární hodnotu ze stejného řádku aktuální tabulky, můžete použít nekvalifikovaný název sloupce.

Pokud název tabulky obsahuje mezery, vyhrazená klíčová slova nebo nepovolené znaky, musíte uzavřít název tabulky do jednoduchých uvozovek. Do uvozovek musíte také uzavřít názvy tabulek, pokud obsahují znaky mimo rozsah alfanumerických znaků ANSI – bez ohledu na to, jestli vaše národní prostředí tuto sadu znaků podporuje, nebo ne. Pokud například otevřete sešit obsahující názvy tabulek zapsaných v cyrilici, například Таблица, název tabulky musí být uzavřen v uvozovkách, i když neobsahuje mezery.

Vyhrazená klíčová slova

Pokud je název, který použijete pro tabulku, stejný jako klíčové slovo rezervované službou Analysis Services, vyvolá se chyba a je nutné tabulku přejmenovat. V názvech objektů ale můžete klíčová slova použít, pokud je název objektu uzavřený v hranatých závorkách (platí pro sloupce) nebo uvozovkách (platí pro tabulky).

Speciální znaky

Následující znaky a typy znaků nejsou platné v názvech tabulek, sloupců nebo měr:

- Mezery na začátku nebo na konci; pokud mezery nejsou uzavřeny oddělovači názvů, hranatými závorkami nebo jednoduchými apostrofy.
- Řídící znaky
- Následující znaky nejsou platné v názvech objektů:

.,;':\|?&%\$!+=()[]{}<>

.....

.....

.....

.....

Datové typy

Nemusíte přetypovat, převést ani jinak určit datový typ pro sloupce nebo hodnoty používané ve vzorci DAX. Když ve vzorci DAX použijete data, DAX automaticky identifikuje datové typy v odkazovaných sloupcích a hodnotách, které zadáte, a v případě potřeby provede implicitní převody nutné pro dokončení zadané operace.

Pokud se například pokusíte přidat číslo k hodnotě data, modul interpretuje operaci v kontextu funkce a převede čísla na běžný datový typ a pak výsledek prezentuje v zamýšleném formátu, čili jako datum.

Existují však určitá omezení pro hodnoty, které lze úspěšně převést. Pokud má hodnota nebo sloupec datový typ, který je nekompatibilní s aktuální operací, DAX vrátí chybu. Jazyk DAX také neposkytuje funkce, které umožňují explicitně měnit, převádět nebo přetypovat datový typ existujících dat, která jste importovali do datového modelu.

.....
.....
.....
.....

Slovníček pojmů

Počítaný sloupec

Výpočet modelu, pomocí kterého se do tabulkového modelu přidá sloupec zápisem vzorce DAX. Vzorec musí vracet skalární hodnotu a vyhodnocuje se pro každý řádek v tabulce. Počítaný sloupec se dá přidat do tabulky režimu úložiště Import nebo DirectQuery.

Počítaná míra

V modelování tabulek neexistuje žádný koncept *počítané míry*. Místo toho použijte *míru*. V pojmech počítané tabulky a počítané sloupce se používá slovo *počítané*. Odlišuje je od tabulek a sloupců, které pocházejí z Power Query. Power Query nemá koncept míry.

Počítaná tabulka

Výpočet modelu, pomocí kterého se do tabulkového modelu přidá tabulka zápisem vzorce DAX. Vzorec musí vracet objekt tabulky. Výsledkem je tabulka, která používá režim úložiště Import.

Výpočet

Komplikovaný proces, který převede nejméně jeden vstup na jeden nebo více výsledků. V modelu tabulkových dat může být výpočet objekt modelu; buď počítaná tabulka, počítaný sloupec, nebo míra.

Kontext

Popisuje prostředí, ve kterém se vyhodnocuje vzorec DAX. Existují dva typy kontextu: *Kontext řádku* a *kontext filtru*. Kontext řádku představuje aktuální řádek a používá se k vyhodnocení vzorců počítaných sloupců jako výrazů, které se používají v iterátorech tabulek. Kontext filtru se používá k vyhodnocení měr a představuje filtry zavedené přímo pro sloupce modelu a filtry šířené relacemi modelu.

.....

.....

.....

.....

.....

Názvy parametrů

NÁZVY PARAMETRŮ	
Označení	Definice
expression	Libovolný výraz DAX, který vrací jednu skalární hodnotu a který se má vyhodnotit několikrát (pro každý řádek/kontext)
hodnota	Libovolný výraz DAX, který vrací jednu skalární hodnotu a který se má vyhodnotit právě jednou před všemi ostatními operacemi.
table	Libovolný výraz DAX, který vrací tabulku dat.
tableName	Název existující tabulky pomocí standardní syntaxe jazyka DAX. Nemůže se jednat o výraz.
columnName	Název existujícího sloupce pomocí standardní syntaxe jazyka DAX, obvykle plně kvalifikovaný. Nemůže se jednat o výraz.
jméno	Řetězcová konstanta, která se použije k zadání názvu nového objektu.
order	Výčet sloužící k určení pořadí řazení.
ties	Výčet sloužící k určení zpracování hodnot vazeb.
type	Výčet sloužící k určení datového typu pro PathItem a PathItemReverse.

.....

.....

.....

.....

.....

Operátory

Aritmetické operátory

K provádění základních matematických operací, jako je sčítání, odčítání nebo násobení, ke kombinování čísel a vytváření číselných výsledků používejte následující aritmetické operátory.

ARITMETICKÉ OPERÁTORY		
Aritmetický operátor	Význam	Příklad
+ (znaménko plus)	Sčítání	3+3
- (znaménko minus)	Odčítání nebo znaménko minus	3-1-1
* (hvězdička)	Násobení	3*3
/ (lomítko)	Dělení	3/3
^ (stříška)	Umocnění	16^4

.....

.....

.....

.....

Operátory porovnávání

Pomocí následujících operátorů můžete porovnat dvě hodnoty. Při porovnání dvou hodnot pomocí těchto operátorů je výsledkem logická hodnota – TRUE nebo FALSE.

OPERÁTORY POROVNÁVÁNÍ		
Operátor porovnání	Význam	Příklad
=	Je rovno	[Region] = "USA"
==	Je přesně rovno	[Region] = "USA"
>	Je větší než	[Sales Date] > "Jan 2009"
<	Je menší než	[Sales Date] < "Jan 1 2009"
>=	Větší než nebo rovno	[Amount] >= 20000
<=	Menší než nebo rovno	[Amount] <= 100
<>	Není rovno	[Region] <> "USA"

Všechny operátory porovnání kromě operátoru == považují BLANK za rovnající se číslu 0, prázdnému řetězci "", hodnotě DATE(1899, 12, 30) nebo FALSE. V důsledku toho bude mít porovnání [Column] = 0 hodnotu TRUE v případech, kdy hodnota sloupce [Column] je 0 nebo BLANK. Naopak porovnání [Column] = = 0 bude TRUE jenom v případě, že hodnota sloupce [Column] je 0.

.....

.....

.....

.....

.....

Operátor zřetězení textů

Pomocí ampersandu (&) se spojí (neboli zřetězí) dva nebo více textových řetězců, aby vznikl jeden souvislý text.

OPERÁTOR ZŘETĚZENÍ TEXTŮ		
Textový operátor	Význam	Příklad
& (ampersand)	Spojí (neboli zřetězí) dvě hodnoty, aby vznikla jedna souvislá textová hodnota.	[Region] & ", " & [City]

Logické operátory

Pomocí logických operátorů (&&) a (||) můžete kombinovat výrazy, aby vznikl jeden výsledek.

LOGICKÉ OPERÁTORY		
Textový operátor	Význam	Příklady
&& (dvojitý ampersand)	Vytvoří podmínku AND mezi dvěma výrazy, z nichž každý má jako výsledek logickou hodnotu. Pokud oba výrazy vrátí hodnotu TRUE, kombinace výrazů také vrátí hodnotu TRUE, v opačném případě kombinace vrátí hodnotu FALSE.	([Region] = "France") && ([BikeBuyer] = "yes"))
(dvojitý symbol svislé čáry)	Vytvoří podmínku OR mezi dvěma logickými výrazy. Pokud některý výraz vrátí hodnotu TRUE, výsledek je TRUE. Výsledek je FALSE jenom v případě, že oba výrazy jsou FALSE.	(([Region] = "France") ([BikeBuyer] = "yes"))
IN	Vytvoří logickou podmínku OR mezi každým řádkem porovnávaným s tabulkou. Poznámka: Syntaxe konstrukturu tabulky používá složené závorky.	'Product'[Color] IN { "Red", "Blue", "Black" }

.....

.....

.....

.....

.....

Funkce (základní přehled)

Statistické funkce

U vybraných statistických funkcí rozlišujeme to, jaké parametry tyto funkce přijímají a jak tyto funkce pracují s prázdnými hodnotami. Všeobecně platí, že základní statistické funkce počítají napříč celým sloupcem, např. COUNT(), pokud potřebujeme tento výpočet omezit / podmínit, k tomuto slouží funkce končící na X, např. COUNTX() (ekvivalent COUNTIF v MS Excel). Pokud chceme počítat i s prázdnými hodnotami, k tomuto slouží funkce typu COUNTA(), AVERAGEA() apod.

Funkce	Popis
AVERAGE	Vrátí průměr (aritmetickou střední hodnotu) všech čísel ve sloupci.
AVERAGEA	Vrátí průměr (aritmetickou střední hodnotu) hodnot ve sloupci.
AVERAGEX	Vypočítá průměr (aritmetickou střední hodnotu) sady výrazů vyhodnocených v tabulce.
COUNT	Spočítá počet buněk ve sloupci, které obsahují čísla.
COUNTA	Spočítá počet buněk ve sloupci, které nejsou prázdné.
COUNTX	Spočítá neprázdné výsledky při vyhodnocení výsledku výrazu v tabulce.
MAX	Vrátí největší číselnou hodnotu ve sloupci nebo mezi dvěma skalárními výrazy.
MAXA	Vrátí největší hodnotu ve sloupci.
MAXX	Vyhodnotí výraz pro každý řádek tabulky a vrátí největší číselnou hodnotu.
MIN	Vrátí nejmenší číselnou hodnotu ve sloupci nebo mezi dvěma skalárními výrazy.
MINA	Vrátí nejmenší hodnotu ve sloupci včetně všech logických hodnot a čísel vyjádřených jako text.
MINX	Vrátí nejmenší číselnou hodnotu, která je výsledkem vyhodnocení výrazu pro každý řádek tabulky.

Funkce	Popis
COUNTBLANK	Spočítá počet prázdných buněk ve sloupci.
COUNTROWS	Spočítá počet řádků v zadané tabulce nebo v tabulce definované výrazem.
COUNTX	Při vyhodnocování výrazu v tabulce spočítá počet řádků, které obsahují číslo nebo výraz, který se vyhodnotí na číslo.
DISTINCTCOUNT	Spočítá počet jedinečných hodnot ve sloupci.
DISTINCTCOUNTNOBLANK	Spočítá počet jedinečných hodnot ve sloupci.

Funkce pro práce s tabulkami

Tyto funkce vrací tabulku nebo pracují se stávajícími tabulkami.

Funkce	Popis
ADDCOLUMNS	Přidá do dané tabulky nebo výrazu tabulky počítané sloupce.
GENERATE	Vrátí tabulku s kartézským součinem každého řádku tabulky <i>table1</i> s tabulkou, která je výsledkem vyhodnocení tabulky <i>table2</i> v kontextu aktuálního řádku z tabulky <i>table1</i> .
GENERATEALL	Vrátí tabulku s kartézským součinem každého řádku tabulky <i>table1</i> s tabulkou, která je výsledkem vyhodnocení tabulky <i>table2</i> v kontextu aktuálního řádku z tabulky <i>table1</i> .
ROW	Vrátí tabulku s jedním řádkem obsahujícím hodnoty, které jsou výsledkem výrazů zadaných v jednotlivých sloupcích.
SUMMARIZE	Vrátí souhrnnou tabulku pro požadované celkové součty sady skupin.

Relační funkce

Slouží ke správě a využívání relací mezi tabulkami.

Funkce	Popis
CROSSFILTER	Určuje směr křížového filtrování, který se má použít při výpočtu relace, která existuje mezi dvěma sloupci.
RELATED	Vrátí hodnotu v relaci z jiné tabulky.
RELATEDTABLE	Vyhodnotí výraz tabulky v kontextu upraveném danými filtry.
USERELATIONSHIP	Určuje relaci, která se má použít v určitém výpočtu, jako relaci, která existuje mezi sloupcem columnName1 a columnName2.

Informační funkce

Informační funkce DAX považují zadanou buňku nebo řádek za argument a oznamují, jestli hodnota odpovídá očekávanému typu. Například funkce ISERROR vrátí hodnotu TRUE, pokud vámi odkazovaná hodnota obsahuje chybu.

Funkce	Description
HASONEFILTER	Vrátí hodnotu TRUE, pokud je počet přímo filtrovaných hodnot ve sloupci <i>columnName</i> roven jedné; v opačném případě vrátí hodnotu FALSE.
HASONEVALUE	Vrátí hodnotu TRUE, když je kontext pro sloupec <i>columnName</i> vyfiltrovaný jenom na jednu jedinečnou hodnotu. V opačném případě je FALSE.
ISBLANK	Ověří, zda je hodnota prázdná, a vrátí hodnotu TRUE nebo FALSE.
ISCROSSFILTERED	Vrátí hodnotu TRUE, pokud je sloupec <i>columnName</i> nebo jiný sloupec ve stejné nebo související tabulce filtrovaný.
ISEMPTY	Zkontroluje, zda je tabulka prázdná.
ISERROR	Ověří, zda je hodnota chyba, a vrátí hodnotu TRUE nebo FALSE.
ISFILTERED	Vrátí hodnotu TRUE, pokud je sloupec <i>columnName</i> filtrovaný přímo.
ISINSCOPE	Vrátí hodnotu true, pokud je zadaný sloupec úroveň v hierarchii úrovní.

.....

.....

.....

Funkce CALCULATE

Syntaxe

CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])

PARAMETRY	
Pojem	Definice
expression	Výraz, který se má vyhodnotit
filter1, filter2,...	(Nepovinné) Logické výrazy nebo výrazy tabulek, které definují filtry nebo funkce modifikátorů filtrů

Výraz použitý jako první parametr je v podstatě stejný jako míra.

Filtry můžou být:

- Výrazy logických filtrů
- Výrazy filtrů tabulek
- Funkce pro úpravu filtru

Když se používá několik filtrů, vyhodnocují se pomocí logického operátoru AND. To znamená, že všechny podmínky se musí splnit najednou.

Výrazy logických filtrů

Logický filtr výrazů je výraz, který se vyhodnotí jako TRUE, nebo FALSE. Existuje několik pravidel, kterými se tyto výrazy musí řídit.

- Můžou se odkazovat jen na jeden sloupec.
- Nemůžou se odkazovat na míry.
- Nemůžou používat vnořenou funkci CALCULATE.
- Nemůžou používat funkce, které procházejí nebo vracejí tabulku.

To platí i pro agregační funkce.

.....

.....

.....

.....

Výraz filtru tabulky

Filtr tabulkových výrazů použije objekt tabulky jako filtr. Může to být odkaz na tabulku modelu, ale pravděpodobněji to bude funkce, která vrací objekt tabulky. Pomocí funkce FILTER můžete zavádět složité podmínky filtrů, včetně těch, které se nedají definovat výrazem logických filtrů.

Funkce modifikátorů filtrů

Funkce úprav filtrů umožňují dělat více než jen přidávat filtry. Při úpravě kontextu filtru nabízejí i další možnosti.

FUNKCE MODIFIKÁTORŮ FILTRŮ	
Funkce	Účel
REMOVEFILTERS	Odebere všechny filtry, nebo filtry z jednoho nebo více sloupců tabulky, nebo ze všech sloupců jedné tabulky.
ALL ¹ , ALLEXCEPT, ALLNOBLANKROW	Odebere filtry z jednoho nebo více sloupců, nebo ze všech sloupců jedné tabulky.
KEEPFILTERS	Přidá filtr, aniž by se v daných sloupcích odebraly stávající filtry.
USERELATIONSHIP	Zapojí neaktivní relaci mezi souvisejícími sloupci, což znamená, že se aktivní relace automaticky převede na neaktivní.
CROSSFILTER	Změní směr filtru (z obou na jeden nebo z jednoho na oba), nebo zakáže relaci.

¹ Funkce ALL a její varianty se chovají jako modifikátory obou filtrů a jako funkce, které vracejí objekty tabulek. Pokud váš nástroj podporuje funkci REMOVEFILTERS, k odebírání filtrů je lepší použít tu.

.....

.....

.....

.....

.....

Vrácená hodnota

Hodnota, která je výsledkem výrazu

Poznámky

- Když se zadají výrazy filtrů, funkce CALCULATE upraví kontext filtru, aby se vyhodnotil výraz. Pokud výraz filtru není začleněný do funkce KEEPFILTERS, pro jednotlivé výrazy filtrů existují dva standardní výsledky:
 - Pokud se sloupce (nebo tabulky) nenacházejí v kontextu filtru, do kontextu se přidají nové filtry, aby se vyhodnotil výraz.
 - Pokud se sloupce (nebo tabulky) v kontextu filtru už nacházejí, stávající filtry se přepíší novými filtry, aby se vyhodnotil výraz CALCULATE.
- Funkce CALCULATE, která se použije *bez filtrů*, dosáhne konkrétního požadavku. Převeď kontext řádku na kontext filtru. To je zapotřebí, když se výraz (ne míra modelu), který shrnuje data modelu, musí vyhodnotit v kontextu řádku. K tomuto scénáři může dojít ve vzorci počítaného sloupce nebo při vyhodnocení výrazu pomocí funkce iterátoru.

Poznámka: Když se míra modelu použije v kontextu řádků, převod kontextu je automatický.

.....

.....

.....

.....

.....

Funkce časového měřítka

Jazyk DAX (Data Analysis Expressions) zahrnuje funkce časového měřítka, které umožňují pracovat s daty pomocí časových období, včetně dnů, měsíců, čtvrtletí a let, a potom u těchto období provádět a porovnávat výpočty.

Funkce	Popis
DATEADD	Vrátí tabulku se sloupcem kalendářních dat, která jsou v čase posunuta dopředu nebo dozadu o zadaný počet intervalů od dat v aktuálním kontextu.
DATESBETWEEN	Vrátí tabulku obsahující sloupec kalendářních dat, která začínají v zadaný počáteční den a pokračují až do zadaného koncového dne.
DATESINPERIOD	Vrátí tabulku obsahující sloupec kalendářních dat, která začínají v zadaný počáteční den a pokračují po zadaný počet a typ intervalů dat.
DATESMTD	Vrátí tabulku, která obsahuje sloupec kalendářních dat od začátku měsíce v aktuálním kontextu.
DATESQTD	Vrátí tabulku, která obsahuje sloupec kalendářních dat od začátku čtvrtletí v aktuálním kontextu.
DATESYTD	Vrátí tabulku, která obsahuje sloupec kalendářních dat od začátku roku v aktuálním kontextu.

.....

.....

.....

.....

.....

Funkce	Popis
NEXTDAY	Vrátí tabulku, která obsahuje sloupec všech kalendářních dat z následujícího dne na základě prvního data zadaného ve sloupci dates v aktuálním kontextu.
NEXTMONTH	Vrátí tabulku, která obsahuje sloupec všech kalendářních dat z následujícího měsíce na základě prvního data ve sloupci dates v aktuálním kontextu.
NEXTQUARTER	Vrátí tabulku, která obsahuje sloupec všech kalendářních dat v následujícím čtvrtletí na základě prvního data zadaného ve sloupci dates v aktuálním kontextu.
NEXTYEAR	Vrátí tabulku obsahující sloupec všech kalendářních dat v dalším roce na základě prvního data ve sloupci dates v aktuálním kontextu.
PARALLELPERIOD	Vrátí tabulku, která obsahuje sloupec kalendářních dat představující období paralelní ke kalendářním datům v zadaném sloupci dates v aktuálním kontextu s daty posunutými o určitý počet intervalů dopředu nebo zpět v čase.
PREVIOUSDAY	Vrátí tabulku obsahující sloupec všech kalendářních dat představujících den, který předchází prvnímu datu ve sloupci dates v aktuálním kontextu.
PREVIOUSMONTH	Vrátí tabulku obsahující sloupec všech kalendářních dat z předchozího měsíce na základě prvního data ve sloupci dates v aktuálním kontextu.
PREVIOUSQUARTER	Vrátí tabulku obsahující sloupec všech kalendářních dat z předchozího čtvrtletí na základě prvního data ve sloupci dates v aktuálním kontextu.
PREVIOUSYEAR	Vrátí tabulku obsahující sloupec všech kalendářních dat z předchozího roku vzhledem k poslednímu datu ve sloupci dates v aktuálním kontextu.
SAMEPERIODLASTYEAR	Vrátí tabulku se sloupcem kalendářních dat, která jsou posunuta o jeden rok zpět od kalendářních dat v zadaném sloupci dates v aktuálním kontextu.

.....

.....

.....

Zdrojové kódy – výpočty

Počítané tabulky

_1. zakladni ukazatele Cesko =

```
/*  
Funkce filtr ma jako navratovou hodnotu tabulku, nad kterou jsou definovany filtry  
*/  
  
FILTER(  
    'zakladni ukazatele - Cesko', // tabulka, ze ktere funkce filtr vychazi  
    'zakladni ukazatele - Cesko'[Rok] IN {2016,2017,2018,2019} // filtry aplikovane nad sloupcem [Rok]  
)
```

```
_2. zakladni ukazatele Cesko ROW =
```

```
/*
```

```
Funkce ROW() ma navratovou hodnotu tabulku o jednom radku, kdy v ramci vypoctu definujeme nazvy sloupce a jejich hodnoty
```

```
*/
```

```
ROW(
```

```
    "Zakladni ukazatele Cesko - max obyvatel", // nazev sloupce
```

```
    | MAX('zakladni ukazatele - Cesko'[Pocet obyvatel]), // do radku bude vypsana max. hodnota ze 'zakladni ukazatele - Cesko'[Pocet obyvatel])
```

```
    "Zakladni ukazatele Cesko - min obyvatel", // nazev sloupce
```

```
    MIN('_1. zakladni ukazatele Cesko'[Pocet obyvatel]) // do radku bude vypsana max. hodnota ze 'zakladni ukazatele - Cesko'[Pocet obyvatel])
```

```
)
```

_3. zakladni ukazatele Cesko SUMMARIZE =

/*

Funkce SUMMARIZE() ma jako navratovou hodnotu agregovanou tabulku.

*/

SUMMARIZE(

 'zakladni ukazatele - Cesko', // tabulka, nad kterou jsou provadeny vypocty

 'zakladni ukazatele - Cesko'[Zeme], // groupovani hodnod nad sloupcem [Zeme]

 "prumerny pocet obyvatel", // nazev nove pridaneho sloupce

 AVERAGE('zakladni ukazatele - Cesko'[Pocet obyvatel]), // operace, jejiz navratova hodnota je uvedena v radcich sloupce

 "pocet obyvatel v roce 2019", // nazev nove pridaneho sloupce

 // hodnota ve sloupci "pocet obyvatel 2019" je definovana iteratorem

 MAXX(

 FILTER('zakladni ukazatele - Cesko', // tabulka, nad kterou jsou uplatneny filtry

 'zakladni ukazatele - Cesko'[Rok] = 2019), // definice filtru nad sloupcem [Rok]

 'zakladni ukazatele - Cesko'[Pocet obyvatel]) // sloupec, nad kterem je po aplikaci filtru proveden vypocet

)

_4. zakladni ukazatele Cesko SUMMARIZE ADDCOLUMNS =

```
ADDCOLUMNS(
```

```
//ZDE ZACINA FUNKCE SUMMARIZE, JEJIZ NAVRATOVOU HODNOTOU JE TABULKA
```

```
SUMMARIZE(
```

```
    'zakladni ukazatele - Cesko', // tabulka, nad kteoru jsou provadeny vypocty
```

```
    'zakladni ukazatele - Cesko'[Zeme], // uvedeni sloupce, na zaklade ktereho jsou hodnoty groupovany
```

```
    "pocet obyvatel v roce 2019", // nazev noveho sloupce
```

```
    MAXX(
```

```
        FILTER('zakladni ukazatele - Cesko', // filtrovana tabulka
```

```
            'zakladni ukazatele - Cesko'[Rok] = 2019), // definice filtru nad sloupcem [Rok]
```

```
        'zakladni ukazatele - Cesko'[Pocet obyvatel]), // sloupec, který po filtrovani vstupuje do vypoctu
```

```
    "maximum z poctu obyvatel", // nazev noveho sloupce
```

```
    MAX('zakladni ukazatele - Cesko'[Pocet obyvatel])) // funkce, jejiz navratova hodnota je vypsana do radku sloupce
```

```
),
```

```
// ZDE KONCI FUNKCE SUMMARIZE, JEJIZ NAVRATOVOU HODNOTOU JE TABULKA
```

```
    "pocet obyvatel 2019 - maximum pocu obyvatel", // nazev sloupce pridaneho do tabulky, která je výsledkem funkce SUMMARIZE()
```

```
    //pozn. funkce ADDCOLUMNS() umi čist sloupce z tabulky vrácené funkci SUMMARIZE(), proto do argumentu této funkce můžeme vložit sloupce  
    tabulky vytvořené funkcí SUMMARIZE()
```

```
    [pocet obyvatel v roce 2019] - [maximum z pocu obyvatel]) // operace, jejiz vysledek je vypsán do radku nové přidaneho sloupce
```


_5. zakladni ukazatele Cesko CALCULATETABLE =

```
CALCULATETABLE(  
    'zakladni ukazatele - Cesko', // tabulka, ze ktere vychazime pri generovani kalkulovane tabulky za pomoci funkce CALCULATETABLE()  
    'zakladni ukazatele - Cesko'[Rok] = 2019 // filtry aplikovane nad tabulkou dosazenou v prvni argumentu  
)
```

_6. zakladni ukazatele cesko vsechny ukazatele =

```
VALUES('zakladni ukazatele - Cesko'[Rok]) // funkce vracejici hodnoty ze sloupce dosazeneho do argumentu
```

Počítané sloupce

```
pocet obyvatel =
```

```
/*
```

V tomto vypočtu přidáváme kalkulovaný sloupec. Hodnoty do tohoto sloupce dosadíme z tabulky, která se základní tabulkou není připojena žádnou relací.

```
*/
```

```
CALCULATE(
```

```
    SUM('zakladni ukazatele - Cesko'[Pocet obyvatel]), // agrekacni funkce nad sloupcem cizi nepropojene tabulky
```

```
    // za pomoci funkce FILTER() filtrujeme nepropojenou tabulku. Jedna se o stejnou tabulku, nad jejímž sloupcem provádíme funkci pro agregaci
```

```
    FILTER('zakladni ukazatele - Cesko','zakladni ukazatele - Cesko'[Rok]
```

```
/*
```

Funkce EARLIER() vrátí hodnotu na daném řádku ze sloupce, který je dosazen jako argument

Jelikož funkce FILTER() je iterator a funkce EARLIER je v ní vnorena, funkce EARLIER() bude vypočtena na každém řádku a její návratová hodnota poslouží pro filtrování cizí tabulky

```
*/
```

```
    = EARLIER('_6. zakladni ukazatele cesko vsechny ukazatele'[Rok])
```

```
)
```

```
| )
```

podil vek. skupiny 25-49 =

/*

V tomto vypočtu pridávame kalkulovaný sloupec. Hodnoty do tohoto sloupce dosadíme z tabulky, která se základní tabulkou je připojena aktivní relací.

*/

```
MAXX(  
    FILTER( // Funkce FILTER() je iterátorem, která provádí výpočet na každém řádku tabulky dosazené v prvním argumentu  
    RELATEDTABLE('demografické údaje - věk - Česko'), // v tomto případě je touto tabulkou "cizí" tabulka připojena relací  
    'demografické údaje - věk - Česko'[Věková skupina] = "25-49"), // tato "cizí" tabulka je následně filtrována nad sloupcem [Věková skupina]  
    'demografické údaje - věk - Česko'[Podíl]) // sloupec, u kterého je proveden výpočet, tj. dohledáno maximum
```

podil zen =

/*

V tomto vypočtu pridávame kalkulovaný stĺpec. Hodnoty do tohoto stĺpca dosadíme z tabuľky, ktorá sa základni tabuľkou je pripojená neaktívnej relácii, ktorou pro potreby tohoto vypočtu aktivujeme funkciu USERELATIONSHIP().

*/

CALCULATE(

// výraz, tj. funkcia, vykonaná nad filtrovanou tabuľkou. Filtry tabuľky jsou definované v druhom a ďalšom argumentu

MAX('demograficke udaje - pohlavi - Cesko'[Podil]),

// funkcia USERELATIONSHIP() aktivuje neaktívnu reláciu

// ZDE ZACINAJI FILTRY DEFINOVANE V RAMCI FUNKCE CALCULATE()

///Pokud je definováno více filtru, návratovou hodnotou je tabuľka splňující všechny filtry, tj.

// je zde použit operator "AND"

// 1. filtr definování funkci USERELATIONSHIP()

USERELATIONSHIP(

'_6. základní ukazatele cesko všechny ukazatele'[Rok], // sloupec tabuľky, v jejímž kalkulovaném sloupci dochází k výpočtu

'demograficke udaje - pohlavi - Cesko'[Rok]), // sloupec cizí tabuľky propojené neaktívnu reláci

// 2. filtr definovaný funkci FILTER(), zde již nepoužíváme žádné relace

FILTER('demograficke udaje - pohlavi - Cesko','demograficke udaje - pohlavi - Cesko'[Pohlavi]="Ženy")

// ZDE KONCI FILTRY DEFINOVANE V RAMCI FUNKCE CALCULATE()

)

Kalkulované míry

```
_1 AVERAGEX Pocet obyvatel =
```

```
/*
```

Nize uvedený výpočet pracuje s tabulkou 'základní ukazatele', kterou je iterováno (funkce typu AVERAGEX(), SUMX() apod. jsou iteratory). Protože není nijak definován filtr, do výpočtu vstupuje celá tabulka, výpočet však respektuje kontext výpočtu dany ostatními vizuály a filtry.

```
*/
```

```
AVERAGEX('základní ukazatele - Česko', // prvním argumentem je tabulka bez určených filtrů, nad jejímž sloupcem dochází k výpočtu
```

```
    'základní ukazatele - Česko'[Pocet obyvatel] // druhým argumentem je sloupec, nad kterým dochází k výpočtu  
)
```

_2 AVERAGEX FILTER Pocet obyvatel =

```
/*
Nize uvedený výpočet pracuje s tabulkou 'základní ukazatele', kterou je iterováno (funkce typu AVERAGEX, SUMX apod. jsou iteratory).
Protože není nijak definován filtr, do výpočtu vstupuje celá tabulka, výpočet však respektuje kontext výpočtu dany ostatními vizuály a filtry.
Pokud uživatel zvolí ve vizuálu filtr nad sloupcem, jeho filtr definuje i funkce FILTER(), tak návratovou hodnotou je tabulka splňující
jak filtr vizuálu, tak filtr v rámci výpočtu. Tj. Pokud si uživatel zvolí ve vizuálu období 2015-2017, výsledná hodnota počítá data
pouze za rok 2017, který je průnikem filtru {2015,2016,2017} a {2017}
*/
```

```
AVERAGEX(
    // prvním argumentem je tabulka bez určených filtru, nad jejímž sloupcem dochází k výpočtu
    FILTER('základní ukazatele - Česko','základní ukazatele - Česko'[Rok] = 2017),
    // druhým argumentem je sloupec, nad kterým dochází k výpočtu
    'základní ukazatele - Česko'[Pocet obyvatel]
)
```

_3 AVERAGEX ALL Pocet obyvatel =

```
/*
Nize uvedený výpočet pracuje s tabulkou 'základní ukazatele', kterou je iterováno (funkce typu AVERAGEX, SUMX apod. jsou iteratory).
Protože není nijak definován filtr, do výpočtu vstupuje celá tabulka, výpočet však respektuje kontext výpočtu dany ostatními vizuály a filtry.
*/
```

```
AVERAGEX(
    ALL('základní ukazatele - Česko'), // prvním argumentem je tabulka bez určených filtru, nad jejímž sloupcem dochází k výpočtu
    'základní ukazatele - Česko'[Pocet obyvatel] // druhým argumentem je sloupec, nad kterým dochází k výpočtu
)
```

Kalkulované míry – úpravy kontextu výpočtu

_1. pocet obyvatel bez upravy filtru =

/*
Definice filtru je nepovinnou casti funkce CALCULATE(), pokud neni ve vypoctu upraveno,
jak ma funkce CALCULATE() upravit kontext vypoctu, prevezme vsechny aplikovane filtry.

*/

```
CALCULATE(  
    SUM('zakladni ukazatele'[Pocet obyvatel])  
)
```

_2. pocet obyvatel vseh zakladni ukazatele zeme ve vybranem roce =

```
CALCULATE(  
    SUM('zakladni ukazatele'[Pocet obyvatel]), // vypocet nad tabulkou vracenou v druhe casti funkce  
  
    /*  
    Funkce ALL() v tomto pripade vraci vsechny hodnoty, tj. "rusi" filtry nad sloupcem 'zakladni ukazatele'[Zeme].  
    Funkce ALL() v tomto pripade odstranjuje pouze prime filtry nad sloupcem, tj. pokud je sloupec tzv. cross-filtrovan,  
    tyto cross-filtry jsou aplikovany.  
    Filtry nad jinymi sloupci jsou prebirany.  
    */  
    ALL('zakladni ukazatele'[Zeme])  
)
```

_3. pocet obyvatel vseh zeme ve vybranem roce =

```
CALCULATE(  
    SUM('zakladni ukazatele'[Pocet obyvatel]), // vypocet nad tabulkou vracenou v druhe casti funkce  
  
    /*  
    Funkce ALL() v tomto pripade vraci vsechny hodnoty, tj. "rusi" filtry nad celou tabulkou 'zakladni ukazatele',  
    a to jak prime filtry, tak tzv. cross-filtry.  
    */  
    ALL('zakladni ukazatele')  
)
```