

Contents

Referência do DAX (Data Analysis Expressions)

Aprender

[Visão Geral do DAX](#)

[Vídeos](#)

Funções do DAX

[Referência da função do DAX](#)

[Novas funções do DAX](#)

Funções de data e hora

[Visão Geral das funções de data e hora](#)

[CALENDAR](#)

[CALENDARAUTO](#)

[DATE](#)

[DATEDIFF](#)

[DATEVALUE](#)

[DAY](#)

[EDATE](#)

[EOMONTH](#)

[HOUR](#)

[MINUTE](#)

[MÊS](#)

[NOW](#)

[QUARTER](#)

[SECOND](#)

[TIME](#)

[TIMEVALUE](#)

[TODAY](#)

[UTCNOW](#)

[UTCTODAY](#)

[WEEKDAY](#)

WEEKNUM

YEAR

YEARFRAC

Funções de inteligência de dados temporais

Visão Geral das funções de inteligência de dados temporais

CLOSINGBALANCEMONTH

CLOSINGBALANCEQUARTER

CLOSINGBALANCEYEAR

DATEADD

DATESBETWEEN

DATESINPERIOD

DATESMTD

DATESQTD

DATESYTD

ENDOFMONTH

ENDOFQUARTER

ENDOFYEAR

FIRSTDATE

FIRSTNONBLANK

FIRSTNONBLANKVALUE

LASTDATE

LASTNONBLANK

LASTNONBLANKVALUE

NEXTDAY

NEXTMONTH

NEXTQUARTER

NEXTYEAR

OPENINGBALANCEMONTH

OPENINGBALANCEQUARTER

OPENINGBALANCEYEAR

PARALLELPERIOD

PREVIOUSDAY

PREVIOUSMONTH
PREVIOUSQUARTER
PREVIOUSYEAR
SAMEPERIODLASTYEAR
STARTOFMONTH
STARTOFQUARTER
STARTOFYEAR
TOTALMTD
TOTALQTD
TOTALYTD

Funções de filtro

Visão Geral das funções de filtro

ADDMISSINGITEMS
ALL
ALLCROSSFILTERED
ALLEXCEPT
ALLNOBLANKROW
ALLSELECTED
CALCULAR
CALCULATETABLE
CROSSFILTER
DISTINCT (coluna)
DISTINCT (tabela)
EARLIER
EARLIEST
FILTER
FILTERS
HASONEFILTER
HASONEVALUE
ISCROSSFILTERED
ISFILTERED
KEEPFILTERS

RELATED

RELATEDTABLE

REMOVEFILTERS

SELECTEDVALUE

SUBSTITUTEWITHINDEX

USERRELATIONSHIP

VALUES

Funções de informações

Visão Geral das funções de informações

CONTAINS

CUSTOMDATA

Operador IN / função CONTAINSROW

ISBLANK

ISERROR

ISEVEN

ISINSCOPE

ISLOGICAL

ISNONTEXT

ISNUMBER

ISODD

ISONORAFTER

ISTEXT

LOOKUPVALUE

USERNAME

USEROBJECTID

USERPRINCIPALNAME

Funções lógicas

Visão Geral das funções lógicas

AND

COALESCE

\FALSE

IF

IF.EAGER

IFERROR

IN

NOT

OR

SWITCH

\TRUE

Funções matemáticas e trigonométricas

Visão Geral das funções matemáticas e trigonométricas

ABS

ACOS

ACOSH

ACOT

ACOTH

ASIN

ASINH

ATAN

ATANH

CEILING

COMBIN

COMBINA

COS

COSH

CURRENCY

DEGREES

DIVIDE

EVEN

EXP

FACT

FLOOR

GCD

INT

ISO.CEILING

LCM

LN

LOG

LOG10

MOD

MROUND

ODD

PI

POWER

PRODUTO

PRODUCTX

QUOTIENT

RADIANS

RAND

RANDBETWEEN

ROUND

ROUNDDOWN

ROUNDUP

SIGN

SQRT

SUM

SUMX

TRUNC

Outras funções

Visão Geral das outras funções

CONVERT

DATATABLE

ERRO

EXCEPT

GENERATESERIES

GROUPBY

INTERSECT

ISEMPTY

ISSELECTEDMEASURE

NATURALINNERJOIN

NATURALLEFTOUTERJOIN

SELECTEDMEASURE

SELECTEDMEASUREFORMATSTRING

SELECTEDMEASURENAME

SUMMARIZECOLUMNS

Construtor de tabela

TREATAS

UNION

Funções pai e filho

Visão Geral das funções pai e filho

Compreensão das funções para hierarquias pai-filho

PATH

PATHCONTAINS

PATHITEM

PATHITEMREVERSE

PATHLENGTH

Funções estatísticas

Visão Geral das funções estatísticas

ADDCOLUMNS

APPROXIMATEDISTINCTCOUNT

AVERAGE

AVERAGEA

AVERAGEX

BETA.DIST

BETA.INV

CHISQ.DIST

CHISQ.DIST.RT

CHISQ.INV

CHISQ.INV.RT
CONFIDENCE.NORM
CONFIDENCE.T
COT
COTH
CONTAGEM
COUNTA
COUNTAX
COUNTBLANK
COUNTROWS
COUNTX
CROSSJOIN
DISTINCTCOUNT
EXPON.DIST
GENERATE
GENERATEALL
GEOMEAN
GEOMEANX
MAX
MAXA
MAXX
MEDIAN
MEDIANX
MIN
MINA
MINX
NORM.DIST
NORM.INV
NORM.S.DIST
NORM.S.INV
PERCENTILE.EXC
PERCENTILE.INC

PERCENTILEX.EXC

PERCENTILEX.INC

PERMUT

POISSON.DIST

RANK.EQ

RANKX

ROW

SAMPLE

SELECTCOLUMNS

SIN

SINH

STDEV.S

STDEV.P

STDEVX.S

STDEVX.P

SQRTPI

SUMMARIZE

T.DIST

T.DIST.2T

T.DIST.RT

T.INV

T.INV.2T

TAN

TANH

TOPN

VAR.S

VAR.P

VARX.S

VARX.P

XIRR

XNPV

Funções de texto

Visão Geral das funções de texto

BLANK

CODE

COMBINEVALUES

CONCATENATE

CONCATENATEX

EXACT

FIND

FIXED

FORMAT

FORMAT

Formatos numéricos predefinidos para FORMAT

Formatos numéricos personalizados para FORMAT

Formatos de data e hora predefinidos para FORMAT

Formatos de data e hora personalizados para FORMAT

LEFT

LEN

LOWER

MID

REPLACE

REPT

RIGHT

SEARCH

SUBSTITUTE

TRIM

UNICHAR

UNICODE

UPPER

VALUE

Instruções DAX

Visão geral das instruções

DEFINIR

AVALIAR

ORDER BY

VAR

Sintaxe do DAX

Operadores DAX

Consultas do DAX

Nomenclatura de parâmetro do DAX

Visão geral do DAX

12/05/2020 • 63 minutes to read

O DAX (Data Analysis Expressions) é uma linguagem de expressão de fórmula usada nos Analysis Services, no Power BI Desktop e no Power Pivot no Excel. As fórmulas DAX incluem funções, operadores e valores para realizar cálculos avançados e consultas em dados nas tabelas e colunas relacionadas nos modelos de dados tabulares.

Este artigo fornece apenas uma introdução básica aos conceitos mais importantes no DAX. Ele descreve o DAX, pois se aplica a todos os produtos que o usam. Algumas funcionalidades podem não se aplicar a determinados produtos ou casos de uso. Confira a documentação do seu produto que descreve a implementação específica do DAX.

Cálculos

As fórmulas de cálculo DAX são usadas em medidas, colunas calculadas, tabelas calculadas e filtros de linha.

Medidas

As medidas são fórmulas de cálculo dinâmico em que os resultados mudam dependendo do contexto. As medidas são usadas em relatórios que dão suporte à combinação e à filtragem dos dados de modelo usando vários atributos, como um relatório do Power BI ou a Tabela Dinâmica ou o Gráfico Dinâmico do Excel. As medidas são criadas usando a barra de fórmulas DAX no designer do modelo.

Uma fórmula em uma medida pode usar as funções de agregação padrão criadas automaticamente usando o recurso Autossoma, como COUNT ou SUM, ou você pode definir sua própria fórmula usando a barra de fórmulas DAX. Medidas nomeadas podem ser passadas como um argumento para outras medidas.

Quando você define uma fórmula para uma medida na barra de fórmulas, um recurso Dica de Ferramentas mostra uma visualização de quais seriam os resultados para o total no contexto atual, mas, caso contrário, os resultados não são imediatamente transmitidos em qualquer lugar. O motivo pelo qual você não pode ver os resultados (filtrados) do cálculo imediatamente é porque o resultado de uma medida não pode ser determinado sem contexto. Para avaliar uma medida, é necessário um aplicativo cliente de relatório que pode fornecer o contexto necessário para recuperar os dados relevantes para cada célula e avaliar a expressão para cada célula. Esse cliente pode ser uma Tabela Dinâmica ou um Gráfico Dinâmico do Excel, um relatório do Power BI ou uma expressão de tabela em uma consulta DAX no SSMS (SQL Server Management Studio).

Independentemente do cliente, uma consulta separada é executada para cada célula nos resultados. Isso significa que cada combinação de cabeçalhos de linha e de coluna em uma Tabela Dinâmica ou cada seleção de segmentações e filtros em um relatório do Power BI gera um subconjunto diferente de dados sobre o qual a medida é calculada. Por exemplo, usando esta fórmula de medida muito simples:

```
Total Sales:=SUM([Sales Amount])
```

Quando um usuário coloca a medida TotalSales na janela Valores em uma Tabela Dinâmica e a coluna Categoria do Produto de uma tabela Produto na janela Filtros, a soma do Valor de Vendas é calculado e exibido para cada categoria de produtos.

Diferentemente das colunas calculadas e filtros de linha, a sintaxe de uma medida inclui o nome da medida que antecede a fórmula. No exemplo que acabou de ser fornecido, o nome **Total de Vendas** é exibido antes da fórmula. Após criar uma medida, o nome e sua definição serão exibidos na Lista de Campo do aplicativo cliente de relatório e dependendo se as perspectivas e funções estarão disponíveis a todos os usuários do modelo.

Para saber mais, confira:

[Medidas no Power BI Desktop](#)

[Medidas no Analysis Services](#)

[Medidas no Power Pivot](#)

Colunas calculadas

Uma coluna calculada é uma coluna que você adiciona a uma tabela existente (no designer de modelo) e cria uma fórmula DAX que define os valores da coluna. Como uma coluna calculada é criada em uma tabela no modelo de dados, não há suporte a ela em modelos que recuperam dados exclusivamente de uma fonte de dados relacional que usa o modo DirectQuery.

Quando uma coluna calculada contiver uma fórmula DAX válida, os valores serão calculados para cada coluna assim que a fórmula for inserida. Os valores serão armazenados no modelo de dados na memória. Por exemplo, em uma tabela Data, quando a fórmula é inserida na barra de fórmulas:

```
=[Calendar Year] & " Q" & [Calendar Quarter]
```

Um valor para cada linha na tabela é calculado inserindo valores da coluna Ano Calendário (na mesma Tabela de data), adicionando um espaço e a letra Q maiúscula e adicionando os valores da coluna Trimestre Calendário (na mesma Tabela de data). O resultado de cada linha na coluna calculada é calculado imediatamente e é exibido, por exemplo, como **T1 2017**. Os valores de coluna apenas serão recalculados se a tabela ou qualquer tabela relacionada for processada (atualização) ou o modelo for descarregado da memória e recarregado, como ocorre ao fechar e reabrir um arquivo do Power BI Desktop.

Para saber mais, confira:

[Colunas calculadas no Power BI Desktop](#)

[Colunas calculadas no Analysis Services](#)

[Colunas calculadas no Power Pivot.](#)

Tabelas calculadas

Uma tabela calculada é um objeto computado com base em uma consulta ou expressão de fórmula DAX, derivada de todas ou de parte de outras tabelas no mesmo modelo. Em vez de consultar e carregar valores nas colunas da sua nova tabela de uma fonte de dados, uma fórmula DAX define os valores da tabela.

As tabelas calculadas podem ser úteis em uma dimensão com função múltipla. Um exemplo é a Tabela de data, como OrderDate, ShipDate ou DueDate, dependendo da relação de chave estrangeira. Ao criar uma tabela calculada para ShipDate explicitamente, você obtém uma tabela autônoma disponível para consultas como sendo totalmente operável como qualquer outra tabela. Tabelas calculadas também são úteis ao configurar um conjunto de linhas filtrado ou um subconjunto ou superconjunto de colunas de outras tabelas existentes. Com isso, você pode manter a tabela original intacta ao criar variações dessa tabela para dar suporte a cenários específicos.

Tabelas calculadas dão suporte a relações com outras tabelas. As colunas na tabela calculada têm tipos de dados, formatação e podem pertencer a uma categoria de dados. Tabelas calculadas podem ser nomeadas e exibidas ou ocultas assim como qualquer outra tabela. Tabelas calculadas serão recalculadas se qualquer uma das tabelas das quais elas recebem dados por pull forem renovadas ou atualizadas.

Para saber mais, confira:

[Tabelas calculadas no Power BI Desktop](#)

[Tabelas calculadas no Analysis Services.](#)

Filtros de linha (segurança em nível de linha)

Em filtros de linha, também conhecidos como Segurança em nível de linha, uma fórmula DAX deve ser avaliada como uma condição booleana TRUE/FALSE, definindo quais linhas podem ser retornadas pelos resultados de uma consulta por membros de uma função específica. Por exemplo, para membros da função Vendas, a Tabela de

clientes com a seguinte fórmula DAX:

```
=Customers[Country] = "USA"
```

Os membros da função Vendas só poderão exibir dados para clientes nos EUA e agregações, como SUM, são retornadas apenas para clientes nos EUA. Os filtros de linha não estão disponíveis no Power Pivot no Excel.

Ao definir um filtro de linha usando a fórmula DAX, você criará um conjunto de linhas permitido. Isso não nega o acesso a outras linhas; em vez disso, elas simplesmente não são retornadas como parte do conjunto de linhas permitido. Outras funções podem permitir o acesso às linhas excluídas pela fórmula DAX. Se um usuário for membro de outra função e os filtros de linha dessa função permitirem acesso a esse conjunto de linhas específico, o usuário poderá exibir os dados dessa linha.

Os filtros de linha se aplicam às linhas especificadas assim como linhas relacionadas. Quando uma tabela tem várias relações, os filtros aplicam segurança para a relação ativa. Os filtros de linha serão interseccionados com outros filtros de linha definidos para tabelas relacionadas.

Para saber mais, confira:

[Segurança no nível da linha \(RLS\) com Power BI](#)

[Filtros de linha no Analysis Services](#)

Consultas

As consultas DAX podem ser criadas e executadas no SSMS (SQL Server Management Studio) e em ferramentas de software livre, como o DAX Studio (daxstudio.org). Ao contrário das fórmulas de cálculo DAX, que só podem ser criadas em modelos de dados tabulares, as consultas DAX também podem ser executadas em modelos multidimensionais do Analysis Services. As consultas DAX geralmente são mais fáceis de serem escritas e mais eficientes do que as consultas MDX (Multidimensional Data Expressions).

Uma consulta DAX é uma instrução, semelhante a uma instrução SELECT no T-SQL. O tipo mais básico de consulta DAX é uma instrução *evaluate*. Por exemplo,

```
EVALUATE  
( FILTER ( 'DimProduct', [SafetyStockLevel] < 200 ) )  
ORDER BY [EnglishProductName] ASC
```

Retorna nos resultados uma tabela que lista apenas esses produtos com um SafetyStockLevel menor do que 200, em ordem crescente por EnglishProductName.

Você pode criar medidas como parte da consulta. As medidas existem apenas durante a consulta. Para saber mais, confira [Consultas DAX](#).

Fórmulas

As fórmulas DAX são essenciais para criar cálculos em colunas e medidas calculadas, além de proteger seus dados usando filtros de nível de linha. Para criar fórmulas para colunas e medidas calculadas, você usará a barra de fórmulas ao longo da parte superior da janela do designer de modelos ou do Editor DAX. Para criar fórmulas para filtros de linha, você usará a caixa de diálogo Gerenciador de Funções. As informações nesta seção devem introduzir você aos conceitos básicos de fórmulas DAX.

Conceitos básicos de fórmula

As fórmulas DAX podem ser muito simples ou muito complexas. A tabela a seguir mostra alguns exemplos de fórmulas simples que poderiam ser usadas em uma coluna calculada.

Fórmula	Descrição
<code>=TODAY()</code>	Insere a data de hoje em cada linha de uma coluna calculada.
<code>=3</code>	Insere o valor 3 em cada linha de uma coluna calculada.
<code>=[Column1] + [Column2]</code>	Adiciona os valores na mesma linha de [Coluna1] e [Coluna2] e coloca os resultados na coluna calculada da mesma linha.

Se a fórmula criada for simples ou complexa, você poderá usar as seguintes etapas ao criar uma fórmula:

1. Cada fórmula deve começar com um sinal de igual (=).
2. Você pode digitar ou selecionar um nome de função ou digitar uma expressão.
3. Comece digitando as primeiras letras da função ou dando o nome desejado. AutoComplete exibe uma lista de funções, tabelas e colunas disponíveis. Pressione TAB para adicionar um item da lista AutoComplete à fórmula.

Você também pode clicar no botão **Fx** para exibir uma lista de funções disponíveis. Para selecionar uma função na lista suspensa, use as teclas de seta para realçar o item e clique em **OK** para adicionar a função à fórmula.
4. Forneça os argumentos para a função selecionando-os em uma lista suspensa de possíveis tabelas e colunas ou digitando valores.
5. Veja se há erros de sintaxe: verifique se todos os parênteses estão fechados e se as colunas, as tabelas e os valores estão referenciados corretamente.
6. Pressione ENTER para aceitar a fórmula.

NOTE

Em uma coluna calculada, assim que você inserir a fórmula e ela for validada, a coluna será populada com valores. Em uma medida, pressionar ENTER salva a definição da medida com a tabela. Se uma fórmula estiver inválida, um erro será exibido.

Neste exemplo, vamos examinar uma fórmula em uma medida chamada **Dias no Trimestre Atual**:

```
Days in Current Quarter = COUNTROWS( DATESBETWEEN( 'Date'[Date], STARTOFQUARTER( LASTDATE('Date'[Date])),
ENDOFQUARTER('Date'[Date])) ) )
```

essa medida é usada para criar uma taxa de comparação entre um período incompleto e o período anterior. A fórmula deve levar em conta a proporção do período decorrido e compará-lo com a mesma proporção no período anterior. Nesse caso, $\frac{[\text{Dias desde o Início do Trimestre Atual}]}{[\text{Dias no Trimestre Atual}]}$ fornece a proporção decorrida no período atual.

Esta fórmula contém os seguintes elementos:

ELEMENTO DA FÓRMULA	DESCRIÇÃO
<code>Days in Current Quarter</code>	O nome da medida.
<code>=</code>	O sinal de igual (=) inicia a fórmula.

ELEMENTO DA FÓRMULA	DESCRIÇÃO
COUNTROWS	COUNTROWS conta o número de linhas na Tabela de data
()	Os parênteses de abertura e fechamento especifica os argumentos.
DATESBETWEEN	A função DATESBETWEEN retorna as datas entre a última data para cada valor na coluna Data da Tabela de data.
'Date'	Especifica a tabela Date. As tabelas estão entre aspas simples.
[Date]	Especifica a coluna Date na tabela Date. As colunas estão entre colchetes.
,	
STARTOFQUARTER	A função STARTOFQUARTER retorna a data do início do trimestre.
LASTDATE	A função LASTDATE retorna a última data do trimestre.
'Date'	Especifica a tabela Date.
[Date]	Especifica a coluna Date na tabela Date.
,	
ENDOFQUARTER	A função ENDOFQUARTER
'Date'	Especifica a tabela Date.
[Date]	Especifica a coluna Date na tabela Date.

Usar a fórmula AutoComplete

A barra de fórmulas no designer de modelos e a janela Filtros de Linha da fórmula na caixa de diálogo Gerenciador de Funções fornecem um recurso AutoComplete. O recurso AutoComplete ajuda você a inserir uma sintaxe de fórmula válida fornecendo opções para cada elemento na fórmula.

- Você pode usar a fórmula AutoComplete no meio de uma fórmula existente com funções aninhadas. O texto imediatamente antes do ponto de inserção é usado para exibir valores na lista suspensa e todo o texto após o ponto de inserção permanece inalterado.
- AutoComplete não adiciona o parêntese de fechamento de funções nem faz a correspondência automática deles. Você deve verificar se cada fórmula está sintaticamente correta ou você não pode salvar ou usar a fórmula.

Usar várias funções em uma fórmula

Você pode aninhar funções, o que significa que você usa os resultados de uma função como um argumento de outra função. Você pode aninhar até 64 níveis de funções em colunas calculadas. No entanto, o aninhamento pode dificultar a criação ou a solução de problemas de fórmulas. Muitas funções são criadas para serem usadas exclusivamente como funções aninhadas. Essas funções retornam uma tabela, que não pode ser salva diretamente como um resultado; ele deve ser fornecido como entrada para uma função de tabela. Por exemplo, as funções SUMX, AVERAGEX e MINX requerem uma tabela como o primeiro argumento.

Funções

Uma função é uma fórmula nomeada dentro de uma expressão. A maioria das funções tem argumentos obrigatórios e opcionais, também chamados de parâmetros, como entrada. Quando a função é executada, um valor é retornado. O DAX inclui funções que podem ser usadas para executar cálculos usando datas e horas, criar valores condicionais, trabalhar com cadeias de caracteres, executar pesquisas com base em relações, além de incluir a capacidade de iterar em uma tabela para executar cálculos recursivos. Se você já conhecer as fórmulas do Excel, muitas dessas funções parecerão muito semelhantes; no entanto, as fórmulas DAX são diferentes das seguintes maneiras importantes:

- Uma função DAX sempre referencia uma coluna ou uma tabela completa. Se desejar usar apenas valores específicos de uma tabela ou coluna, é possível adicionar filtros à fórmula.
- Se precisar personalizar cálculos linha por linha, o DAX fornecerá funções que permitirão que você use o valor da linha atual ou um valor relacionado como um tipo de parâmetro para realizar cálculos que variam de acordo com o contexto. Para entender como essas funções funcionam, confira [Contexto](#) neste artigo.
- O DAX inclui muitas funções que retornam uma tabela em vez de um valor. A tabela não é exibida em um cliente de relatório, mas é usada para fornecer entrada para outras funções. Por exemplo, é possível recuperar uma tabela e, em seguida, contar os valores distintos contidos nela, ou calcular somas dinâmicas em diferentes colunas ou tabelas filtradas.
- As funções DAX incluem uma variedade de funções de *inteligência de dados temporais*. Essas funções permitem que você defina ou selecione intervalos de data e execute cálculos dinâmicos com base nessas datas ou nesse intervalo. Por exemplo, é possível comparar somas em períodos paralelos.

Funções de data e hora

As funções de data e hora no DAX são semelhantes a funções de data e hora no Microsoft Excel. No entanto, as funções DAX são baseadas nos tipos de dados **datetime** usados pelo Microsoft SQL Server. Para obter mais informações, confira [Funções de data e hora](#).

Funções de filtro

As funções de filtro no DAX retornam tipos de dados específicos, pesquisam valores em tabelas relacionadas e filtram por valores relacionados. As funções de pesquisa funcionam usando tabelas e relações, como um banco de dados. As funções de filtragem permitem que você manipule o contexto de dados para criar cálculos dinâmicos. Para obter mais informações, confira [Funções de filtro](#).

Funções de informações

Uma função de informação examina a célula ou a linha fornecida como um argumento e informa se o valor corresponde ao tipo esperado. Por exemplo, a função ISERROR retornará TRUE se o valor referenciado contiver um erro. Para obter mais informações, confira [Funções de informação](#).

Funções lógicas

As funções lógicas agem em uma expressão para retornar informações sobre os valores dela. Por exemplo, a função TRUE permite que você saiba se uma expressão que está sendo avaliada retorna um valor TRUE. Para obter mais informações, confira [Funções lógicas](#).

Funções matemáticas e trigonométricas

As funções matemáticas e trigonométricas no DAX são muito semelhantes às funções matemáticas e trigonométricas do Excel. Existem algumas pequenas diferenças nos tipos de dados numéricos usados por funções DAX. Para obter mais informações, confira [Funções matemáticas e trigonométricas](#).

Outras funções

Essas funções executam ações exclusivas que não podem ser definidas por nenhuma das categorias às quais a maioria das outras funções pertence. Para obter mais informações, confira [Outras funções](#).

Funções estatísticas

O DAX fornece funções estatísticas que executam agregações. Além de criar somas e médias ou de encontrar os valores mínimo e máximo, no DAX também é possível filtrar uma coluna antes de agregar ou criar agregações com base em tabelas relacionadas. Para obter mais informações, confira [Funções estatísticas](#).

Funções de texto

As funções de texto no DAX são muito semelhantes às suas equivalentes no Excel. Você pode retornar parte de uma cadeia de caracteres, pesquisar um texto em uma cadeia de caracteres ou concatenar valores de cadeia de caracteres. O DAX também fornece funções para controlar os formatos de datas, horas e números. Para obter mais informações, confira [Funções de texto](#).

Funções de inteligência de dados temporais

Com as funções de inteligência de dados temporais fornecidas no DAX, você pode criar cálculos que usam conhecimento interno sobre calendários e datas. Usando intervalos de data e hora em combinação com agregações ou cálculos, você pode criar comparações significativas em períodos de tempo comparáveis para vendas, estoque etc. Para obter mais informações, confira [Funções de inteligência de dados temporais \(DAX\)](#).

Funções com valor de tabela

Há funções DAX que geram tabelas, usam tabelas como entrada ou fazem ambos. Como uma tabela pode ter uma única coluna, as funções com valor de tabela também usam colunas únicas como entradas. Entender como usar as funções com valor de tabela é importante para utilizar completamente fórmulas DAX. O DAX inclui os seguintes tipos de funções com valor de tabela:

- Funções de filtro retornam uma coluna, tabela ou valores relacionados à linha atual.
- Funções de agregação agregam qualquer expressão ao longo das linhas de uma tabela.
- As funções de inteligência de dados temporais retornam uma tabela de datas ou usam uma tabela de datas para calcular uma agregação.

Os exemplos de funções de tabela incluem: FILTER, ALL, VALUES, DISTINCT, RELATEDTABLE.

Variáveis

Você pode criar variáveis dentro de uma expressão usando o [VAR](#). Tecnicamente, o VAR não é uma função; é uma palavra-chave usada para armazenar o resultado de uma expressão como uma variável nomeada. Essa variável pode ser passada como um argumento para outras expressões de medida. Por exemplo:

```
VAR
    TotalQty = SUM ( Sales[Quantity] )

Return
    IF (
        TotalQty > 1000,
        TotalQty * 0.95,
        TotalQty * 1.25
    )
```

Neste exemplo, TotalQty pode ser passado como uma variável nomeada para outras expressões. As variáveis podem ser de qualquer tipo de dados escalares, incluindo tabelas. O uso de variáveis em suas fórmulas DAX pode ser incrivelmente eficiente.

Tipos de dados

Você pode importar dados em um modelo de muitas fontes de dados diferentes que podem dar suporte a diferentes tipos de dados. Quando você importa dados para um modelo, eles são convertidos em um dos tipos de

dados de modelo tabular. Quando os dados do modelo são usados em um cálculo, eles são convertidos em um tipo de dados DAX para a duração e a saída do cálculo. Quando você cria uma fórmula DAX, os termos usados nela determinarão automaticamente o tipo de dados de valor retornado.

O DAX dá suporte aos seguintes tipos de dados:

TIPO DE DADOS NO MODELO	TIPO DE DADOS NO DAX	DESCRIÇÃO
Número Inteiro	Um valor inteiro de 64 bits (oito bytes) 1, 2	Números que não têm casas decimais. Inteiros podem ser números positivos ou negativos, mas devem ser números inteiros entre - 9.223.372.036.854.775.808 (-2^{63}) e 9.223.372.036.854.775.807 ($2^{63}-1$).
Número Decimal	Um número real de 64 bits (oito bytes) 1, 2	Números reais são números que podem ter casas decimais. Os números reais abrangem uma grande variedade de valores: Valores negativos de -1,79E +308 a - 2,23E -308 Zero Valores positivos de 2,23E -308 a 1,79E + 308 No entanto, o número de dígitos significativos é limitado a 17 dígitos decimais.
Booleano	Booleano	Um valor True ou False.
Texto	Cadeia de caracteres	Uma cadeia de dados de caractere Unicode. Pode ser composto de cadeias de caracteres, números ou datas representados em um formato de texto.
Data	Date/time	Datas e horas em uma representação de data e hora aceita. As datas válidas são todas as datas após 1º de março de 1900.
Moeda	Moeda	O tipo de dados de moeda permite valores entre - 922.337.203.685.477,5808 e 922.337.203.685.477,5807 com quatro dígitos decimais de precisão fixa.
N/D	Em Branco	Um espaço em branco é um tipo de dados no DAX que representa e substitui nulos SQL. Você pode criar um espaço em branco usando a função BLANK e testar se há espaços em branco usando a função lógica ISBLANK.

Modelos de dados tabulares também incluem o tipo de dados de *tabela* como a entrada ou a saída para muitas

funções DAX. Por exemplo, a função FILTER usa uma tabela como entrada e gera uma outra tabela que contém apenas as linhas que atendem às condições de filtro. Ao combinar funções de tabela com funções de agregação, você pode executar cálculos complexos em conjuntos de dados definidos dinamicamente.

Embora os tipos de dados normalmente sejam definidos automaticamente, é importante entender os tipos de dados e como eles se aplicam, em particular, a fórmulas DAX. Erros em fórmulas ou resultados inesperados, por exemplo, geralmente são causados usando um operador específico que não pode ser usado com um tipo de dados especificado em um argumento. Por exemplo, a fórmula, `= 1 & 2`, retorna um resultado de cadeia de caracteres 12. No entanto, a fórmula, `= "1" + "2"`, retorna um resultado inteiro 3.

Context

Contexto é um conceito importante que deve ser entendido ao criar fórmulas DAX. O contexto é o que permite que você execute uma análise dinâmica conforme os resultados de uma fórmula são alterados para refletir a seleção atual de linha ou de célula e dados relacionados. Entender o contexto e usá-lo com eficiência é crítico para criar análises dinâmicas de alto desempenho e para solucionar problemas em fórmulas.

As fórmulas em modelos tabulares podem ser avaliadas em um contexto diferente, dependendo de outros elementos de design:

- Filtros aplicados em uma Tabela Dinâmica ou um relatório
- Filtros definidos dentro de uma fórmula
- Relações especificadas usando funções especiais dentro de uma fórmula

Há diferentes tipos de contexto: *contexto de linha*, *contexto de consulta* e *contexto de filtro*.

Contexto de linha

O *contexto de linha* pode ser considerado "a linha atual". Se você criar uma fórmula em uma coluna calculada, o contexto de linha dessa fórmula incluirá os valores de todas as colunas na linha atual. Se a tabela estiver relacionada a outra tabela, o conteúdo também incluirá todos os valores da outra tabela relacionados à linha atual.

Por exemplo, vamos supor que você cria uma coluna calculada, `= [Freight] + [Tax]`, que adiciona valores de duas colunas, Frete e Imposto, da mesma tabela. Essa fórmula obtém automaticamente somente os valores da linha atual nas colunas especificadas.

O contexto de linha também segue relações definidas entre tabelas, incluindo aquelas definidas dentro de uma coluna calculada usando fórmulas DAX para determinar quais linhas em tabelas relacionadas estão associadas à linha atual.

Por exemplo, a fórmula a seguir usa a função RELATED para buscar um valor de imposto de uma tabela relacionada, com base na região à qual o pedido foi enviado. O valor do imposto é determinado usando o valor da região na tabela atual, pesquisando a região na tabela relacionada e obtendo a taxa de imposto para essa região da tabela relacionada.

```
= [Freight] + RELATED('Region'[TaxRate])
```

Essa fórmula obtém a taxa de imposto para a região atual da tabela Região e a adiciona ao valor da coluna Frete. Em fórmulas DAX, você não precisa saber ou especificar a relação específica que conecta as tabelas.

Contexto de várias linhas

O DAX inclui funções que iteram cálculos em uma tabela. Essas funções podem ter várias linhas atuais, cada uma com seu próprio contexto de linha. Em essência, essas funções permitem que você crie fórmulas que executam cálculos recursivamente em um loop interno e externo.

Por exemplo, vamos supor que seu modelo contenha uma tabela **Produtos** e uma tabela **Vendas**. Os usuários talvez queiram percorrer toda a tabela de vendas, que está cheia de transações que envolvem vários produtos, e

encontrar a maior quantidade pedida de cada produto em qualquer transação.

Com o DAX, você pode criar uma única fórmula que retorna o valor correto e os resultados serão atualizados automaticamente sempre que um usuário adicionar dados às tabelas.

```
=MAXX(FILTER(Sales,[ProdKey]=EARLIER([ProdKey])),Sales[OrderQty])
```

Para obter um exemplo detalhado desta fórmula, confira [EARLIER](#).

Para resumir, a função EARLIER armazena o contexto de linha da operação que precedeu a operação atual. Em todos os momentos, a função armazena na memória dois conjuntos de contexto: um deles representa a linha atual do loop interno da fórmula e o outro representa a linha atual do loop externo da fórmula. O DAX alimenta automaticamente valores entre os dois loops para que você possa criar agregações complexas.

Contexto de consulta

Contexto de consulta refere-se ao subconjunto de dados recuperado implicitamente para uma fórmula. Quando um usuário coloca uma medida ou outro campo de valor em uma Tabela Dinâmica ou em um relatório baseado em um modelo tabular, o mecanismo examina os cabeçalhos de linha e coluna, a Segmentação de Dados e os filtros de relatório para determinar o contexto. Em seguida, as consultas necessárias são executadas na fonte de dados para obter o subconjunto correto de dados, fazer os cálculos definidos pela fórmula e, em seguida, preencher cada célula na Tabela Dinâmica ou no relatório. O conjunto de dados recuperado é o contexto de consulta para cada célula.

Como o contexto muda dependendo do local em que você coloca a fórmula, os resultados dela também podem mudar.

Por exemplo vamos supor que você crie uma fórmula que some os valores na coluna **Lucro** da tabela **Vendas**:

`=SUM('Sales'[Profit])`. Se você usar essa fórmula em uma coluna calculada na tabela **Vendas**, os resultados da fórmula serão os mesmos para toda a tabela, porque o contexto de consulta da fórmula sempre é todo o conjunto de dados da tabela **Vendas**. Os resultados terão lucro para todas as regiões, todos os produtos, todos os anos e assim por diante.

No entanto, os usuários normalmente não querem ver o mesmo resultado centenas de vezes; em vez disso, desejam obter o lucro de um ano específico, um país específico, um produto específico ou alguma combinação deles e, em seguida, obter um total geral.

Em uma Tabela Dinâmica, o contexto pode ser alterado adicionando ou removendo cabeçalhos de coluna e linha e adicionando ou removendo a Segmentação de Dados. Sempre que os usuários adicionarem cabeçalhos de coluna ou de linha à Tabela Dinâmica, eles mudarão o contexto de consulta no qual a medida é avaliada. As operações de segmentação e filtragem também afetam o contexto. Portanto, a mesma fórmula, usada em uma medida, é avaliada em um *contexto de consulta* diferente para cada célula.

Contexto de filtro

Contexto de filtro é o conjunto de valores permitido em cada coluna ou nos valores recuperados de uma tabela relacionada. Os filtros podem ser aplicados à coluna no designer ou na camada de apresentação (relatórios e Tabelas Dinâmicas). Os filtros também podem ser definidos explicitamente por expressões de filtro dentro da fórmula.

O contexto de filtro é adicionado quando você especifica restrições de filtro no conjunto de valores permitido em uma coluna ou tabela, usando argumentos para uma fórmula. O contexto de filtro é aplicado com base em outros contextos, como contexto de linha ou de consulta.

Em modelos tabulares, há várias maneiras de criar o contexto de filtro. Dentro do contexto de clientes que podem consumir o modelo, como relatórios do Power BI, os usuários podem criar filtros dinamicamente adicionando segmentações ou filtros de relatório nos cabeçalhos de linha e coluna. Você também pode especificar expressões de filtro diretamente dentro da fórmula para especificar valores relacionados, filtrar tabelas usadas como entradas

ou obter contexto dinamicamente para os valores usados em cálculos. Você também pode limpar completamente ou seletivamente os filtros em colunas específicas. Isso é muito útil ao criar fórmulas que calculam totais gerais.

Para obter mais informações sobre como criar filtros dentro de fórmulas, confira a [Função FILTER \(DAX\)](#).

Para obter um exemplo de como os filtros podem ser limpos para criar totais gerais, confira a [Função ALL \(DAX\)](#).

Para obter exemplos de como limpar seletivamente e aplicar filtros dentro de fórmulas, confira [ALLEXCEPT](#).

Determinar contexto em fórmulas

Quando você cria uma fórmula DAX, ela é testada primeiro quanto à validade da sintaxe e, em seguida, para verificar se os nomes das colunas e tabelas incluídas na fórmula podem ser encontrados no contexto atual. Se qualquer coluna ou tabela especificada pela fórmula não puder ser encontrada, será retornado um erro.

O contexto durante a validação (e operações de recálculo) é determinado conforme descrito nas seções anteriores, usando as tabelas disponíveis no modelo, relações entre as tabelas e filtros que foram aplicados.

Por exemplo, se você tiver acabado de importar alguns dados em uma nova tabela e eles não estiverem relacionados a nenhuma outra (e você não tiver aplicado nenhum filtro), o *contexto atual* será o conjunto inteiro de colunas na tabela. Se a tabela estiver vinculada por relações com outras tabelas, o contexto atual incluirá as tabelas relacionadas. Se você adicionar uma coluna da tabela a um relatório que tenha uma Segmentação de Dados e talvez alguns filtros de relatório, o contexto da fórmula será o subconjunto de dados em cada célula do relatório.

O contexto é um conceito avançado que também pode dificultar a solução de problemas de fórmula.

Recomendamos que você comece com fórmulas e relações simples para ver como o contexto funciona. A seção a seguir fornece alguns exemplos de como as fórmulas usam diferentes tipos de contexto para retornar resultados dinamicamente.

Operadores

A linguagem DAX usa quatro tipos diferentes de operadores de cálculo em fórmulas:

- Operadores de comparação para comparar valores e retornar um valor TRUE\FALSE lógico.
- Operadores aritméticos para executar cálculos aritméticos que retornam valores numéricos.
- Operadores de concatenação de texto para unir duas ou mais cadeias de caracteres de texto.
- Operadores lógicos que combinam duas ou mais expressões para retornar um único resultado.

Para obter informações detalhadas sobre operadores usados em fórmulas DAX, confira [Operadores DAX](#).

Trabalhar com tabelas e colunas

Tabelas em modelos de dados tabulares se parecem com tabelas do Excel, mas são diferentes na maneira como funcionam com os dados e com fórmulas:

- As fórmulas só funcionam com tabelas e colunas, não com células individuais, referências de intervalo ou matrizes.
- As fórmulas podem usar relações para obter valores de tabelas relacionadas. Os valores recuperados sempre são relacionados ao valor de linha atual.
- Você não pode ter dados irregulares ou desbalanceados como pode acontecer em uma planilha do Excel. Cada linha em uma tabela deve conter o mesmo número de colunas. No entanto, você pode ter valores vazios em algumas colunas. Tabelas de dados do Excel e tabelas de dados de modelo tabular não são intercambiáveis.
- Como um tipo de dados é definido para cada coluna, cada valor nela deve ser do mesmo tipo.

Referenciar tabelas e colunas em fórmulas

Você pode referenciar qualquer tabela e coluna usando seu nome. Por exemplo, a fórmula a seguir ilustra como referenciar colunas de duas tabelas usando o nome *totalmente qualificado*.

```
=SUM('New Sales'[Amount]) + SUM('Past Sales'[Amount])
```

Quando uma fórmula é avaliada, o designer de modelo primeiro verifica a sintaxe geral e, em seguida, verifica os nomes das colunas e das tabelas que você fornece em relação às colunas e tabelas possíveis no contexto atual. Se o nome for ambíguo ou se a coluna ou a tabela não puder ser encontrada, você receberá um erro na fórmula (uma cadeia de caracteres #ERROR em vez de um valor de dados nas células em que o erro ocorre). Para obter mais informações sobre requisitos de nomenclatura para tabelas, colunas e outros objetos, confira Requisitos de Nomenclatura na [sintaxe DAX](#).

Relações de tabela

Ao criar relações entre tabelas, você tem a capacidade de pesquisar dados em outra tabela e usar valores relacionados para executar cálculos complexos. Por exemplo, você pode usar uma coluna calculada para pesquisar todos os registros de envio relacionados ao revendedor atual e somá-los para cada um. Em muitos casos, no entanto, uma relação pode não ser necessária. Você pode usar a função [LOOKUPVALUE](#) em uma fórmula para retornar o valor em *result_columnName* para a linha que atende aos critérios especificados nos argumentos *search_column* e *search_value*.

Muitas funções DAX exigem que uma relação exista entre as tabelas ou entre várias tabelas, a fim de localizar as colunas que você referenciou e retornar os resultados que fazem sentido. Outras funções tentarão identificar a relação; contudo, para obter os melhores resultados, você sempre deve criar uma relação quando possível. Os modelos de dados tabulares dão suporte a várias relações entre tabelas. Para evitar confusão ou resultados incorretos, apenas uma relação por vez é designada como a ativa, mas você pode alterar a relação ativa, conforme necessário, para atravessar conexões diferentes nos dados em cálculos. A função [USERELATIONSHIP](#) pode ser usada para especificar uma ou mais relações a serem usadas em um cálculo específico.

É importante observar essas regras de design de fórmula ao usar relações:

- Quando tabelas forem conectadas por uma relação, você deve verificar se as duas colunas usadas como chaves têm valores correspondentes. A integridade referencial não é imposta, portanto, é possível ter valores não correspondentes em uma coluna de chave e ainda criar uma relação. Se isso acontecer, você deverá estar ciente de que valores em branco ou não correspondentes podem afetar os resultados de fórmulas.
- Quando você vincula tabelas em seu modelo usando relações, você aumenta o escopo ou o *contexto*, no qual as fórmulas são avaliadas. Alterações no contexto resultantes da adição de novas tabelas, novas relações ou de alterações na relação ativa podem fazer seus resultados serem alterados de maneiras que você não pode prever. Para obter informações, confira [Contexto](#) neste artigo.

Processar e atualizar

Processo e *recálculo* são duas operações separadas, mas relacionadas. Você deve entender minuciosamente esses conceitos ao criar um modelo que contém fórmulas complexas, grandes quantidades de dados ou dados obtidos de fontes de dados externas.

O *processo (atualização)* está atualizando os dados em um modelo com novos dados de uma fonte de dados externa.

Recálculo é o processo de atualizar os resultados de fórmulas para refletir alterações nas próprias fórmulas e nos dados subjacentes. O recálculo pode afetar o desempenho das seguintes maneiras:

- Os valores em uma coluna calculada são calculados e armazenados no modelo. Para atualizar os valores na coluna calculada, você deve processar o modelo usando um dos três comandos de processamento – Processar Completo, Processar Dados ou Processar Recálculo. O resultado da fórmula deverá ser

recalculado para toda a coluna sempre que você alterar a fórmula.

- Os valores calculados por medidas são avaliados dinamicamente sempre que um usuário adiciona a medida a uma tabela dinâmica ou abre um relatório; conforme o usuário modifica o contexto, os valores retornados pela medida são alterados. Os resultados da medida sempre refletem o mais recente no cache na memória.

O processamento e o recálculo não têm efeito sobre as fórmulas de filtro de linha, a menos que o resultado de um recálculo retorne um valor diferente, tornando, portanto, a linha consultável ou não pelos membros de função.

Solução de problemas

Se você receber um erro ao definir uma fórmula, ela poderá conter um *erro sintático*, *erro semântico* ou *erro de cálculo*.

Erros sintáticos são os mais fáceis de serem resolvidos. Normalmente, eles envolvem um parêntese ou uma vírgula ausente. .

O outro tipo de erro ocorre quando a sintaxe está correta, mas o valor ou uma coluna referenciado não faz sentido no contexto da fórmula. Esses erros semânticos e de cálculo podem ser causados por qualquer um dos seguintes problemas:

- A fórmula refere-se a uma coluna, tabela ou função não existente.
- A fórmula parece estar correta, mas quando o mecanismo de dados os busca, ele encontra tipos incompatíveis e gera um erro.
- A fórmula passa um número ou um tipo de argumento incorreto para uma função.
- A fórmula refere-se a uma coluna diferente que tem um erro e, portanto, seus valores são inválidos.
- A fórmula refere-se a uma coluna que não foi processada, o que significa que ela tem metadados, mas nenhum dado real a ser usado para cálculos.

Nos primeiros quatro casos, o DAX sinaliza a coluna inteira que contém a fórmula inválida. No último caso, o DAX deixa a coluna esmaecida para indicar que a coluna está em um estado não processado.

Aplicativos e ferramentas

Power BI Desktop



O [Power BI Desktop](#) é um aplicativo gratuito de modelagem de dados e de relatório. O designer do modelo inclui um editor DAX para criar fórmulas de cálculo DAX.

Power Pivot no Excel



O designer de modelos do [Power Pivot no Excel](#) inclui um editor DAX para criar fórmulas de cálculo DAX.

Visual Studio



O [SSDT](#) (SQL Server Data Tools) é uma ferramenta essencial para criar e implantar modelos de dados do Analysis

Services. O designer do modelo inclui um editor DAX para criar fórmulas de cálculo DAX.

A extensão [Projetos do Analysis Services](#) (VSIX) inclui a mesma funcionalidade no SSDT para criar projetos de modelagem do Analysis Services. Não instale pacotes se o SSDT já estiver instalado.

SQL Server Management Studio



O [SSMS](#) (SQL Server Management Studio) é uma ferramenta essencial para trabalhar com o Analysis Services. O SSMS inclui um editor de consultas DAX para consultar modelos tabulares e multidimensionais.

DAX Studio



O [DAX Studio](#) é uma ferramenta de cliente de software livre para criar e executar consultas DAX em modelos do Analysis Services, do Power BI Desktop e do Power Pivot no Excel.

Recursos de aprendizado

Ao conhecer o DAX, é melhor usar o aplicativo que você usará para criar os modelos de dados. O Analysis Services, o Power BI Desktop e o Power Pivot no Excel têm artigos e tutoriais que incluem lições sobre como criar medidas, colunas calculadas e filtros de linha usando o DAX. Veja alguns recursos adicionais:

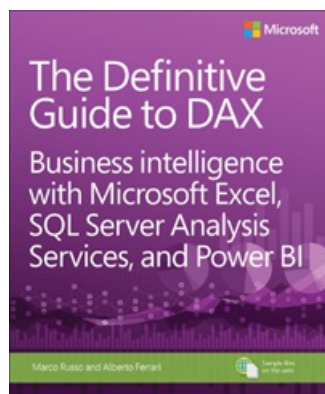
Vídeos

[Noções básicas do DAX no Power BI Desktop](#)

[Início Rápido: Conheça as noções básicas do DAX em 30 minutos \(Power Pivot no Excel\)](#)

[Wiki da Central de Recursos do DAX](#)

The [Definitive Guide to DAX](#) (O Guia Definitivo para o DAX) de Alberto Ferrari e Marco Russo (Microsoft Press). Este guia extensivo fornece noções básicas para técnicas inovadoras de alto desempenho para modeladores de dados e profissionais de BI iniciantes.



Vídeos

21/05/2020 • 2 minutes to read

Se você está usando o Power BI Desktop, o Power Pivot no Excel ou o Analysis Services, aprender a usar o DAX (Data Analysis Expressions) é essencial para a criação de modelos de dados efetivos. Confira abaixo alguns vídeos para ajudar você a começar a usar essa linguagem de expressão avançada.

Conceitos básicos do DAX

Neste vídeo intitulado Conceitos básicos do DAX, o Microsoft Partner Alberto Ferrari apresenta os conceitos essenciais do DAX. Com exemplos práticos e claros, você aprenderá mais sobre medidas, colunas calculadas e expressões de modelagem de dados básicas com o DAX.

DAX avançado

Neste vídeo intitulado DAX avançado, o Microsoft Partner Alberto Ferrari descreve a teoria do DAX, o contexto de filtro e linha e outros conceitos essenciais do DAX.

Referência de funções DAX

26/05/2020 • 5 minutes to read

Essa referência de funções fornece informações detalhadas, incluindo sintaxe, parâmetros, valores retornados e exemplos para cada uma das mais de 200 funções usadas em fórmulas DAX (Data Analysis Expression).

IMPORTANT

Nem todas as funções DAX têm suporte ou estão incluídas em versões anteriores do Power BI Desktop, do Analysis Services e do Power Pivot no Excel.

Exemplos

A maioria dos artigos de referência contém exemplos que mostram fórmulas e resultados criados em uma pasta de trabalho do Excel com um Power Pivot no modelo de dados do Excel. O modelo de dados contém os dados importados do banco de dados de exemplo AdventureWorksDW. A pasta de trabalho de exemplo não está mais disponível.

Esta referência não serve como um tutorial nem fornece diretrizes detalhadas sobre como criar fórmulas para modelos de dados que contêm dados de uma fonte de dados específica.

Nesta seção

Novas funções DAX – essas são funções novas ou existentes que foram atualizadas de maneira significativa.

Funções de data e hora (DAX) – essas funções do DAX são semelhantes às funções de data e hora do Microsoft Excel. No entanto, as funções DAX são baseadas nos tipos de dados datetime usados pelo Microsoft SQL Server.

Funções de inteligência de dados temporais (DAX) – essas funções ajudam você a criar cálculos que usam o conhecimento interno sobre calendários e datas. Usando intervalos de data e hora em combinação com agregações ou cálculos, você pode criar comparações significativas em períodos de tempo comparáveis para vendas, estoque etc.

Funções de filtro (DAX) – essas funções ajudam você a retornar tipos de dados específicos, pesquisar valores em tabelas relacionadas e filtrar o conteúdo por valores relacionados. As funções de pesquisa funcionam usando tabelas e relações entre elas. As funções de filtragem permitem que você manipule o contexto de dados para criar cálculos dinâmicos.

Funções de informações (DAX) – essas funções examinam uma tabela ou uma coluna fornecida como um argumento para outra função e informa se o valor corresponde ao tipo esperado. Por exemplo, a função ISERROR retornará TRUE se o valor referenciado contiver um erro.

Funções lógicas (DAX) – essas funções retornam informações sobre os valores em uma expressão. Por exemplo, a função TRUE permite que você saiba se uma expressão que está sendo avaliada retorna um valor TRUE.

Funções matemáticas e trigonométricas (DAX) – as funções matemáticas do DAX são semelhantes às funções matemáticas e trigonométricas do Excel. No entanto, há algumas diferenças nos tipos de dados numéricos usados pelas funções DAX.

Outras funções (DAX) – essas funções executam ações exclusivas que não podem ser definidas por nenhuma das categorias às quais a maioria das outras funções pertencem.

[Funções pai e filho \(DAX\)](#) – essas funções DAX (Data Analysis Expressions) ajudam os usuários a gerenciar os dados que são apresentados como uma hierarquia pai/filho nos modelos de dados.

[Funções estatísticas \(DAX\)](#) – essas funções executam agregações. Além de criar somas e médias ou encontrar os valores mínimo e máximo, no DAX, você também pode filtrar uma coluna antes de agregá-la ou criar agregações baseadas em tabelas relacionadas.

[Funções de texto \(DAX\)](#) – com essas funções, você pode retornar parte de uma cadeia de caracteres, pesquisar um texto em uma cadeia de caracteres ou concatenar valores de cadeia de caracteres. Funções adicionais destinam-se a controlar os formatos de datas, horas e números.

Consulte também

[Referência de Sintaxe do DAX](#)

[Referência do operador DAX](#)

[Convenções de nomenclatura de parâmetro DAX](#)

Novas funções do DAX

08/05/2020 • 2 minutes to read

O DAX está sendo continuamente aprimorado com novas funções e funcionalidades para dar suporte a novos recursos. Novas funções e atualizações são incluídas nas atualizações de serviço, aplicativo e ferramenta que, na maioria dos casos, são mensais. O SQL Server Analysis Services e o Excel são atualizados na próxima atualização cumulativa. Verifique se você está usando a versão mais recente se você tem preferência em usá-la.

Embora as funções e a funcionalidade estejam sendo atualizadas o tempo todo, somente as atualizações que têm uma alteração visível e funcional exposta aos usuários são descritas na documentação. Novas funções e atualizações de funções existentes no último ano são mostradas aqui.

IMPORTANT

Nem todas as funções são compatíveis com versões anteriores do Power BI Desktop, do Analysis Services e do Power Pivot no Excel ou estão incluídas nessas versões. Funções novas e atualizadas geralmente são introduzidas primeiro no Power BI Desktop.

Novas funções

FUNÇÃO	MÊS
COALESCE	Março de 2020
FIRSTNONBLANKVALUE	Fev. 2020
LASTNONBLANKVALUE	Fev. 2020
CONVERT	Set 2019
REMOVEFILTERS	Set 2019
SELECTEDMEASUREFORMATSTRING	Maio de 2019
ALLCROSSFILTERED	Abril de 2019

Funções atualizadas

FUNÇÃO	MÊS
--------	-----

Funções de data e hora

08/05/2020 • 3 minutes to read

Essas funções ajudam você a criar cálculos baseados em datas e horas. Muitas das funções do DAX são semelhantes às funções de data e hora do Excel. No entanto, as funções DAX usam um tipo de dados **datetime** e podem usar valores de uma coluna como argumento.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
CALENDAR	Retorna uma tabela com uma única coluna chamada "Data" que contém um conjunto contíguo de datas.
CALENDARAUTO	Retorna uma tabela com uma única coluna chamada "Data" que contém um conjunto contíguo de datas.
DATE	Retorna a data especificada no formato datetime.
DATEDIFF	Retorna a contagem de limites de intervalo cruzados entre duas datas.
DATEVALUE	Converte uma data no formato de texto em uma data no formato datetime.
DAY	Retorna o dia do mês, um número de 1 a 31.
EDATE	Retorna a data que corresponde ao número indicado de meses antes ou depois da data de início.
EOMONTH	Retorna a data no formato datetime do último dia do mês antes ou depois de um número especificado de meses.
HOUR	Retorna a hora como um número de 0 (0h) a 23 (23h).
MINUTE	Retorna o minuto como um número de 0 a 59, considerando um valor de data e hora.
MONTH	Retorna o mês como um número entre 1 (janeiro) e 12 (dezembro).
NOW	Retorna a data e a hora atuais no formato datetime .
QUARTER	Retorna o trimestre como um número de 1 a 4.
SECOND	Retorna os segundos de um valor temporal, como um número de 0 a 59.
TIME	Converte horas, minutos e segundos atribuídos como números em uma hora no formato datetime.

FUNÇÃO	DESCRIÇÃO
TIMEVALUE	Converte uma hora no formato de texto em uma hora no formato datetime.
TODAY	Retorna a data atual.
WEEKDAY	Retorna um número de 1 a 7 que identifica o dia da semana de uma data.
WEEKNUM	Retorna o número da semana para a data e o ano especificados de acordo com o valor de return_type.
YEAR	Retorna o ano de uma data como um inteiro de quatro dígitos no intervalo 1900-9999.
YEARFRAC	Calcula a fração do ano representada pelo número de dias inteiros existentes entre duas datas.

CALENDAR

08/05/2020 • 2 minutes to read

Retorna uma tabela com uma única coluna chamada "Data" que contém um conjunto contíguo de datas. O intervalo de datas é da data de início especificada até a data de término especificada, inclusive essas duas datas.

Sintaxe

```
CALENDAR(<start_date>, <end_date>)
```

Parâmetros

TERMO	DEFINIÇÃO
start_date	Qualquer expressão DAX que retorna um valor datetime.
end_date	Qualquer expressão DAX que retorna um valor datetime.

Valor retornado

Retorna uma tabela com uma única coluna chamada "Data" contendo um conjunto contíguo de datas. O intervalo de datas é da data de início especificada até a data de término especificada, inclusive essas duas datas.

Comentários

Um erro será retornado se start_date for maior que end_date.

Exemplos

A fórmula a seguir retorna uma tabela com datas entre 1º de janeiro de 2005 e 31 de dezembro de 2015.

```
=CALENDAR (DATE (2005, 1, 1), DATE (2015, 12, 31))
```

Para um modelo de dados que inclui os dados de vendas reais e as previsões de vendas futuras. A expressão a seguir retorna a tabela de datas que abrange o intervalo de datas nessas duas tabelas.

```
=CALENDAR (MINX (Sales, [Date]), MAXX (Forecast, [Date]))
```


CALENDARAUTO

08/05/2020 • 2 minutes to read

Retorna uma tabela com uma única coluna chamada "Data" que contém um conjunto contíguo de datas. O intervalo de datas é calculado automaticamente com base nos dados no modelo.

Sintaxe

```
CALENDARAUTO([fiscal_year_end_month])
```

Parâmetros

TERMO	DEFINIÇÃO
fiscal_year_end_month	Qualquer expressão DAX que retorna um inteiro de 1 a 12. Se omitido, o padrão será o valor especificado no modelo de tabela de calendário para o usuário atual, se houver; caso contrário, o padrão será 12.

Valor retornado

Retorna uma tabela com uma única coluna chamada "Data" que contém um conjunto contíguo de datas. O intervalo de datas é calculado automaticamente com base nos dados no modelo.

Comentários

O intervalo de datas é calculado da seguinte maneira:

- A primeira data no modelo que não está em uma coluna calculada ou tabela calculada é considerada como MinDate.
- A última data no modelo que não está em uma coluna calculada ou tabela calculada é considerada como MaxDate.
- O intervalo de datas retornado corresponde às datas entre o início do ano fiscal associado a MinDate e o fim do ano fiscal associado a MaxDate.

Um erro será retornado se o modelo não contiver nenhum valor de data e hora que não esteja em colunas calculadas ou tabelas calculadas.

Exemplo

Neste exemplo, MinDate e MaxDate no modelo de dados são 1º de julho de 2010 e 30 de junho de 2011.

`CALENDARAUTO()` retornará todas as datas entre 1º de janeiro de 2010 e 31 de dezembro de 2011.

`CALENDARAUTO(3)` retornará todas as datas entre 1º de março de 2010 e 28 de fevereiro de 2012.

DATE

08/05/2020 • 9 minutes to read

Retorna a data especificada no formato **datetime**.

Sintaxe

```
DATE(<year>, <month>, <day>)
```

Parâmetros

TERMO	DEFINIÇÃO
ano	<p>Um número que representa o ano.</p> <p>O valor do argumento year pode incluir de um a quatro dígitos. O argumento year é interpretado de acordo com o sistema de data usado pelo computador.</p> <p>Há suporte para datas que começam em 1º de março de 1900.</p> <p>Se você inserir um número que tenha casas decimais, o número será arredondado.</p> <p>Para valores maiores que 9999 ou menores que zero (valores negativos), a função retorna um erro #VALUE! .</p> <p>Se o valor de year estiver entre 0 e 1899, o valor será adicionado a 1900 para produzir o valor final. Confira os exemplos abaixo. Observação: Você deverá usar quatro dígitos para o argumento year sempre que possível para evitar resultados indesejados. Por exemplo, o uso de 07 retorna 1907 como o valor de ano.</p>

TERMO	DEFINIÇÃO
mês	<p>Um número que representa o mês ou um cálculo de acordo com as seguintes regras:</p> <p>Se month for um número de 1 a 12, ele representará um mês do ano. 1 representa janeiro, 2 representa fevereiro e assim por diante até 12, que representa dezembro.</p> <p>Se você inserir um inteiro maior que 12, ocorrerá o seguinte cálculo: a data será calculada adicionando o valor de month ao year. Por exemplo, se você tiver DATE(2008, 18, 1), a função retornará um valor de datetime equivalente a 1º de junho de 2009, porque 18 meses são adicionados ao início de 2008, resultando em um valor igual a junho de 2009. Veja os exemplos abaixo.</p> <p>Se você inserir um inteiro negativo, ocorrerá o seguinte cálculo: a data será calculada subtraindo o valor de month de year. Por exemplo, se você tiver DATE(2008, -6, 15), a função retornará um valor de datetime equivalente a 15 de junho de 2007, porque quando 6 meses são subtraídos do início de 2008, ele resulta em um valor igual a junho de 2007. Veja os exemplos abaixo.</p>
day	<p>Um número que representa o dia ou um cálculo de acordo com as seguintes regras:</p> <p>Se day for um número de 1 até o último dia do mês especificado, ele representará um dia do mês.</p> <p>Se você inserir um inteiro maior que o último dia do mês especificado, ocorrerá o seguinte cálculo: a data será calculada adicionando o valor de day ao month. Por exemplo, na fórmula DATE(2008, 3, 32), a função DATE retorna um valor de datetime equivalente a 1º de abril de 2008, pois 32 dias são adicionados ao início de março, resultando em um valor igual a 1º de abril.</p> <p>Se você inserir um inteiro negativo, ocorrerá o seguinte cálculo: a data será calculada subtraindo o valor de day de month. Por exemplo, na fórmula DATE(2008, 5, -15), a função DATE retorna um valor de datetime equivalente a 15 de abril de 2008, pois 15 dias são subtraídos do início de maio de 2008, resultando em um valor igual a abril de 2008.</p> <p>Se day contiver uma parte decimal, ele será arredondado para o valor inteiro mais próximo.</p>

Valor retornado

Retorna a data especificada (**datetime**) .

Comentários

A função DATE usa os inteiros que são inseridos como argumentos e gera a data correspondente. A função DATE é mais útil em situações em que o ano, o mês e o dia são fornecidos por fórmulas. Por exemplo, os dados subjacentes podem conter datas em um formato que não é reconhecido como uma data, como YYYYMMDD. Você pode usar a função DATE em conjunto com outras funções para converter as datas em um número que pode ser reconhecido como uma data.

Ao contrário do Microsoft Excel, que armazena datas como um número de série, as funções de data DAX sempre retornam um tipo de dados **datetime**. No entanto, você poderá usar a formatação para exibir datas como números de série, se desejar.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

Exemplo: Como retorno de uma data simples

Descrição

A seguinte fórmula retorna a data 8 de julho de 2009:

Código

```
=DATE(2009,7,8)
```

Exemplo: Anos anteriores a 1899

Descrição

Se o valor inserido para o argumento **year** estiver entre 0 (zero) e 1899 (inclusive), esse valor será adicionado a 1900 para o cálculo do ano. A seguinte fórmula retorna 2 de janeiro de 1908: (1900+08).

Código

```
=DATE(08,1,2)
```

Exemplo: Anos anteriores a 1899

Descrição

Se o valor inserido para o argumento **year** estiver entre 0 (zero) e 1899 (inclusive), esse valor será adicionado a 1900 para o cálculo do ano. A seguinte fórmula retorna 2 de janeiro de 3700: (1900+1800).

Código

```
=DATE(1800,1,2)
```

Exemplo: Anos posteriores a 1899

Descrição

Se **year** estiver entre 1900 e 9999 (inclusive), esse valor será usado como o ano. A seguinte fórmula retorna 2 de janeiro de 2008:

Código

```
=DATE(2008,1,2)
```

Exemplo: Como trabalhar com meses

Descrição

Se **month** for maior que 12, **month** adicionará esse número de meses ao primeiro mês do ano especificado. A

seguinte fórmula retorna a data 2 de fevereiro de 2009:

Código

```
=DATE(2008,14,2)
```

Comentário

Se o valor de **month** for menor que 1, a função DATE subtrairá a magnitude desse número de meses, mais 1, do primeiro mês do ano especificado. A seguinte fórmula retorna 2 de setembro de 2007:

```
=DATE(2008,-3,2)
```

Exemplo: Como trabalhar com dias

Descrição

Se **day** for maior que o número de dias do mês especificado, **day** adicionará esse número de dias ao primeiro dia do mês. A seguinte fórmula retorna a data 4 de fevereiro de 2008:

Código

```
=DATE(2008,1,35)
```

Comentário

Se **day** for menor que 1, **day** subtrairá a magnitude do número de dias, mais um, do primeiro dia do mês especificado. A seguinte fórmula retorna 16 de dezembro de 2007:

```
=DATE(2008,1,-15)
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função DAY \(DAX\)](#)

[Função TODAY \(DAX\)](#)

DATEDIFF

09/05/2020 • 2 minutes to read

Retorna a contagem de limites de intervalo cruzados entre duas datas.

Sintaxe

```
DATEDIFF(<start_date>, <end_date>, <interval>)
```

Parâmetros

TERMO	DEFINIÇÃO
start_date	Um valor de datetime escalar.
end_date	Um Valor retornado do valor datetime escalar.
interval	<p>O intervalo a ser usado ao comparar datas. O valor pode ser um dos seguintes:</p> <ul style="list-style-type: none">- SECOND- MINUTE- HOUR- DAY- WEEK- MONTH- QUARTER- YEAR

Valor retornado

A contagem de limites de intervalo cruzados entre duas datas.

Comentários

Um erro será retornado se start_date for maior que end_date.

Exemplo

DATA
2012-12-31 23:59:59
2013-01-01 00:00:00

Todos os seguintes retornam 1:

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), SECOND )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MINUTE )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), HOUR )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), DAY )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), WEEK )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), MONTH )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), QUARTER )
```

```
DATEDIFF(MIN( Calendar[Date] ), MAX( Calendar[Date]), YEAR )
```

DATEVALUE

08/05/2020 • 3 minutes to read

Converte uma data no formato de texto em uma data no formato **datetime**.

Sintaxe

```
DATEVALUE(date_text)
```

Parâmetros

TERMO	DEFINIÇÃO
date_text	Texto que representa uma data.

Valor da propriedade/valor retornado

Uma data no formato **datetime**.

Comentários

Ao converter, o DATEVALUE usa as configurações da localidade e de data/hora do modelo para determinar um valor de data. Se o modelo atual de data/hora representar datas no formato de Mês/Dia/Ano, a cadeia de caracteres "1/8/2009" é convertida em um valor de **datetime** equivalente a 8 de janeiro de 2009. No entanto, se o modelo de data/hora representar datas no formato de Dia/Mês/Ano, a mesma cadeia de caracteres é convertida como um valor de **datetime** equivalente a 1º de agosto de 2009.

Se a conversão usando as configurações da localidade e de data/hora do modelo falhar, o DATEVALUE tentará usar outros formatos de data. Nesse caso, algumas linhas podem ser convertidas usando um formato e outras linhas são convertidas usando um formato diferente. Por exemplo, "5/4/2018" pode ser convertido em 4 de maio de 2018 e "20/4/2018" pode ser convertido em 20 de abril de 2018.

Se a parte de ano do argumento **date_text** for omitida, a função DATEVALUE usará o ano atual do relógio interno do computador. As informações de hora no argumento **date_text** são ignoradas.

As configurações de localidade e de data/hora do modelo são inicialmente determinadas pelo aplicativo e computador quando o modelo é criado.

Exemplo

O exemplo a seguir retorna outro valor de **datetime**, dependendo da localidade do modelo e das configurações de como as datas e as horas são apresentadas.

- Nas configurações de data/hora em que o dia precede o mês, o exemplo retorna um valor de **datetime** correspondente a 8 de janeiro de 2009.
- Nas configurações de data/hora em que o mês precede o dia, o exemplo retorna um valor de **datetime** correspondente a 1º de agosto de 2009.

=DATEVALUE("8/1/2009")

Consulte também

[Funções de data e hora \(DAX\)](#)

Retorna o dia do mês, um número de 1 a 31.

Sintaxe

```
DAY(<date>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	Uma data no formato datetime ou uma representação de texto de uma data.

Valor retornado

Um número inteiro que indica o dia do mês.

Comentários

A função DAY usa como um argumento a data do dia que você está tentando localizar. As datas podem ser fornecidas para a função usando outra função de data, usando uma expressão que retorna uma data ou digitando uma data em um formato **datetime**. Você também pode digitar uma data em um dos formatos de cadeia de caracteres aceitos para datas.

Os valores retornados pelas funções YEAR, MONTH e DAY serão valores gregorianos, independentemente do formato de exibição do valor de data fornecido. Por exemplo, se o formato de exibição da data fornecida for Hijri, os valores retornados para as funções YEAR, MONTH e DAY serão valores associados à data gregoriana equivalente.

Quando o argumento de data é uma representação de texto da data, a função day usa as configurações de localidade e de data/hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. Se as configurações atuais de data/hora representarem datas no formato de Mês/Dia/Ano, a cadeia de caracteres "1/8/2009" será interpretada como valor de **datetime** equivalente a 8 de janeiro de 2009 e a função retorna 8. No entanto, se as configurações atuais de data/hora representarem datas no formato de Dia/Mês/Ano, a mesma cadeia de caracteres será interpretada como um valor de **datetime** equivalente a 1º de agosto de 2009 e a função retornará 1.

Exemplo: Como obter o dia de uma coluna de data

A fórmula a seguir retorna o dia da data na coluna [DataDeNascimento].

```
=DAY([Birthdate])
```

Exemplo – como obter o dia de uma data de cadeia de caracteres

As fórmulas a seguir retornam o dia 4 usando datas que foram fornecidas como cadeias de caracteres em um

formato de texto aceito.

```
=DAY("3-4-1007")  
=DAY("March 4 2007")
```

Exemplo – como usar um valor de dia como uma condição

A expressão a seguir retornará o dia em que cada pedido de venda foi feito e sinalizará a linha como um item de liquidação promocional se o pedido tiver sido feito no dia 10 do mês.

```
=IF( DAY([SalesDate])=10,"promotion","")
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função TODAY \(DAX\)](#)

[Função DATE \(DAX\)](#)

EDATE

09/05/2020 • 3 minutes to read

Retorna a data que corresponde ao número indicado de meses antes ou depois da data de início. Use EDATE para calcular datas de vencimento ou datas de conclusão que caem no mesmo dia do mês que a data de emissão.

Sintaxe

```
EDATE(<start_date>, <months>)
```

Parâmetros

TERMO	DEFINIÇÃO
start_date	Uma data no formato datetime ou texto que representa a data de início.
meses	Um número inteiro que representa o número de meses antes ou depois de start_date .

Valor retornado

Uma data (**datetime**).

Comentários

Ao contrário do Microsoft Excel, que armazena datas como números de série sequenciais, o DAX trabalha com datas no formato **datetime**. As datas armazenadas em outros formatos são convertidas de forma implícita.

Se **start_date** não for uma data válida, a função EDATE retornará um erro. Verifique se a referência de coluna ou a data que você fornece como primeiro argumento é uma data.

Se **meses** não for um número inteiro, ele estará truncado.

Quando o argumento de data é uma representação de texto da data, a função EDATE usa as configurações de localidade e de data/hora do computador cliente para interpretar o valor de texto a fim de realizar a conversão. Se as configurações atuais de data/hora representarem uma data no formato de Mês/Dia/Ano, a cadeia de caracteres a seguir "1/8/2009" será interpretada como valor de datetime equivalente a 8 de janeiro de 2009. No entanto, se as configurações atuais de data/hora representarem uma data no formato de Dia/Mês/Ano, a mesma cadeia de caracteres será interpretada como um valor de datetime equivalente a 1º de agosto de 2009.

Se a data solicitada for posterior ao último dia do mês correspondente, então será retornado o último dia do mês. Por exemplo, as seguintes funções: EDATE ("2009-01-29", 1), EDATE ("2009-01-30", 1), EDATE ("2009-01-31", 1) retornam 28 de fevereiro de 2009. Isso corresponde a um mês após a data de início.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

Exemplo

O exemplo a seguir retorna a data três meses após a data do pedido, que é armazenada na coluna [TransactionDate].

```
=EDATE([TransactionDate],3)
```

Consulte também

[Função EOMONTH \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

EOMONTH

09/05/2020 • 3 minutes to read

Retorna a data no formato **datetime** do último dia do mês antes ou depois de um número especificado de meses. Use EOMONTH para calcular datas de vencimento ou datas de conclusão que se enquadram no último dia do mês.

Sintaxe

```
EOMONTH(<start_date>, <months>)
```

Parâmetros

TERMO	DEFINIÇÃO
start_date	A data de início no formato datetime ou em uma representação de texto aceita de uma data.
meses	Um número que representa o número de meses antes ou depois de start_date . Observação: Se você inserir um número que não seja um inteiro, o número será arredondado para cima ou para baixo até o número inteiro mais próximo.

Valor retornado

Uma data (**datetime**).

Comentários

Ao contrário do Microsoft Excel, que armazena datas como números de série sequenciais, o DAX trabalha com datas no formato **datetime**. A função EOMONTH pode aceitar datas em outros formatos, com as seguintes restrições:

Se **start_date** não for uma data válida, EOMONTH retornará um erro.

Se **start_date** for um valor numérico que não esteja em um formato **datetime**, EOMONTH converterá o número em uma data. Para evitar resultados inesperados, converta o número em um formato **datetime** antes de usar a função EOMONTH.

Se **start_date** mais meses gerar uma data inválida, EOMONTH retornará um erro. As datas anteriores a 1º de março de 1900 e depois de 31 de dezembro de 9999 são inválidas.

Quando o argumento de data é uma representação de texto da data, a função EDATE usa as configurações de localidade e de data/hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. Se as configurações atuais de data/hora representarem uma data no formato de Mês/Dia/Ano, a cadeia de caracteres a seguir "1/8/2009" será interpretada como valor de datetime equivalente a 8 de janeiro de 2009. No entanto, se as configurações atuais de data/hora representarem uma data no formato de Dia/Mês/Ano, a mesma cadeia de caracteres será interpretada como um valor de datetime equivalente a 1º de agosto de 2009.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

Exemplo

A expressão a seguir retorna 31 de maio de 2008, porque o argumento **meses** é arredondado para 2.

```
=EOMONTH("March 3, 2008",1.5)
```

Consulte também

[Função EDATE \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

HOUR

09/05/2020 • 2 minutes to read

Retorna a hora como um número de 0 (0h) a 23 (23h).

Sintaxe

```
HOUR(<datetime>)
```

Parâmetros

TERMO	DEFINIÇÃO
datetime	Um valor datetime , como 16:48:00 ou 16h48.

Valor retornado

Um número inteiro de 0 a 23.

Comentários

A função HOUR usa como argumento o horário que contém a hora que você deseja localizar. Você pode fornecer a hora usando uma função de data/hora, uma expressão que retorna um **datetime** ou digitando o valor diretamente em um dos formatos de hora aceitos. Os horários também podem ser inseridos como qualquer representação de texto aceita de uma hora.

Quando o argumento **datetime** é uma representação de texto da data e da hora, a função usa as configurações de localidade e de data/hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. A maioria das localidades usa os dois-pontos (:) como separador de hora, de modo que todo texto de entrada que usar os dois-pontos como separador de hora será analisado corretamente. Examine as configurações de localidade para entender os resultados.

Exemplo

O exemplo a seguir retorna a hora da coluna **TransactionTime** de uma tabela chamada **Orders**.

```
=HOUR('Orders'[TransactionTime])
```

Exemplo

O exemplo a seguir retorna 15, o que significa que a hora corresponde a 15h em um relógio de 24 horas. O valor de texto é analisado automaticamente e convertido em um valor de data/hora.

```
=HOUR("March 3, 2008 3:00 PM")
```

Consulte também

Funções de data e hora (DAX)

Função MINUTE (DAX)

Função YEAR (DAX)

Função SECOND (DAX)

MINUTE

08/05/2020 • 2 minutes to read

Retorna o minuto como um número de 0 a 59, considerando um valor de data e hora.

Sintaxe

```
MINUTE(<datetime>)
```

Parâmetros

TERMO	DEFINIÇÃO
datetime	Um valor ou texto de datetime em um formato de hora aceito, como 16:48:00 ou 4:48 PM.

Valor retornado

Um número inteiro de 0 a 59.

Comentários

Diferente do Microsoft Excel, que armazena as datas e horas em um formato numérico serial, o DAX usa o tipo de dados **datetime** para trabalhar com datas e horas. Você pode fornecer o valor de **datetime** para a função MINUTE referenciando uma coluna que armazena datas e horas, usando uma função de data/hora ou usando uma expressão que retorna uma data e hora.

Quando o argumento **datetime** é uma representação de texto da data e da hora, a função usa as configurações de localidade e de data/hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. A maioria das localidades usa os dois-pontos (:) como separador de hora, de modo que todo texto de entrada que usar os dois-pontos como separador de hora será analisado corretamente. Verifique as configurações de localidade para entender os resultados.

Exemplo

O exemplo a seguir retorna o minuto do valor armazenado na coluna **TransactionTime** da tabela **Orders**.

```
=MINUTE(Orders[TransactionTime])
```

Exemplo

O exemplo a seguir retorna 45, que é o número de minutos na hora 1:45 PM.

```
=MINUTE("March 23, 2008 1:45 PM")
```

Consulte também

Funções de data e hora (DAX)

Função HOUR (DAX)

Função YEAR (DAX)

Função SECOND (DAX)

MÊS

09/05/2020 • 3 minutes to read

Retorna o mês como um número entre 1 (janeiro) e 12 (dezembro).

Sintaxe

```
MONTH(<datetime>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	Uma data em datetime ou em formato de texto.

Valor retornado

Um número inteiro de 1 a 12.

Comentários

Ao contrário do Microsoft Excel, que armazena datas como números de série, o DAX usa o formato **datetime** ao trabalhar com datas. Você pode inserir a data usada como argumento para a função MONTH digitando um formato **datetime** aceito, fornecendo uma referência a uma coluna que contenha datas ou usando uma expressão que retorna uma data.

Os valores retornados pelas funções YEAR, MONTH e DAY serão valores gregorianos, independentemente do formato de exibição do valor de data fornecido. Por exemplo, se o formato de exibição da data fornecida for Hijri, os valores retornados para as funções YEAR, MONTH e DAY serão valores associados à data gregoriana equivalente.

Quando o argumento de data é uma representação de texto da data, a função usa as configurações de localidade e de data e hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. Se as configurações atuais de data/hora representarem uma data no formato de Mês/Dia/Ano, a cadeia de caracteres a seguir "1/8/2009" será interpretada como valor de datetime equivalente a 8 de janeiro de 2009 e a função retornará 1. No entanto, se as configurações atuais de data/hora representarem uma data no formato de Dia/Mês/Ano, a mesma cadeia de caracteres será interpretada como um valor datetime equivalente a 1º de agosto de 2009 e a função retornará 8.

Se a representação de texto da data não puder ser convertida corretamente em um valor datetime, a função retornará um erro.

Exemplo

A expressão a seguir retorna 3, que é o inteiro correspondente a março, o mês no argumento **data**.

```
=MONTH("March 3, 2008 3:45 PM")
```

Exemplo

A expressão a seguir retorna o mês da data na coluna **TransactionDate** da tabela **Orders**.

```
=MONTH(Orders[TransactionDate])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função HOUR \(DAX\)](#)

[Função MINUTE \(DAX\)](#)

[Função YEAR \(DAX\)](#)

[Função SECOND \(DAX\)](#)

NOW

09/05/2020 • 2 minutes to read

Retorna a data e a hora atuais no formato **datetime**.

A função NOW é útil quando você precisa exibir a data e a hora atuais em uma planilha ou calcular um valor com base na data e hora atuais e ter esse valor atualizado sempre que abrir a planilha.

Sintaxe

```
NOW()
```

Valor retornado

Uma data (**datetime**).

Comentários

O resultado da função NOW só é alterado quando a coluna que contém a fórmula é atualizada. Ele não é atualizado continuamente.

A função TODAY retorna a mesma data, mas não é precisa em relação à hora; a hora retornada é sempre 12h00min00s e somente a data é atualizada.

Exemplo

O exemplo a seguir retorna a data e a hora atuais mais 3,5 dias:

```
=NOW()+3.5
```

Consulte também

[Função UTCNOW](#)

[Função TODAY \(DAX\)](#)

QUARTER

08/05/2020 • 2 minutes to read

Retorna o trimestre como um número de 1 (janeiro – março) a 4 (outubro – dezembro).

Sintaxe

```
QUARTER(<date>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	Uma data.

Valor retornado

Um número inteiro de 1 a 4.

Comentários

Confira funções DAX relacionadas: [YEAR](#), [MONTH](#), [DAY](#).

Se o valor de entrada for BLANK, o valor de saída também será BLANK.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { QUARTER(DATE(2019, 2, 1)), QUARTER(DATE(2018, 12, 31)) }
```

Retorna:

[VALOR]
1
4

Exemplo

A seguinte consulta DAX:

```
EVALUATE
ADDCOLUMNS(
    FILTER(
        VALUES(
            FactInternetSales[OrderDate]),
            [OrderDate] >= DATE(2008, 3, 31) && [OrderDate] <= DATE(2008, 4, 1)
        ),
        "Quarter", QUARTER([OrderDate])
    )
)
```

Retorna:

FACTINTERNETSALES[ORDERDATE]	[QUARTER]
31/03/2008	1
01/04/2008	2

SECOND

09/05/2020 • 3 minutes to read

Retorna os segundos de um valor temporal, como um número de 0 a 59.

Sintaxe

```
SECOND(<time>)
```

Parâmetros

TERMO	DEFINIÇÃO
hora	Um valor de hora no formato datetime , como 16:48:23 ou 4:48:47 da tarde.

Valor retornado

Um número inteiro de 0 a 59.

Comentários

Ao contrário do Microsoft Excel, que armazena datas e horas como números de série, o DAX usa o formato **datetime** ao trabalhar com datas e horas. Se os dados de origem não estiverem nesse formato, o DAX converterá implicitamente os dados. Você poderá usar a formatação para exibir as datas e horas como um número de série se precisar.

O valor de data/hora que você fornecer como argumento para a função SECOND poderá ser inserido como uma cadeia de texto entre aspas (por exemplo, "6:45 da tarde"). Você também poderá fornecer um valor temporal como resultado de outra expressão ou como referência a uma coluna que contenha um valor de hora.

Se você fornecer um valor numérico de outro tipo de dados (como 13,60, por exemplo), o valor será interpretado como um número de série e será representado como um tipo de dados **datetime** antes de extrair o valor de segundos. Para facilitar a compreensão dos resultados, talvez seja melhor representar esses números como datas antes de usá-los na função SECOND. Por exemplo, se você usar a função SECOND com uma coluna que contém um valor numérico – como 25,56 – a fórmula retornará 24. Isso ocorrerá porque, ao ser formatado como data, o valor 25,56 será equivalente a 25 de janeiro de 1900, às 1:26:24 da tarde.

Quando o argumento de **hora** é uma representação de texto de uma data e hora, a função usa as configurações de localidade e de data/hora do computador cliente para interpretar o valor de texto a fim de executar a conversão. A maioria das localidades usa os dois-pontos (:) como separador de hora, de modo que todo texto de entrada que usar os dois-pontos como separador de hora será analisado corretamente. Examine as configurações de localidade para entender os resultados.

Exemplo

A fórmula a seguir retorna o número de segundos na hora contida na coluna **TransactionTime** de uma tabela chamada **Orders**.

```
=SECOND('Orders'[TransactionTime])
```

Exemplo

A fórmula a seguir retorna 3, que é o número de segundos da hora representada pelo valor **3 de março de 2008 às 12:00:03**.

```
=SECOND("March 3, 2008 12:00:03")
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função HOUR \(DAX\)](#)

[Função MINUTE \(DAX\)](#)

[Função YEAR \(DAX\)](#)

TIME

08/05/2020 • 4 minutes to read

Converte horas, minutos e segundos atribuídos como números em uma hora no formato **datetime**.

Sintaxe

```
TIME(hour, minute, second)
```

Parâmetros

TERMO	DEFINIÇÃO
hour	Um número de 0 a 23, que representa a hora. Qualquer valor maior que 23 será dividido por 24 e o resto será tratado como o valor da hora.
minute	Um número de 0 a 59, que representa o minuto. Qualquer valor maior que 59 será convertido em horas e minutos.
second	Um número de 0 a 59, que representa o segundo. Qualquer valor maior que 59 será convertido em horas, minutos e segundos.

Valor retornado

Uma hora (**datetime**).

Comentários

Diferente do Microsoft Excel, que armazena datas e horas como números de série, o DAX trabalha com valores de data e hora em um formato **datetime**. Os números em outros formatos são convertidos implicitamente quando você usa um valor de data/hora em uma função DAX. Se precisar usar números de série, você poderá usar a formatação para alterar a forma como os números são exibidos.

Os valores de hora são uma parte de um valor de data e no sistema de números de série são representados por um número decimal. Portanto, o valor de **datetime** 12:00 é equivalente a 0,5, porque é a metade de um dia.

Você pode fornecer os argumentos para a função TIME como valores digitados diretamente, como o resultado de outra expressão ou por uma referência a uma coluna que contém um valor numérico. As seguintes restrições se aplicam:

- Qualquer valor de **horas** que for maior que 23 será dividido por 24 e o resto será tratado como o valor da hora.
- Qualquer valor de **minutos** que for maior que 59 será convertido em horas e minutos.
- Qualquer valor de **segundos** que for maior que 59 será convertido em horas, minutos e segundos.

- Para minutos ou segundos, qualquer valor maior que 24 horas será dividido por 24 e o resto será tratado como o valor da hora. Um valor que excede 24 horas não altera a parte de data.

Para melhorar a legibilidade dos valores de hora retornados por essa função, recomendamos que você formate a coluna ou a célula da Tabela Dinâmica que contém os resultados da fórmula usando um dos formatos de hora fornecidos pelo Microsoft Excel.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

Exemplo

Os exemplos a seguir retornam a hora, 3:00:

```
=TIME(27,0,0)
=TIME(3,0,0)
```

Exemplo

Os exemplos a seguir retornam a hora, 12:30 PM:

```
=TIME(0,750,0)
=TIME(12,30,0)
```

Exemplo

O exemplo a seguir cria uma hora com base nos valores nas colunas, `intHours`, `intMinutes`, `intSeconds` :

```
=TIME([intHours],[intMinutes],[intSeconds])
```

Consulte também

[Função DATE \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

Função TIMEVALUE

09/05/2020 • 2 minutes to read

Converte uma hora no formato de texto em uma hora no formato datetime.

Sintaxe

```
TIMEVALUE(time_text)
```

Parâmetros

Termo	Definição
time_text	Uma cadeia de caracteres de texto que representa uma determinada hora do dia. Todas as informações de data incluídas no argumento time_text são ignoradas.

Valor retornado

Uma data (**datetime**).

Comentários

Os valores de hora são uma parte de um valor de data e são representados por um número decimal. Por exemplo, 12:00 PM é representado como 0.5 porque é a metade de um dia.

Quando o argumento **time_text** é uma representação de texto da data e da hora, a função usa as configurações de localidade e de data/hora do modelo para reconhecer o valor de texto a fim de fazer a conversão. A maioria das localidades usa os dois-pontos (:) como o separador de hora e qualquer texto de entrada usando dois-pontos como separadores de hora serão analisados corretamente. Examine as configurações de localidade para entender os resultados.

Exemplo

```
=TIMEVALUE("20:45:30")
```

Consulte também

[Funções de data e hora \(DAX\)](#)

TODAY

09/05/2020 • 2 minutes to read

Retorna a data atual.

Sintaxe

```
TODAY()
```

Valor retornado

Uma data (**datetime**).

Comentários

A função TODAY é útil quando você precisa ter a data atual exibida em uma planilha, independentemente de quando abrir a pasta de trabalho. Ela também é útil para calcular intervalos.

NOTE

Se a função TODAY não atualizar a data quando você espera, talvez seja necessário alterar as configurações que controlam quando a coluna ou a pasta de trabalho é atualizada.

A função NOW é semelhante, mas retorna a hora exata, enquanto TODAY retorna o valor temporal 12:00:00 para todas as datas.

Exemplo

Se souber que alguém nasceu em 1963, você poderá usar a fórmula a seguir para descobrir a idade dessa pessoa com base no aniversário deste ano:

```
=YEAR(TODAY())-1963
```

Esta fórmula usa a função TODAY como um argumento para a função YEAR para obter o ano atual e, em seguida, subtrai 1963, retornando a idade da pessoa.

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função NOW \(DAX\)](#)

UTCNOW

09/05/2020 • 2 minutes to read

Retorna a data e a hora UTC atuais

Sintaxe

```
UTCNOW()
```

Valor retornado

Um **(datetime)** .

Comentários

O resultado da função UTCNOW é alterado somente quando a fórmula é atualizada. Ele não é atualizado continuamente.

Exemplo

O seguinte:

```
EVALUATE { FORMAT(UTCNOW(), "General Date") }
```

Retorna:

[VALOR]

2/2/2018 4:48:08 AM

Consulte também

[Função NOW \(DAX\)](#)

[Função UTCTODAY \(DAX\)](#)

UTCTODAY

08/05/2020 • 2 minutes to read

Retorna a data atual no UTC.

Sintaxe

```
UTCTODAY()
```

Valor retornado

Uma data.

Comentários

UTCTODAY retorna o valor temporal 12:00:00 PM para todas as datas.

A função UTCNOW é semelhante, mas retorna a hora e a data exatas.

Exemplo

O seguinte:

```
EVALUATE { FORMAT(UTCTODAY(), "General Date") }
```

Retorna:

[VALOR]

2/2/2018

Consulte também

[Função NOW \(DAX\)](#)

[Função UTCNOW \(DAX\)](#)

WEEKDAY

08/05/2020 • 3 minutes to read

Retorna um número de 1 a 7 que identifica o dia da semana de uma data. Por padrão, o dia varia entre 1 (domingo) e 7 (sábado).

Sintaxe

```
WEEKDAY(<date>, <return_type>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	<p>Uma data no formato datetime.</p> <p>As datas devem ser inseridas com a função DATE, usando expressões que resultam em uma data ou como o resultado de outras fórmulas.</p>
return_type	<p>Um número que determina o valor retornado:</p> <p>Tipo de retorno: 1, a semana começa no domingo (1) e termina no sábado (7). numerado de 1 a 7.</p> <p>Tipo de retorno: 2, a semana começa na segunda-feira (1) e termina no domingo (7).</p> <p>Tipo de retorno: 3, a semana começa na segunda-feira (0) e termina no domingo (6). numerado de 1 a 7.</p>

Valor retornado

Um número inteiro de 1 a 7.

Comentários

Ao contrário do Microsoft Excel, que armazena datas como números de série, o DAX trabalha com datas e horas em um formato **datetime**. Caso precise exibir datas como números de série, use as opções de formatação no Excel.

Além disso, digite datas em uma representação de texto aceita de uma data, mas para evitar resultados inesperados, é melhor converter a data de texto em um formato **datetime** primeiro.

Quando o argumento de data é uma representação de texto da data, a função usa as configurações de localidade e de data/hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. Se as configurações atuais de data/hora representarem datas no formato de Mês/Dia/Ano, a cadeia de caracteres "1/8/2009" será interpretada como valor de **datetime** equivalente a 8 de janeiro de 2009. No entanto, se as configurações atuais de data/hora representarem datas no formato de Dia/Mês/Ano, a mesma cadeia de caracteres será interpretada como um valor de **datetime** equivalente a 1º de agosto de 2009.

Exemplo

O exemplo a seguir obtém a data da coluna [HireDate], adiciona 1 e exibe o dia da semana correspondente a essa data. Como o argumento **return_type** foi omitido, o formato padrão é usado, no qual 1 é domingo e 7 é sábado. Se o resultado for 4, o dia será quarta-feira.

```
=WEEKDAY([HireDate]+1)
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função WEEKNUM \(DAX\)](#)

[Função YEARFRAC \(DAX\)](#)

WEEKNUM

09/05/2020 • 2 minutes to read

Retorna o número da semana para a data e o ano especificados de acordo com o valor de **return_type**. O número da semana indica onde a semana se encaixa numericamente dentro de um ano.

Sintaxe

```
WEEKNUM(<date>, <return_type>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	A data no formato datetime .
return_type	<p>Número que determina o valor retornado: use 1 quando a semana começar no domingo ou 2 quando a semana começar na segunda-feira. O padrão é 1.</p> <p>Tipo de retorno: 1, a semana começa no domingo. Os dias da semana são numerados de 1 a 7.</p> <p>Tipo de retorno: 2, a semana começa na segunda-feira. Os dias da semana são numerados de 1 a 7.</p>

Valor retornado

Um número inteiro.

Comentários

Ao contrário do Microsoft Excel, que armazena as datas como números de série, o DAX usa o tipo de dados **datetime** para trabalhar com datas e horas. Se os dados de origem estiverem em um formato diferente, o DAX converterá implicitamente os dados para o formato **datetime** a fim de executar os cálculos.

Por padrão, a função WEEKNUM usa uma convenção de calendário na qual a semana que contém o dia 1º de Janeiro é considerada a primeira semana do ano. No entanto, o padrão de calendário ISO 8601, amplamente usado na Europa, define a primeira semana como sendo aquela com a maioria dos dias (quatro ou mais) enquadrados no novo ano. Isso significa que nos anos em que há três dias ou menos na primeira semana de Janeiro, a função WEEKNUM retorna números de semana diferentes daqueles retornados segundo a definição da ISO 8601.

Exemplo

O exemplo a seguir retorna o número da semana da data 14 de fevereiro de 2010.

```
=WEEKNUM("Feb 14, 2010", 2)
```

Exemplo

O exemplo a seguir retorna o número da semana da data armazenada na coluna **HireDate** da tabela **Employees**.

```
=WEEKNUM('Employees'[HireDate])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função YEARFRAC \(DAX\)](#)

[Função WEEKDAY \(DAX\)](#)

YEAR

08/05/2020 • 2 minutes to read

Retorna o ano de uma data como um inteiro de quatro dígitos no intervalo 1900-9999.

Sintaxe

```
YEAR(<date>)
```

Parâmetros

TERMO	DEFINIÇÃO
data	Uma data em datetime ou formato de texto, contendo o ano que você deseja localizar.

Valor retornado

Um inteiro no intervalo 1900-9999.

Comentários

Ao contrário do Microsoft Excel, que armazena as datas como números de série, o DAX usa o tipo de dados **datetime** para trabalhar com datas e horas.

As datas devem ser inseridas usando a função DATE ou como resultados de outras fórmulas ou funções. Insira também datas em representações de texto aceitas de uma data, como 3 de março de 2007 ou 3-mar-2003.

Os valores retornados pelas funções YEAR, MONTH e DAY serão valores gregorianos, independentemente do formato de exibição do valor de data fornecido. Por exemplo, se o formato de exibição da data fornecida usar o calendário islâmico, os valores retornados para as funções YEAR, MONTH e DAY serão valores associados à data gregoriana equivalente.

Quando o argumento de data é uma representação de texto da data, a função usa as configurações de localidade e de data e hora do computador cliente para reconhecer o valor de texto a fim de fazer a conversão. Poderão surgir erros se o formato das cadeias de caracteres for incompatível com as configurações de localidade atuais. Por exemplo, se a localidade definir datas a serem formatadas como mês/dia/ano e a data for fornecida como dia/mês/ano, 25/1/2009 não será interpretado como 25 de janeiro de 2009, mas como uma data inválida.

Exemplo

O exemplo a seguir retorna 2007.

```
=YEAR("March 2007")
```

Exemplo – data como resultado da expressão

Descrição

O exemplo a seguir retorna o ano da data de hoje.

=YEAR(TODAY())

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função HOUR \(DAX\)](#)

[Função MINUTE \(DAX\)](#)

[Função YEAR \(DAX\)](#)

[Função SECOND \(DAX\)](#)

YEARFRAC

12/05/2020 • 3 minutes to read

Calcula a fração do ano representada pelo número de dias inteiros existentes entre duas datas. Use a função de planilha YEARFRAC para identificar a proporção de benefícios ou de obrigações de um ano inteiro atribuídas a um período específico.

Sintaxe

```
YEARFRAC(<start_date>, <end_date>, <basis>)
```

Parâmetros

TERMO	DEFINIÇÃO
start_date	A data de início em formato datetime .
end_date	A data de término em formato datetime .
basis	(Opcional) O tipo de base de contagem de dias a ser usado. Todos os argumentos são truncados para números inteiros. Base – Descrição 0 – US (NASD) 30/360 1 – Real/real 2 – Real/360 3 – Real/365 4 – Europeu 30/360

Valor retornado

Um número decimal. O tipo de dados interno é um número de ponto flutuante de dupla precisão IEEE de 64 bits (8 bytes) assinado.

Comentários

Ao contrário do Microsoft Excel, que armazena as datas como números de série, o DAX usa o formato **datetime** para trabalhar com datas e horas. Caso precise exibir datas como números de série, você poderá usar as opções de formatação do Excel.

Se **start_date** ou **end_date** não forem datas válidas, YEARFRAC retornará um erro.

Se **basis** < 0 ou se **basis** > 4, YEARFRAC retornará um erro.

Exemplo

O exemplo a seguir retorna a fração de um ano representada pela diferença entre as datas das duas colunas,

TransactionDate e ShippingDate :

```
=YEARFRAC(Orders[TransactionDate],Orders[ShippingDate])
```

Exemplo

O exemplo a seguir retorna a fração de um ano representada pela diferença entre as datas 1º de janeiro e 1º de março:

```
=YEARFRAC("Jan 1 2007","Mar 1 2007")
```

Use anos de quatro dígitos sempre que possível, para evitar obter resultados inesperados. Quando o ano é truncado, assume-se o ano atual como padrão. Quando a data é ou omitida, assume-se a primeira data do mês como padrão.

O segundo argumento, **basis**, também foi omitido. Portanto, a fração de ano é calculada de acordo com o padrão US (NASD) 30/360.

Consulte também

[Funções de data e hora \(DAX\)](#)

[Função WEEKNUM \(DAX\)](#)

[Função YEARFRAC \(DAX\)](#)

[Função WEEKDAY \(DAX\)](#)

Funções de inteligência de dados temporais

26/05/2020 • 7 minutes to read

A linguagem DAX (Expressões de Análise de Dados) inclui funções de inteligência de dados temporais que permitem manipular dados usando períodos de tempo, inclusive dias, meses, trimestres e anos, além de criar e comparar cálculos referentes a esses períodos.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
CLOSINGBALANCEMONTH	Avalia a expressão na última data do mês no contexto atual.
CLOSINGBALANCEQUARTER	Avalia a expressão na última data do trimestre no contexto atual.
CLOSINGBALANCEYEAR	Avalia a expressão na última data do ano no contexto atual.
DATEADD	Retorna uma tabela que contém uma coluna de datas, deslocada para frente ou para trás no tempo pelo número especificado de intervalos começando nas datas do contexto atual.
DATESBETWEEN	Retorna uma tabela que contém uma coluna de datas que começa com a start_date e continua até a end_date.
DATESINPERIOD	Retorna uma tabela que contém uma coluna de datas que começa com a start_date e continua até o number_of_intervals especificado.
DATESMTD	Retorna uma tabela que contém uma coluna das datas do mês até a data, no contexto atual.
DATESQTD	Retorna uma tabela que contém uma coluna das datas do trimestre até a data, no contexto atual.
DATESYTD	Retorna uma tabela que contém uma coluna das datas do ano até a data, no contexto atual.
ENDOFMONTH	Retorna a última data do mês no contexto atual para a coluna de datas especificada.
ENDOFQUARTER	Retorna a última data do trimestre no contexto atual para a coluna de datas especificada.
ENDOFYEAR	Retorna a última data do ano no contexto atual para a coluna de datas especificada.
FIRSTDATE	Retorna a primeira data no contexto atual para a coluna de datas especificada.

FUNÇÃO	DESCRIÇÃO
FIRSTNONBLANK	Retorna o primeiro valor na coluna, column, filtrado pelo contexto atual, em que a expressão não está em branco
LASTDATE	Retorna a última data no contexto atual para a coluna de datas especificada.
LASTNONBLANK	Retorna o último valor na coluna, column, filtrada pelo contexto atual em que a expressão não está em branco.
NEXTDAY	Retorna uma tabela que contém uma coluna com todas as datas do dia seguinte, com base na primeira data especificada na coluna dates no contexto atual.
NEXTMONTH	Retorna uma tabela que contém uma coluna de todas as datas do mês seguinte, com base na primeira data da coluna dates no contexto atual.
NEXTQUARTER	Retorna uma tabela que contém uma coluna com todas as datas no próximo trimestre, com base na primeira data especificada na coluna dates, no contexto atual.
NEXTYEAR	Retorna uma tabela que contém uma coluna de todas as datas no próximo ano, com base na primeira data na coluna dates, no contexto atual.
OPENINGBALANCEMONTH	Avalia a expressão na primeira data do mês no contexto atual.
OPENINGBALANCEQUARTER	Avalia a expressão na primeira data do trimestre, no contexto atual.
OPENINGBALANCEYEAR	Avalia a expressão na primeira data do ano no contexto atual.
PARALLELPERIOD	Retorna uma tabela que contém uma coluna de datas que representa um período paralelo às datas na coluna dates especificada, no contexto atual, com as datas deslocadas em um número de intervalos para frente ou para trás no tempo.
PREVIOUSDAY	Retorna uma tabela que contém uma coluna de todas as datas que representam o dia anterior à primeira data na coluna dates, no contexto atual.
PREVIOUSMONTH	Retorna uma tabela que contém uma coluna de todas as datas do mês anterior, com base na primeira data na coluna dates, no contexto atual.
PREVIOUSQUARTER	Retorna uma tabela que contém uma coluna com todas as datas do trimestre anterior, com base na primeira data na coluna dates, no contexto atual.
PREVIOUSYEAR	Retorna uma tabela que contém uma coluna de todas as datas do ano anterior, com base na primeira data na coluna dates, no contexto atual.

FUNÇÃO	DESCRIÇÃO
SAMEPERIODLASTYEAR	Retorna uma tabela que contém uma coluna de datas deslocadas para um ano antes das datas na coluna dates especificada, no contexto atual.
STARTOFMONTH	Retorna a primeira data do mês no contexto atual para a coluna de datas especificada.
STARTOFQUARTER	Retorna a primeira data do trimestre no contexto atual para a coluna de datas especificada.
STARTOFYEAR	Retorna a primeira data do ano no contexto atual para a coluna de datas especificada.
TOTALMTD	Avalia o valor da expressão para o mês até a data, no contexto atual.
TOTALQTD	Avalia o valor da expressão para as datas do trimestre até a data, no contexto atual.
TOTALYTD	Avalia o valor do ano até a data da expressão no contexto atual.

CLOSINGBALANCEMONTH

08/05/2020 • 2 minutes to read

Avalia a **expressão** na última data do mês no contexto atual.

Sintaxe

```
CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

Parâmetros

PARÂMETRO	DEFINIÇÃO
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expression** avaliada na última data do mês no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Fim do Mês' do estoque de produtos.

Para ver como isso funciona, crie uma Tabela Dinâmica e adicione os campos CalendarYear, MonthNumberOfYear e DayNumberOfMonth à área **Rótulos de Linha** da Tabela Dinâmica. Em seguida, adicione uma medida, denominada **Valor do Estoque do Fim do Mês** usando a fórmula a seguir para a área **Valores** da Tabela Dinâmica.

```
=CLOSINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime  
[DateKey])
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEYEAR \(DAX\)](#)

[Função CLOSINGBALANCEQUARTER \(DAX\)](#)

CLOSINGBALANCEQUARTER

08/05/2020 • 2 minutes to read

Avalia a **expressão** na última data do trimestre no contexto atual.

Sintaxe

```
CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

Parâmetros

Parâmetro	Definição
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expression** avaliada na última data do trimestre no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Fim do Trimestre' do estoque

de produtos.

```
=CLOSINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEYEAR \(DAX\)](#)

[Função CLOSINGBALANCEMONTH \(DAX\)](#)

CLOSINGBALANCEYEAR

09/05/2020 • 2 minutes to read

Avalia a **expressão** na última data do ano no contexto atual.

Sintaxe

```
CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Parâmetros

Parâmetro	Definição
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Um valor escalar que representa a **expression** avaliada na última data do ano no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Fim do Ano' do estoque de produtos.

```
=CLOSINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEYEAR \(DAX\)](#)

[Função CLOSINGBALANCEQUARTER \(DAX\)](#)

[Função CLOSINGBALANCEMONTH \(DAX\)](#)

DATEADD

09/05/2020 • 3 minutes to read

Retorna uma tabela que contém uma coluna de datas, deslocada para frente ou para trás no tempo pelo número especificado de intervalos começando nas datas do contexto atual.

Sintaxe

```
DATEADD(<dates>,<number_of_intervals>,<interval>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
number_of_intervals	Um inteiro que especifica o número de intervalos a serem adicionados ou subtraídos das datas.
interval	O intervalo pelo qual as datas serão deslocadas. O valor do intervalo pode ser um dos seguintes: <code>year</code> , <code>quarter</code> , <code>month</code> ou <code>day</code> .

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Se o número especificado para **number_of_intervals** for positivo, as datas em **dates** serão avançadas no tempo; se o número for negativo, as datas em **dates** serão retrocedidas no tempo.

O parâmetro **interval** é uma enumeração, não um conjunto de cadeias de caracteres; portanto, os valores não devem ser colocados entre aspas. Além disso, os valores `year`, `quarter`, `month` e `day` devem ser escritos por extenso ao usá-los.

A tabela de resultados inclui apenas as datas que existem na coluna **dates**.

Se as datas do contexto atual não formarem um intervalo contíguo, a função retornará um erro.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo – Como deslocar um conjunto de datas

Descrição

A fórmula a seguir calcula as datas que ocorrem um ano antes das datas do contexto atual.

```
=DATEADD(DateTime[DateKey], -1, year)
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

DATESBETWEEN

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de datas que começa com a **start_date** e continua até a **end_date**.

Sintaxe

```
DATESBETWEEN(<dates>,<start_date>,<end_date>)
```

Parâmetros

Termo	Definição
dates	Uma referência a uma coluna de data/hora.
start_date	Uma expressão de data.
end_date	Uma expressão de data.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Se **start_date** for um valor de data em branco, **start_date** será o valor mais antigo na coluna **dates**.

Se **end_date** for um valor de data em branco, **end_date** será o valor mais recente na coluna **dates**.

As datas usadas como **start_date** e **end_date** são inclusivas: ou seja, se as vendas ocorrerem em 1º de setembro e você usar 1º de setembro como a data de início, as vendas de 1º de setembro serão contadas.

NOTE

A função DATESBETWEEN é fornecida para trabalhar com intervalos de datas personalizados. Se você está trabalhando com intervalos de datas comuns, como meses, trimestres e anos, recomendamos que você use a função apropriada, como DATESINPERIOD.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'Vendas do verão de 2007' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESBETWEEN(DateTime[DateKey],  
    DATE(2007,6,1),  
    DATE(2007,8,31)  
))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATESINPERIOD \(DAX\)](#)

DATESINPERIOD

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de datas que começa com a **start_date** e continua até o **number_of_intervals** especificado.

Sintaxe

```
DATESINPERIOD(<dates>,<start_date>,<number_of_intervals>,<interval>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
start_date	Uma expressão de data.
number_of_intervals	Um inteiro que especifica o número de intervalos a serem adicionados ou subtraídos das datas.
interval	O intervalo pelo qual as datas serão deslocadas. O valor do intervalo pode ser um dos seguintes: <code>year</code> , <code>quarter</code> , <code>month</code> ou <code>day</code> .

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

O argumento **dates** pode ser uma referência a uma coluna de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Se o número especificado para **number_of_intervals** for positivo, as datas serão avançadas no tempo; se o número for negativo, as datas serão retrocedidas no tempo.

O parâmetro **interval** é uma enumeração, não um conjunto de cadeias de caracteres; portanto, os valores não devem ser colocados entre aspas. Além disso, os valores `year`, `quarter`, `month` e `day` devem ser escritos por extenso ao usá-los.

A tabela de resultados inclui apenas as datas que aparecem nos valores da coluna da tabela subjacente.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula a seguir retorna as vendas na Internet dos 21 dias anteriores a 24 de agosto de 2007.

```
= CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]),DATESINPERIOD(DateTime[DateKey],DATE(2007,08,24),-21,day))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATESBETWEEN \(DAX\)](#)

DATESMTD

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna das datas do mês até a data, no contexto atual.

Sintaxe

```
DATESMTD(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor da propriedade/valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Total Acumulado do Mês' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESMTD(DateTime[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATESYTD \(DAX\)](#)

DATESQTD

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna das datas do trimestre até a data, no contexto atual.

Sintaxe

```
DATESQTD(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor da propriedade/valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booleana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Total em Execução Trimestral' de Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESQTD(DateTime[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATESYTD \(DAX\)](#)

DATESYTD

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna das datas do ano até a data, no contexto atual.

Sintaxe

```
DATESYTD(<dates> [, <year_end_date>])
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor da propriedade/valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Total Dinâmico' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), DATESYTD(DateTime[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATESMTD \(DAX\)](#)

[Função DATESQTD \(DAX\)](#)

ENDOFMONTH

09/05/2020 • 2 minutes to read

Retorna a última data do mês no contexto atual para a coluna de datas especificada.

Sintaxe

```
ENDOFMONTH(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booleana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o término do mês para o contexto atual.

```
=ENDOFMONTH(DateTime[DateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função ENDOFYEAR \(DAX\)](#)

ENDOFQUARTER

09/05/2020 • 2 minutes to read

Retorna a última data do trimestre no contexto atual para a coluna de datas especificada.

Sintaxe

```
ENDOFQUARTER(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o término do trimestre para o contexto atual.

```
=ENDOFQUARTER(DateTime[DateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função ENDOFYEAR \(DAX\)](#)

ENDOFYEAR

09/05/2020 • 2 minutes to read

Retorna a última data do ano no contexto atual para a coluna de datas especificada.

Sintaxe

```
ENDOFYEAR(<dates> [, <year_end_date>])
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o final do ano fiscal que termina em 30 de junho para o contexto atual.

```
=ENDOFYEAR(DateTime[DateKey], "06/30/2004")
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função ENDOFMONTH \(DAX\)](#)

[Função ENDOFQUARTER \(DAX\)](#)

FIRSTDATE

09/05/2020 • 2 minutes to read

Retorna a primeira data no contexto atual para a coluna de datas especificada.

Sintaxe

```
FIRSTDATE(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões boolianas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Quando o contexto atual é uma única data, a data retornada pelas funções FIRSTDATE e LASTDATE será a mesma.

O Valor retornado é uma tabela que contém uma única coluna e um único valor. Portanto, essa função pode ser usada como um argumento para qualquer função que exija uma tabela em seus argumentos. Além disso, o valor retornado pode ser usado sempre que um valor de data é necessário.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que obtém a primeira data de quando uma venda foi feita no canal de vendas pela Internet para o contexto atual.

```
=FIRSTDATE('InternetSales_USD'[SaleDateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função LASTDATE \(DAX\)](#)

[Função FIRSTNONBLANK \(DAX\)](#)

FIRSTNONBLANK

09/05/2020 • 2 minutes to read

Retorna o primeiro valor da coluna **column**, filtrada pelo contexto atual, em que a expressão não está em branco.

Sintaxe

```
FIRSTNONBLANK(<column>,<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma expressão de coluna.
expressão	Uma expressão avaliada para espaços em branco para cada valor de column .

Valor da propriedade/valor retornado

Uma tabela com uma única coluna e uma única linha contendo o primeiro valor computado.

Comentários

O argumento **column** poderá ser qualquer um dos seguintes:

- Uma referência a qualquer coluna.
- Uma tabela com uma única coluna.
- Uma expressão booliana que define uma tabela com uma única coluna.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Normalmente, essa função é usada para retornar o primeiro valor de uma coluna para a qual a expressão não está em branco. Por exemplo, você pode obter o último valor para o qual houve vendas de um produto.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Consulte também

[Função LASTNONBLANK \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

[Referência de funções do DAX](#)

FIRSTNONBLANKVALUE

08/05/2020 • 2 minutes to read

Avalia uma expressão filtrada pelos valores classificados de uma coluna e retorna o primeiro valor da expressão que não está em branco.

Sintaxe

```
FIRSTNONBLANKVALUE(<column>, <expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma coluna ou uma expressão que retorna uma tabela de coluna única.
expressão	Uma expressão avaliada para cada valor de <column>.

Valor retornado

O primeiro valor não em branco de <expression> que corresponde aos valores classificados de <column>.

Comentários

O argumento de coluna pode ser qualquer um dos seguintes:

- Uma referência a qualquer coluna.
- Uma tabela com uma única coluna.

Esta função é diferente de FIRSTNONBLANK porque <column> é adicionado ao contexto de filtro para a avaliação de <expression>.

Exemplo

A seguinte consulta DAX:

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "FNBV",
    FIRSTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

Retorna:

DIMPRODUCT[CLASS]	[FNBV]
L	699.0982
H	13778.24
M	1000.4375
	533.83

LASTDATE

09/05/2020 • 2 minutes to read

Retorna a última data no contexto atual para a coluna de datas especificada.

Sintaxe

```
LASTDATE(<dates>)
```

Parâmetros

TERMO	DEFINIÇÃO
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Quando o contexto atual é uma única data, a data retornada pelas funções FIRSTDATE e LASTDATE será a mesma.

Tecnicamente, o Valor retornado é uma tabela que contém uma única coluna e um único valor. Portanto, essa função pode ser usada como um argumento para qualquer função que exija uma tabela em seus argumentos. Além disso, o valor retornado pode ser usado sempre que um valor de data é necessário.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que obtém a última data, para o contexto atual, de quando uma venda foi feita no canal de vendas pela Internet.

Para ver como isso funciona, crie uma Tabela Dinâmica e adicione o campo CalendarYear na área de **Rótulos de Linha** da Tabela Dinâmica. Em seguida, adicione uma medida chamada **LastSaleDate**, usando a fórmula definida na seção de código, à área de **Valores** da Tabela Dinâmica.

```
=LASTDATE('InternetSales_USD'[SaleDateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função FIRSTDATE \(DAX\)](#)

[Função LASTNONBLANK \(DAX\)](#)

LASTNONBLANK

09/05/2020 • 2 minutes to read

Retorna o último valor na coluna, **column**, filtrada pelo contexto atual em que a expressão não está em branco.

Sintaxe

```
LASTNONBLANK(<column>,<expression>)
```

Parâmetros

Termo	Definição
coluna	Uma expressão de coluna.
expressão	Uma expressão avaliada para espaços em branco para cada valor de column .

Valor da propriedade/valor retornado

Uma tabela com uma única coluna e uma única linha contendo o último valor computado.

Comentários

O argumento **column** poderá ser qualquer um dos seguintes:

- Uma referência a qualquer coluna.
- Uma tabela com uma única coluna.
- Uma expressão booliana que define uma tabela com uma única coluna

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Normalmente, essa função é usada para retornar o último valor de uma coluna para a qual a expressão não está em branco. Por exemplo, você pode obter o último valor para o qual houve vendas de um produto.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Consulte também

[Função FIRSTNONBLANK \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

[Referência de funções do DAX](#)

LASTNONBLANKVALUE

08/05/2020 • 2 minutes to read

Avalia uma expressão filtrada pelos valores classificados de uma coluna e retorna o último valor da expressão que não está em branco.

Sintaxe

```
LASTNONBLANKVALUE(<column>, <expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma coluna ou uma expressão que retorna uma tabela de coluna única.
expressão	Uma expressão avaliada para cada valor de <column>.

Valor retornado

O último valor não em branco de <expression> que corresponde aos valores classificados de <column>.

Comentários

O argumento de coluna pode ser qualquer um dos seguintes:

- Uma referência a qualquer coluna.
- Uma tabela com uma única coluna.

Esta função é diferente de LASTNONBLANK porque <column> é adicionado ao contexto de filtro para a avaliação de <expression>.

Exemplo

A seguinte consulta DAX:

```
EVALUATE
SUMMARIZECOLUMNS(
    DimProduct[Class],
    "LNBV",
    LASTNONBLANKVALUE(
        DimDate[Date],
        SUM(FactInternetSales[SalesAmount])
    )
)
```

Retorna:

DIMPRODUCT[CLASS]	[LNBV]
L	132.44
H	137.6
M	84.97
	2288.6

NEXTDAY

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna com todas as datas do dia seguinte, com base na primeira data especificada na coluna **dates** no contexto atual.

Sintaxe

```
NEXTDAY(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do dia seguinte até a primeira data no parâmetro de entrada. Por exemplo, se a primeira data no argumento **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas iguais a 11 de junho de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do dia seguinte' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTDAY('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função NEXTQUARTER \(DAX\)](#)

[Função NEXTMONTH \(DAX\)](#)

[Função NEXTYEAR \(DAX\)](#)

NEXTMONTH

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de todas as datas do mês seguinte, com base na primeira data da coluna **dates** no contexto atual.

Sintaxe

```
NEXTMONTH(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do dia seguinte até a primeira data no parâmetro de entrada. Por exemplo, se a primeira data no argumento **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas do mês de julho de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do mês seguinte' para Vendas pela Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTMONTH('DateTime'[DateKey]))
```


Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função NEXTDAY \(DAX\)](#)

[Função NEXTQUARTER \(DAX\)](#)

[Função NEXTYEAR \(DAX\)](#)

NEXTQUARTER

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna com todas as datas no próximo trimestre, com base na primeira data especificada na coluna **dates**, no contexto atual.

Sintaxe

```
NEXTQUARTER(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do trimestre seguinte, com base na primeira data no parâmetro de entrada. Por exemplo, se a primeira data na coluna **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas do trimestre de julho a setembro de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do próximo trimestre' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTQUARTER('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função NEXTDAY \(DAX\)](#)

[Função NEXTMONTH \(DAX\)](#)

[Função NEXTYEAR \(DAX\)](#)

NEXTYEAR

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de todas as datas no próximo ano, com base na primeira data na coluna **dates**, no contexto atual.

Sintaxe

```
NEXTYEAR(<dates>[,<year_end_date>])
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas no próximo ano, com base na primeira data na coluna de entrada. Por exemplo, se a primeira data na coluna **dates** se refere ao ano de 2007, essa função retorna todas as datas do ano de 2008.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do próximo ano' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), NEXTYEAR('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função NEXTDAY \(DAX\)](#)

[Função NEXTQUARTER \(DAX\)](#)

[Função NEXTMONTH \(DAX\)](#)

OPENINGBALANCEMONTH

09/05/2020 • 2 minutes to read

Avalia a **expressão** na primeira data do mês no contexto atual.

Sintaxe

```
OPENINGBALANCEMONTH(<expression>,<dates>[,<filter>])
```

Parâmetros

Parâmetro	Definição
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expressão** avaliada na última data do mês no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Início do Mês' do estoque de

produtos.

```
=OPENINGBALANCEMONTH(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime  
[DateKey])
```

Consulte também

[Função OPENINGBALANCEYEAR \(DAX\)](#)

[Função OPENINGBALANCEQUARTER \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEMONTH \(DAX\)](#)

OPENINGBALANCEQUARTER

08/05/2020 • 2 minutes to read

Avalia a **expressão** na primeira data do trimestre, no contexto atual.

Sintaxe

```
OPENINGBALANCEQUARTER(<expression>,<dates>[,<filter>])
```

Parâmetros

Parâmetro	Definição
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filte*	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expressão** avaliada na primeira data do trimestre no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Início do Trimestre' do estoque

de produtos.

```
=OPENINGBALANCEQUARTER(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateT  
ime[DateKey])
```

Consulte também

[Função OPENINGBALANCEYEAR \(DAX\)](#)

[Função OPENINGBALANCEMONTH \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEQUARTER \(DAX\)](#)

OPENINGBALANCEYEAR

09/05/2020 • 2 minutes to read

Avalia a **expressão** na primeira data do ano no contexto atual.

Sintaxe

```
OPENINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Parâmetros

Parâmetro	Definição
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Um valor escalar que representa a **expressão** avaliada na última data do ano no contexto atual.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booleana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o 'Valor do Estoque de Início do Ano' do estoque de produtos.

```
=OPENINGBALANCEYEAR(SUMX(ProductInventory,ProductInventory[UnitCost]*ProductInventory[UnitsBalance]),DateTime[DateKey])
```

Consulte também

[Função OPENINGBALANCEQUARTER \(DAX\)](#)

[Função OPENINGBALANCEMONTH \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função CLOSINGBALANCEYEAR \(DAX\)](#)

PARALLELPERIOD

08/05/2020 • 4 minutes to read

Retorna uma tabela que contém uma coluna de datas que representa um período paralelo às datas na coluna **dates** especificada, no contexto atual, com as datas deslocadas em um número de intervalos para frente ou para trás no tempo.

Sintaxe

```
PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
number_of_intervals	Um inteiro que especifica o número de intervalos a serem adicionados ou subtraídos das datas.
interval	O intervalo pelo qual as datas serão deslocadas. O valor do intervalo pode ser um dos seguintes: <code>year</code> , <code>quarter</code> , <code>month</code> .

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função usa o conjunto atual de datas na coluna especificada por **dates**, desloca a primeira e a última data o número especificado de intervalos e retorna todas as datas contíguas entre as duas datas deslocadas. Se o intervalo for um intervalo parcial de mês, trimestre ou ano, os meses parciais no resultado também serão preenchidos para concluir todo o intervalo.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booleana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Se o número especificado para **number_of_intervals** for positivo, as datas em **dates** serão avançadas no tempo; se o número for negativo, as datas em **dates** serão retrocedidas no tempo.

O parâmetro **interval** é uma enumeração, não um conjunto de cadeias de caracteres; portanto, os valores não devem ser colocados entre aspas. Além disso, os valores `year`, `quarter`, `month` devem ser escritos por extenso ao usá-los.

A tabela de resultados inclui apenas as datas que aparecem nos valores da coluna da tabela subjacente.

A função `PARALLELPERIOD` é semelhante à função `DATEADD`, mas `PARALLELPERIOD` sempre retorna períodos completos no nível de granularidade especificado em vez dos períodos parciais que `DATEADD` retorna. Por exemplo, se você tiver uma seleção de datas que começa em 10 de junho e termina em 21 de junho do mesmo ano e deseja deslocar essa seleção para frente um mês, a função `PARALLELPERIOD` retornará todas as datas do mês seguinte (1º a 31 de julho). No entanto, se `DATEADD` for usado, o resultado incluirá apenas datas de 10 a 21 de julho.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as vendas do no anterior para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PARALLELPERIOD(DateTime[DateKey], -1, year))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função DATEADD \(DAX\)](#)

PREVIOUSDAY

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de todas as datas que representam o dia anterior à primeira data na coluna **dates**, no contexto atual.

Sintaxe

```
PREVIOUSDAY(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função determina a primeira data no parâmetro de entrada e, em seguida, retorna todas as datas correspondentes ao dia anterior dessa primeira data. Por exemplo, se a primeira data no argumento **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas iguais a 9 de junho de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do dia anterior' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSDAY('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função PREVIOUSMONTH \(DAX\)](#)

[Função PREVIOUSQUARTER \(DAX\)](#)

[Função PREVIOUSYEAR \(DAX\)](#)

PREVIOUSMONTH

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de todas as datas do mês anterior, com base na primeira data na coluna **dates**, no contexto atual.

Sintaxe

```
PREVIOUSMONTH(<dates>)
```

Parâmetros

Termo	Definição
Datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do mês anterior, usando a primeira data na coluna usada como entrada. Por exemplo, se a primeira data no argumento **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas do mês de maio de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do ano anterior' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSMONTH('DateTime'[DateKey]))
```


Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função PREVIOUSDAY \(DAX\)](#)

[Função PREVIOUSQUARTER \(DAX\)](#)

[Função PREVIOUSYEAR \(DAX\)](#)

PREVIOUSQUARTER

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna com todas as datas do trimestre anterior, com base na primeira data na coluna **dates**, no contexto atual.

Sintaxe

```
PREVIOUSQUARTER(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do trimestre anterior usando a primeira data na coluna de entrada. Por exemplo, se a primeira data no argumento **dates** se referir a 10 de junho de 2009, essa função retornará todas as datas do trimestre de janeiro a março de 2009.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as 'vendas do trimestre anterior' para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSQUARTER('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função PREVIOUSMONTH \(DAX\)](#)

[Função PREVIOUSDAY \(DAX\)](#)

[Função PREVIOUSYEAR \(DAX\)](#)

PREVIOUSYEAR

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de todas as datas do ano anterior, com base na primeira data na coluna **dates**, no contexto atual.

Sintaxe

```
PREVIOUSYEAR(<dates>[, <year_end_date>])
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Uma tabela que contém uma única coluna de valores de data.

Comentários

Essa função retorna todas as datas do ano anterior, considerando a data mais recente no parâmetro de entrada. Por exemplo, se a data mais recente no argumento **dates** se referir ao ano 2009, essa função retornará todas as datas para o ano de 2008, até o **year_end_date** especificado.

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data é ignorada.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as vendas do no anterior para Vendas na Internet.

```
=CALCULATE(SUM(InternetSales_USD[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função PREVIOUSMONTH \(DAX\)](#)

[Função PREVIOUSDAY \(DAX\)](#)

[Função PREVIOUSQUARTER \(DAX\)](#)

SAMEPERIODLASTYEAR

09/05/2020 • 2 minutes to read

Retorna uma tabela que contém uma coluna de datas deslocadas para um ano antes das datas na coluna **dates** especificada, no contexto atual.

Sintaxe

```
SAMEPERIODLASTYEAR(<dates>)
```

Parâmetros

Termo	Definição
dates	Uma coluna que contém datas.

Valor da propriedade/valor retornado

Uma tabela de coluna única de valores de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões boolianas são descritas no tópico [Função CALCULATE \(DAX\)](#).

As datas retornadas são as mesmas que as datas retornadas por esta fórmula equivalente:

```
DATEADD(dates, -1, year)
```

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula as vendas do ano anterior das Vendas do revendedor.

```
=CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), SAMEPERIODLASTYEAR(DateTime[DateKey]))
```

Consulte também

[Funções de inteligência de dados temporais \(DAX\)](#)

[Funções de data e hora \(DAX\)](#)

[Função PREVIOUSYEAR \(DAX\)](#)

[Função PARALLELPERIOD \(DAX\)](#)

STARTOFMONTH

09/05/2020 • 2 minutes to read

Retorna a primeira data do mês no contexto atual para a coluna de datas especificada.

Sintaxe

```
STARTOFMONTH(<dates>)
```

Parâmetros

Termo	Definição
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o início do mês para o contexto atual.

```
=STARTOFMONTH(DateTime[DateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função STARTOFYEAR \(DAX\)](#)

STARTOFQUARTER

09/05/2020 • 2 minutes to read

Retorna a primeira data do trimestre no contexto atual para a coluna de datas especificada.

Sintaxe

```
STARTOFQUARTER(<dates>)
```

Parâmetros

TERMO	DEFINIÇÃO
datas	Uma coluna que contém datas.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o início do trimestre para o contexto atual.

```
=STARTOFQUARTER(DateTime[DateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função STARTOFYEAR \(DAX\)](#)

[Função STARTOFMONTH \(DAX\)](#)

STARTOFYEAR

09/05/2020 • 2 minutes to read

Retorna a primeira data do ano no contexto atual para a coluna de datas especificada.

Sintaxe

```
STARTOFYEAR(<dates>)
```

Parâmetros

TERMO	DEFINIÇÃO
datas	Uma coluna que contém datas.
YearEndDate	(Opcional) Um valor de data de término do ano.

Valor retornado

Uma tabela que contém uma única coluna e uma única linha com um valor de data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo de fórmula a seguir cria uma medida que retorna o início do ano para o contexto atual.

```
=STARTOFYEAR(DateTime[DateKey])
```

Consulte também

[Funções de data e hora \(DAX\)](#)

[Funções de inteligência de dados temporais \(DAX\)](#)

[Função STARTOFQUARTER \(DAX\)](#)

TOTALMTD

08/05/2020 • 2 minutes to read

Avalia o valor da **expressão** para o mês até a data, no contexto atual.

Sintaxe

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

Parâmetros

PARÂMETRO	DEFINIÇÃO
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expressão** avaliada para as datas no mês atual até a data, dadas as datas em **dates**.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o "total do mês atual" ou a "soma do mês atual" de

vendas na Internet.

```
=TOTALMTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Consulte também

[Função ALL \(DAX\)](#)

[Função CALCULATE \(DAX\)](#)

[Função TOTALYTD \(DAX\)](#)

[Função TOTALQTD \(DAX\)](#)

TOTALQTD

09/05/2020 • 2 minutes to read

Avalia o valor da **expressão** para as datas do trimestre até a data, no contexto atual.

Sintaxe

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

Parâmetros

PARÂMETRO	DEFINIÇÃO
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.

Valor retornado

Um valor escalar que representa a **expressão** avaliada para todas as datas no trimestre atual até a data, determinadas as datas em **dates**.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora.
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora.
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o "total em execução trimestral" ou a "soma em

execução trimestral" de vendas na Internet.

```
=TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Consulte também

[Função ALL \(DAX\)](#)

[Função CALCULATE \(DAX\)](#)

[Função TOTALYTD \(DAX\)](#)

[Função TOTALMTD \(DAX\)](#)

Avalia o valor do ano até a data da **expressão** no contexto atual.

Sintaxe

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

Parâmetros

PARÂMETRO	DEFINIÇÃO
expressão	Uma expressão que retorna um valor escalar.
datas	Uma coluna que contém datas.
filter	(opcional) Uma expressão que especifica um filtro a ser aplicado ao contexto atual.
year_end_date	(opcional) Uma cadeia de caracteres literal com uma data que define a data de término do ano. O padrão é 31 de dezembro.

Valor retornado

Um valor escalar que representa a **expressão** avaliada para as **datas** do ano atual até a data.

Comentários

O argumento **dates** pode ser um dos seguintes:

- Uma referência a uma coluna de data/hora,
- Uma expressão de tabela que retorna uma única coluna de valores de data/hora,
- Uma expressão booliana que define uma tabela de coluna única de valores de data/hora.

NOTE

As restrições em expressões booleanas são descritas no tópico [Função CALCULATE \(DAX\)](#).

NOTE

A expressão **filter** tem as restrições descritas no tópico [Função CALCULATE \(DAX\)](#).

O parâmetro **year_end_date** é um literal de cadeia de caracteres de uma data, na mesma localidade que a usada pelo cliente em que a pasta de trabalho foi criada. A parte de ano da data não é obrigatória e é ignorada.

Por exemplo, a fórmula a seguir especifica um valor para (fiscal) **year_end_date** igual a 6/30 em uma pasta de

trabalho na localidade EN-US.

```
=TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey], ALL('DateTime'), "6/30")
```

Neste exemplo, year_end_date pode ser especificado como "6/30", "Jun 30", "30 de junho" ou qualquer cadeia de caracteres que seja resolvida para um mês/dia. No entanto, é recomendável especificar year_end_date usando "mês/dia" (como mostrado) para garantir que a cadeia de caracteres seja resolvida como uma data.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula de exemplo a seguir cria uma medida que calcula o "total do ano atual" ou a "soma do ano atual" de vendas na Internet.

```
=TOTALYTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Consulte também

[Função ALL \(DAX\)](#)

[Função CALCULATE \(DAX\)](#)

[Função DATESYTD \(DAX\)](#)

[Função TOTALMTD \(DAX\)](#)

[Função TOTALQTD \(DAX\)](#)

Funções de filtro

09/05/2020 • 4 minutes to read

As funções de filtro e valor no DAX são algumas das mais complexas e poderosas e diferem muito das funções do Excel. As funções de pesquisa funcionam usando tabelas e relações, como um banco de dados. As funções de filtragem permitem que você manipule o contexto de dados para criar cálculos dinâmicos.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
ADDMISSINGITEMS	Adiciona combinações de itens de várias colunas a uma tabela caso ainda não existam.
ALL	Retorna todas as linhas de uma tabela ou todos os valores de uma coluna, ignorando todos os filtros que estiverem aplicados.
ALLCROSSFILTERED	Limpa todos os filtros aplicados a uma tabela.
ALLEXCEPT	Remove todos os filtros de contexto na tabela, exceto filtros aplicados às colunas especificadas.
ALLNOBLANKROW	Da tabela pai de uma relação, retorna todas as linhas, exceto a linha em branco, ou todos os valores distintos de uma coluna, exceto a linha em branco, e ignora os filtros de contexto que possam existir.
ALLSELECTED	Remove filtros de contexto de colunas e linhas na consulta atual, mantendo todos os outros filtros de contexto ou filtros explícitos.
CALCULATE	Avalia uma expressão em um contexto que é modificado pelos filtros especificados.
CALCULATETABLE	Avalia uma expressão de tabela em um contexto modificado pelos filtros especificados.
CROSSFILTER	Especifica a direção de filtragem cruzada a ser usada em um cálculo para uma relação existente entre duas colunas.
DISTINCT	Retorna uma tabela de coluna única que contém os valores distintos da coluna especificada.
EARLIER	Retorna o valor atual da coluna especificada em uma etapa de avaliação externa da coluna mencionada.
EARLIEST	Retorna o valor atual da coluna especificada em uma etapa de avaliação externa da coluna especificada.

FUNÇÃO	DESCRIÇÃO
FILTER	Retorna uma tabela que representa um subconjunto de outra tabela ou expressão.
FILTERS	Retorna os valores que são aplicados diretamente como filtros para <i>columnName</i> .
HASONEFILTER	Retorna TRUE quando o número de valores filtrados diretamente em <i>columnName</i> é um; caso contrário, retorna FALSE.
HASONEVALUE	Retorna TRUE quando o contexto para <i>columnName</i> foi filtrado para apenas um valor distinto. Caso contrário, será FALSE.
ISCROSSFILTERED	Retorna TRUE quando <i>columnName</i> ou outra coluna na mesma tabela ou relacionada está sendo filtrada.
ISFILTERED	Retorna TRUE quando <i>columnName</i> está sendo filtrado diretamente.
KEEPFILTERS	Modifica como os filtros são aplicados durante a avaliação de uma função CALCULATE ou CALCULATETABLE.
RELATED	Retorna um valor relacionado de outra tabela.
RELATEDTABLE	Avalia uma expressão de tabela em um contexto modificado pelos filtros especificados.
REMOVEFILTERS	Limpa filtros das tabelas ou colunas especificadas.
SELECTEDVALUE	Retorna o valor quando o contexto para <i>columnName</i> foi filtrado para apenas um valor distinto. Caso contrário, retorna <i>alternateResult</i> .
SUBSTITUTEWITHINDEX	Retorna uma tabela que representa uma semijunção à esquerda das duas tabelas fornecidas como argumentos.
USERELATIONSHIP	Especifica a relação a ser usada em um cálculo específico como aquela que existe entre <i>columnName1</i> e <i>columnName2</i> .
VALUES	Retorna uma tabela de coluna única que contém os valores distintos da tabela ou coluna especificada.

ADDMISSINGITEMS

08/05/2020 • 4 minutes to read

Adiciona combinações de itens de várias colunas a uma tabela caso ainda não existam. A determinação de quais combinações de item adicionar baseia-se na referência de colunas de origem que contêm todos os valores possíveis para as colunas.

Para determinar as combinações de itens de colunas diferentes a serem avaliadas: AutoExist é aplicado a colunas dentro da mesma tabela, enquanto CrossJoin é aplicado entre tabelas diferentes.

A função ADDMISSINGITEMS retornará valores BLANK para as colunas IsSubtotal de linhas em branco que ele adiciona.

Sintaxe

```
ADDMISSINGITEMS(<showAllColumn>[, <showAllColumn>]..., <table>, <groupingColumn>[, <groupingColumn>]...[, filterTable]...)
```

```
ADDMISSINGITEMS(<showAllColumn>[, <showAllColumn>]..., <table>, [ROLLUPISSUBTOTAL(<groupingColumn>[, <isSubtotal_columnName>][, <groupingColumn>[, <isSubtotal_columnName>]...[])], [, filterTable]...)
```

Parâmetros

TERMO	DEFINIÇÃO
showAllColumn	Uma coluna para a qual retornar itens sem dados para as medidas usadas.
tabela	Uma tabela que contém todos os itens contendo dados (não vazios) para as medidas usadas.
groupingColumn	Uma coluna usada para agrupar no argumento de tabela fornecido.
isSubtotal_columnName	Uma coluna booliana no argumento de tabela fornecido que contém valores ISSUBTOTAL para a coluna groupingColumn correspondente.
filterTable	Uma tabela que representa os filtros a serem incluídos na lógica para determinar se devem ser adicionadas combinações específicas de itens sem dados. Usado para evitar que o ADDMISSINGITEMS adicione combinações de itens que não estão presentes porque foram removidos por um filtro.

ADDMISSINGITEMS com ROLLUPGROUP

ROLLUPGROUP é usado dentro da função ROLLUPISSUBTOTAL para refletir ROLLUPGROUPs presentes no argumento de tabela fornecido.

Restrições

- Se ROLLUPISSUBTOTAL tiver sido usado para definir o argumento de tabela fornecido (ou as linhas

equivalentes e as colunas ISSUBTOTAL tiverem sido adicionadas por algum outro meio), ROLLUPISSUBTOTAL deverá ser usado com os mesmos argumentos em ADDMISSINGITEMS. Isso também será verdadeiro para o ROLLUPGROUP se ele tiver sido usado com ROLLUPISSUBTOTAL para definir o argumento de tabela fornecido.

- A função ADDMISSINGITEMS exige que, se ROLLUPISSUBTOTAL tiver sido usado para definir o argumento de tabela fornecido, as colunas ISSUBTOTAL correspondentes a cada grupo por coluna ou ROLLUPGROUP estejam presentes no argumento de tabela fornecido. Além disso, os nomes das colunas ISSUBTOTAL devem ser fornecidos na função ROLLUPISSUBTOTAL dentro de ADDMISSINGITEMS e devem corresponder aos nomes das colunas booleanas no argumento de tabela fornecido. Isso permite que a função ADDMISSINGITEMS identifique valores em branco que se originam do fato de que uma linha é uma linha de subtotal de outros valores em branco.
- Se ROLLUPGROUP for usado com ROLLUPISSUBTOTAL para definir o argumento de tabela fornecido, exatamente um nome de coluna subtotal deverá ser fornecido por ROLLUPGROUP e deverá corresponder ao nome de coluna ISSUBTOTAL correspondente no argumento de tabela fornecido.

Exemplo

Adicione linhas em branco a colunas com "mostrar itens sem dados" ativados. A função ADDMISSINGITEMS retornará NULLs/BLANKs para as colunas IsSubtotal de linhas em branco que ela adiciona.

```
VAR 'RowHeadersShowAll' =  
CALCULATETABLE  
(  
  ADDMISSINGITEMS  
  (  
    [Sales Territory Country],  
    [Sales Territory Region],  
    'RowHeadersInCrossTab',  
    ROLLUPISSUBTOTAL  
    (  
      [Sales Territory Group],  
      [Subtotal for Sales Territory Group],  
      [Sales Territory Country],  
      [Subtotal for Sales Territory Country],  
      [Sales Territory Region],  
      [Subtotal for Sales Territory Region]  
    ),  
    'RowHeaders'  
  ),  
  'DateFilter', 'TerritoryFilter'  
)
```

Exemplo com ROLLUPGROUP

```
VAR 'RowHeadersShowAll' =  
CALCULATETABLE  
(  
ADDMISSINGITEMS  
(  
[Sales Territory Country],  
[Sales Territory Region],  
'RowHeadersInCrossTab',  
ROLLUPISSUBTOTAL  
(  
ROLLUPGROUP  
(  
[Sales Territory Group],  
[Sales Territory Country]  
),  
[Subtotal for Sales Territory Country],  
[Sales Territory Region],  
[Subtotal for Sales Territory Region]  
),  
'RowHeaders'  
)
```

ALL

08/06/2020 • 15 minutes to read

Retorna todas as linhas de uma tabela ou todos os valores de uma coluna, ignorando todos os filtros que estiverem aplicados. Essa função é útil para limpar filtros e criar cálculos em todas as linhas em uma tabela.

Sintaxe

```
ALL( [<table> | <column>[, <column>[, <column>[,...]]]] )
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela da qual você deseja limpar os filtros.
coluna	A coluna da qual você deseja limpar os filtros.

O argumento para a função ALL deve ser uma referência a uma tabela base ou uma referência a uma coluna base. Você não pode usar expressões de tabela nem expressões de coluna com a função ALL.

Valor retornado

A tabela ou coluna com filtros removidos.

Comentários

- Essa função não é usada por si só, mas serve como uma função intermediária que pode ser usada para alterar o conjunto de resultados em que algum outro cálculo é executado.
- O comportamento normal das expressões DAX que contêm a função ALL() será ignorar os filtros que forem aplicados. No entanto, há alguns cenários em que esse não é o caso devido à *auto-exist*, uma tecnologia DAX que otimiza a filtragem para reduzir a quantidade de processamento necessária para determinadas consultas DAX. A auto-exist e ALL() fornecem resultados inesperados, por exemplo, durante a filtragem em duas ou mais colunas da mesma tabela (como ao usar segmentações) e quando há uma medida na mesma tabela que usa ALL(). Nesse caso, a auto-exist *mesclará* vários filtros em um e filtrará apenas as combinações de valores existentes. Por causa dessa mesclagem, a medida será calculada com base nas combinações existentes de valores e o resultado será baseado em valores filtrados em vez de em todos os valores, como seria esperado. Para saber mais sobre a auto-exist e o efeito dela sobre os cálculos, confira o artigo do Microsoft MVP, Alberto Ferrari, [Compreensão da Auto-Exist da DAX](#), em [sql.bi.com](#).
- A tabela a seguir descreve como você pode usar as funções ALL e ALLEXCEPT em cenários diferentes.

FUNÇÃO E USO	DESCRIÇÃO
ALL()	Remove todos os filtros em todos os lugares. A função ALL() só pode ser usada para limpar filtros, mas não para retornar uma tabela.

FUNÇÃO E USO	DESCRIÇÃO
ALL(Table)	Remove todos os filtros da tabela especificada. Na verdade, ALL(table) retorna todos os valores na tabela, removendo todos os filtros do contexto que, de outra forma, poderiam ter sido aplicados. Essa função é útil quando você está trabalhando com muitos níveis de agrupamento e deseja criar um cálculo que gere uma taxa de um valor agregado para o valor total. O primeiro exemplo demonstra esse cenário.
ALL (Column[, Column[, ...]])	Remove todos os filtros das colunas especificadas na tabela. Todos os outros filtros em outras colunas na tabela ainda se aplicam. Todos os argumentos de coluna devem vir da mesma tabela. A variante ALL(Column) é útil quando você deseja remover os filtros de contexto para uma ou mais colunas específicas e manter todos os outros filtros de contexto. O segundo e o terceiro exemplos demonstram esse cenário.
ALLEXCEPT(Table, Column1 [,Column2]...)	Remove todos os filtros de contexto na tabela, exceto filtros aplicados às colunas especificadas. Esse é um atalho conveniente para situações em que você deseja remover os filtros em muitas colunas, mas não todas, em uma tabela.

Exemplo

Calcular a taxa de vendas de categoria para o total de vendas

Suponha que você queira encontrar a quantidade de vendas da célula atual, em sua Tabela Dinâmica, dividida pelo total de vendas para todos os revendedores. Para garantir que o denominador seja o mesmo, independentemente de como o usuário da Tabela Dinâmica esteja filtrando ou agrupando os dados, defina uma fórmula que use ALL para criar o total geral correto.

A tabela a seguir mostra os resultados quando uma nova medida, **Taxa de Todas as Vendas do Revendedor**, é criada usando a fórmula mostrada na seção de código. Para ver como isso funciona, adicione o campo, CalendarYear, à área **Rótulos de Linha** da Tabela Dinâmica e adicione o campo, ProductCategoryName, à área **Rótulos de Coluna**. Em seguida, arraste a medida **Taxa de Todas as Vendas do Revendedor** para a área **Valores** da Tabela Dinâmica. Para exibir os resultados como porcentagens, use os recursos de formatação do Excel para aplicar uma formatação de número percentual às células que contêm a medida.

TODAS AS VENDAS DO REVENDEDOR	RÓTULOS DE COLUNA				
Rótulos de linha	Acessórios	Bikes	Clothing	Componentes	Grande Total
2005	0,02%	9,10%	0,04%	0,75%	9,91%
2006	0,11%	24,71%	0,60%	4,48%	29,90%
2007	0,36%	31,71%	1,07%	6,79%	39,93%
2008	0,20%	16,95%	0,48%	2,63%	20,26%
Grande Total	0,70%	82,47%	2,18%	14,65%	100,00%

TODAS AS VENDAS DO REVENDEDOR	RÓTULOS DE COLUNA				
-------------------------------	-------------------	--	--	--	--

Fórmula

```
=SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/SUMX(ALL(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD])
```

A fórmula é construída da seguinte maneira:

1. O numerador, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, é a soma dos valores em `ResellerSales_USD[SalesAmount_USD]` para a célula atual na Tabela Dinâmica, com filtros de contexto aplicados em `CalendarYear` e `ProductCategoryName`.
2. Para o denominador, você começa especificando uma tabela, `ResellerSales_USD` e usa a função `ALL` para remover todos os filtros de contexto na tabela.
3. Em seguida, você usa a função `SUMX` para somar os valores na coluna `ResellerSales_USD[SalesAmount_USD]`. Em outras palavras, você obtém a soma de `ResellerSales_USD[SalesAmount_USD]` para todas as vendas de revendedores.

Exemplo 1

Calcular a Taxa de Vendas do Produto para o Total de Vendas Até o Ano Atual

Suponha que você queira criar uma tabela mostrando o percentual de vendas comparado com os anos para cada categoria de produto (`ProductCategoryName`). Para obter o percentual de cada ano em relação a cada valor de `ProductCategoryName`, você precisa dividir a soma de vendas para esse ano e categoria de produto específico pela soma de vendas da mesma categoria de produto em todos os anos. Em outras palavras, você deseja manter o filtro em `ProductCategoryName`, mas remover o filtro no ano ao calcular o denominador do percentual.

A tabela a seguir mostra os resultados quando uma nova medida, **Ano de Vendas do Revendedor**, é criada usando a fórmula mostrada na seção de código. Para ver como isso funciona, adicione o campo, `CalendarYear`, à área **Rótulos de Linha** da Tabela Dinâmica e adicione o campo, `ProductCategoryName`, à área **Rótulos de Coluna**. Para exibir os resultados como porcentagens, use os recursos de formatação do Excel para aplicar um formato de número percentual às células que contêm a medida, **Ano de Vendas do Revendedor**.

ANO DE VENDAS DO REVENDEDOR	RÓTULOS DE COLUNA				
Rótulos de linha	Acessórios	Bikes	Clothing	Componentes	Grande Total
2005	3,48%	11,03%	1,91%	5,12%	9,91%
2006	16,21%	29,96%	27,29%	30,59%	29,90%
2007	51,62%	38,45%	48,86%	46,36%	39,93%
2008	28,69%	20,56%	21,95%	17,92%	20,26%
Grande Total	100,00%	100,00%	100,00%	100,00%	100,00%

Fórmula 1

```
=SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE( SUM(
ResellerSales_USD[SalesAmount_USD]), ALL(DateTime[CalendarYear]))
```

A fórmula é construída da seguinte maneira:

- 1. O numerador, SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD]) , é a soma dos valores em ResellerSales_USD[SalesAmount_USD] para a célula atual na Tabela Dinâmica, com filtros de contexto aplicados nas colunas CalendarYear e ProductCategoryName.
- 2. Para o denominador, você remove o filtro existente em CalendarYear usando a função ALL(Column). Isso calcula a soma sobre as linhas restantes na tabela ResellerSales_USD depois de aplicar os filtros de contexto existentes dos rótulos de coluna. O efeito líquido é que, para o denominador, a soma é calculada sobre o ProductCategoryName selecionado (o filtro de contexto implícito) e para todos os valores em Year.

Exemplo 2

Calcular a contribuição de categorias de produtos para o total de vendas por ano

Suponha que você queira criar uma tabela que mostre o percentual de vendas para cada categoria de produto ano a ano. Para obter o percentual para cada categoria de produto em um ano específico, você precisa calcular a soma das vendas para essa categoria de produto específica (ProductCategoryName) no ano n e dividir o valor resultante pela soma de vendas do ano n em todas as categorias de produtos. Em outras palavras, você deseja manter o filtro no ano, mas remover o filtro em ProductCategoryName ao calcular o denominador do percentual.

A tabela a seguir mostra os resultados quando uma nova medida, **CategoryName de Vendas do Revendedor**, é criada usando a fórmula mostrada na seção de código. Para ver como isso funciona, adicione o campo, CalendarYear, à área **Rótulos de Linha** da Tabela Dinâmica e adicione o campo, ProductCategoryName, à área **Rótulos de Coluna**. Em seguida, adicione a nova medida à área **Valores** da Tabela Dinâmica. Para exibir os resultados como porcentagens, use os recursos de formatação do Excel para aplicar um formato de número percentual às células que contêm a nova medida, **CategoryName de Vendas do Revendedor**.

CATEGORYNAME DE VENDAS DO REVENDEDOR	RÓTULOS DE COLUNA				
Rótulos de linha	Acessórios	Bikes	Clothing	Componentes	Grande Total
2005	0,25%	91,76%	0,42%	7,57%	100,00%
2006	0,38%	82,64%	1,99%	14,99%	100,00%
2007	0,90%	79,42%	2,67%	17,01%	100,00%
2008	0,99%	83,69%	2,37%	12,96%	100,00%
Grande Total	0,70%	82,47%	2,18%	14,65%	100,00%

Fórmula 2

```
=SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])/CALCULATE( SUM(  
ResellerSales_USD[SalesAmount_USD]), ALL(ProductCategory[ProductCategoryName]))
```

A fórmula é construída da seguinte maneira:

1. O numerador, `SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])`, é a soma dos valores em `ResellerSales_USD[SalesAmount_USD]` para a célula atual na Tabela Dinâmica, com filtros de contexto aplicados nos campos `CalendarYear` e `ProductCategoryName`.
2. Para o denominador, use a função `ALL(Column)` para remover o filtro em `ProductCategoryName` e calcular a soma sobre as linhas restantes na tabela `ResellerSales_USD` depois de aplicar os filtros de contexto existentes dos rótulos de linha. O efeito líquido é que, para o denominador, a soma é calculada sobre o ano selecionado (o filtro de contexto implícito) e para todos os valores de `ProductCategoryName`.

Confira também

[Funções de filtro \(DAX\)](#)

[Função ALL \(DAX\)](#)

[Função ALLEXCEPT \(DAX\)](#)

[Função FILTER \(DAX\)](#)

ALLCROSSFILTERED

08/05/2020 • 2 minutes to read

Limpa todos os filtros aplicados a uma tabela.

Sintaxe

```
ALLCROSSFILTERED(<table>)
```

Parâmetros

Termo	Definição
tabela	A tabela da qual você deseja limpar os filtros.

Valor retornado

N/A. Veja os comentários.

Comentários

ALLCROSSFILTERED só pode ser usado para limpar filtros, mas não para retornar uma tabela.

Exemplo

```
DEFINE
MEASURE FactInternetSales[TotalQuantity1] =
    CALCULATE(SUM(FactInternetSales[OrderQuantity]), ALLCROSSFILTERED(FactInternetSales))
MEASURE FactInternetSales[TotalQuantity2] =
    CALCULATE(SUM(FactInternetSales[OrderQuantity]), ALL(FactInternetSales))
EVALUATE
    SUMMARIZECOLUMNS(DimSalesReason[SalesReasonName],
        "TotalQuantity1", [TotalQuantity1],
        "TotalQuantity2", [TotalQuantity2])
ORDER BY DimSalesReason[SalesReasonName]
```

Retorna

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
Evento de demonstração	60398	
Anúncio em revista	60398	
Fabricante	60398	1818
Na promoção	60398	7390

DIMSALESREASON[SALESREASONNAME]	[TOTALQUANTITY1]	[TOTALQUANTITY2]
Outros	60398	3653
Preço	60398	47733
Qualidade	60398	1551
Examinar	60398	1640
Patrocínio	60398	
Anúncio de televisão	60398	730

NOTE

Há uma relação de muitos para muitos indireta ou direta entre a tabela FactInternetSales e a tabela DimSalesReason.

ALLEXCEPT

09/05/2020 • 4 minutes to read

Remove todos os filtros de contexto na tabela, exceto filtros aplicados às colunas especificadas.

Sintaxe

```
ALLEXCEPT(<table>,<column>[,<column>[,...]])
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela sobre a qual todos os filtros de contexto são removidos, exceto filtros nas colunas especificadas em argumentos subsequentes.
coluna	A coluna para a qual os filtros de contexto devem ser preservados.

O primeiro argumento para a função ALLEXCEPT deve ser uma referência a uma tabela base; todos os argumentos subsequentes devem ser referências a colunas base. Você não pode usar expressões de tabela nem expressões de coluna com a função ALLEXCEPT.

Valor retornado

Uma tabela com todos os filtros removidos, exceto pelos filtros nas colunas especificadas.

Comentários

Essa função não é usada por si só, mas serve como uma função intermediária que pode ser usada para alterar o conjunto de resultados em que algum outro cálculo é executado.

Conforme descrito na tabela a seguir, você pode usar as funções ALL e ALLEXCEPT em cenários diferentes.

FUNÇÃO E USO	DESCRIÇÃO
ALL(Table)	Remove todos os filtros da tabela especificada. Na verdade, ALL(table) retorna todos os valores na tabela, removendo todos os filtros do contexto que, de outra forma, poderiam ter sido aplicados. Essa função é útil quando você está trabalhando com muitos níveis de agrupamento e deseja criar um cálculo que gere uma taxa de um valor agregado para o valor total.
ALL (Column[, Column[, ...]])	Remove todos os filtros das colunas especificadas na tabela. Todos os outros filtros em outras colunas na tabela ainda se aplicam. Todos os argumentos de coluna devem vir da mesma tabela. A variante ALL(Column) é útil quando você deseja remover os filtros de contexto para uma ou mais colunas específicas e manter todos os outros filtros de contexto.

FUNÇÃO E USO	DESCRIÇÃO
ALLEXCEPT(Table, Column1 [,Column2]...)	Remove todos os filtros de contexto na tabela, exceto filtros aplicados às colunas especificadas. Esse é um atalho conveniente para situações em que você deseja remover os filtros em muitas colunas, mas não todas, em uma tabela.

Exemplo

O exemplo a seguir apresenta uma fórmula que você pode usar em uma medida.

A fórmula soma SalesAmount_USD e usa a função ALLEXCEPT para remover qualquer filtro de contexto na tabela DateTime, exceto se o filtro foi aplicado à coluna CalendarYear.

```
=CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]), ALLEXCEPT(DateTime, DateTime[CalendarYear]))
```

Como a fórmula usa ALLEXCEPT, sempre que qualquer coluna, exceto CalendarYear, da tabela DateTime for usada para fatiar a Tabela Dinâmica, a fórmula removerá todos os filtros da segmentação, fornecendo um valor igual à soma de SalesAmount_USD para o valor do rótulo da coluna, conforme mostra a Tabela 1.

No entanto, se a coluna CalendarYear for usada para fatiar a Tabela Dinâmica, os resultados serão diferentes. Como CalendarYear é especificado como o argumento para ALLEXCEPT, quando os dados são fatiados no ano, um filtro é aplicado em anos no nível de linha, como mostra a Tabela 2. O usuário é incentivado a comparar essas tabelas para entender o comportamento de ALLEXCEPT().

Consulte também

[Funções de filtro \(DAX\)](#)

[Função ALL \(DAX\)](#)

[Função FILTER \(DAX\)](#)

ALLNOBLANKROW

08/05/2020 • 9 minutes to read

Da tabela pai de uma relação, retorna todas as linhas, exceto a linha em branco, ou todos os valores distintos de uma coluna, exceto a linha em branco, e ignora os filtros de contexto que possam existir.

Sintaxe

```
ALLNOBLANKROW( {<table> | <column>[, <column>[, <column>[,...]]]} )
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela sobre a qual todos os filtros de contexto são removidos.
coluna	A coluna sobre a qual todos os filtros de contexto são removidos.

Apenas um parâmetro deve ser passado; o parâmetro é uma tabela ou uma coluna.

Valor retornado

Uma tabela, quando o parâmetro passado era uma tabela, ou uma coluna de valores, quando o parâmetro passado era uma coluna.

Comentários

A função ALLNOBLANKROW apenas filtra a linha em branco que uma tabela pai, em uma relação, mostrará quando houver uma ou mais linhas na tabela filho contendo valores não correspondentes para a coluna pai. Confira o exemplo abaixo para obter uma explicação completa.

A tabela a seguir resume as variações de ALL fornecidas no DAX e suas diferenças:

FUNÇÃO E USO	DESCRIÇÃO
ALL(Column)	Remove todos os filtros das colunas especificadas na tabela. Todos os outros filtros na tabela, nas outras colunas, ainda se aplicam.
ALL(Table)	Remove todos os filtros da tabela especificada.
ALLEXCEPT(Table,Col1,Col2...)	Substitui todos os filtros de contexto na tabela, exceto nas colunas especificadas.
ALLNOBLANK(table column)	Da tabela pai de uma relação, retorna todas as linhas, exceto a linha em branco, ou todos os valores distintos de uma coluna, exceto a linha em branco, e ignora os filtros de contexto que possam existir

Para obter uma descrição geral de como a função ALL funciona, junto com exemplos passo a passo que usam ALL(Table) e ALL(Column), confira [função ALL \(DAX\)](#).

Exemplo

Nos dados de exemplo, a tabela ResellerSales_USD contém uma linha que não tem valores e, portanto, não pode estar relacionada a nenhuma das tabelas pai nas relações na pasta de trabalho. Você usará essa tabela em uma Tabela Dinâmica para que possa ver o comportamento da linha em branco e como tratar as contagens em dados não relacionados.

Etapa 1: Verificar os dados não relacionados

Abra a janela **Power Pivot** e, em seguida, selecione a tabela ResellerSales_USD. Na coluna ProductKey, filtre valores em branco. Uma linha permanecerá. Nessa linha, todos os valores de coluna devem estar em branco, exceto SalesOrderLineNumber.

Etapa 2: Criar uma Tabela Dinâmica

Crie uma nova Tabela Dinâmica, então arraste a coluna datetime.[Calendar Year] para o painel rótulos de linha. A seguinte tabela mostra os resultados esperados:

RÓTULOS DE LINHA
2005
2006
2007
2008
Total Geral

Observe o rótulo em branco entre **2008** e **Total Geral**. Esse rótulo em branco representa o membro Desconhecido, que é um grupo especial criado para considerar quaisquer valores na tabela filho que não têm valor correspondente na tabela pai, neste exemplo, a coluna datetime.[Calendar Year].

Quando você vir esse rótulo em branco na Tabela Dinâmica, saberá que, em algumas das tabelas relacionadas à coluna datetime.[Calendar Year], há valores em branco ou valores não correspondentes. A tabela pai é aquela que mostra o rótulo em branco, mas as linhas que não correspondem estão em uma ou mais das tabelas filho.

As linhas adicionadas a esse grupo de rótulos em branco são valores que não correspondem a nenhum valor na tabela pai (por exemplo, uma data que não existe na tabela datetime) ou valores nulos, ou seja, nenhum valor para a data. Neste exemplo, colocamos um valor em branco em todas as colunas da tabela de vendas filho. Ter mais valores na tabela pai do que nas tabelas filho não causa um problema.

Etapa 3: Contar linhas usando ALL e ALLNOBLANK

Adicione as duas medidas a seguir à tabela datetime para contar as linhas da tabela: **Countrows ALLNOBLANK de datetime**, **Countrows ALL de datetime**. As fórmulas que você pode usar para definir essas medidas são:

```
// Countrows ALLNOBLANK of datetime
= COUNTROWS(ALLNOBLANKROW('DateTime'))

// Countrows ALL of datetime
= COUNTROWS(ALL('DateTime'))

// Countrows ALLNOBLANKROW of ResellerSales_USD
=COUNTROWS(ALLNOBLANKROW('ResellerSales_USD'))

// Countrows ALL of ResellerSales_USD
=COUNTROWS(ALL('ResellerSales_USD'))
```

Em uma Tabela Dinâmica em branco, adicione a coluna datetime.[Calendar Year] para os rótulos de linha e, em seguida, adicione as medidas recém-criadas. Os resultados devem ser semelhantes à tabela a seguir:

RÓTULOS DE LINHA	COUNTROWS ALLNOBLANK DE DATETIME	COUNTROWS ALL DE DATETIME
2005	1280	1281
2006	1280	1281
2007	1280	1281
2008	1280	1281
	1280	1281
Total Geral	1280	1281

Os resultados mostram uma diferença de uma linha na contagem de linhas da tabela. No entanto, se você abrir a **janela Power Pivot** e selecionar a tabela datetime, não poderá encontrar nenhuma linha em branco na tabela porque a linha em branco especial mencionada aqui é o membro Desconhecido.

Etapa 4: Verifique se a contagem é precisa

Para provar que o ALLNOBLANKROW não conta todas as linhas realmente em branco e só processa a linha em branco especial na tabela pai, adicione as duas medidas a seguir à tabela ResellerSales_USD: **Countrows ALLNOBLANKROW de ResellerSales_USD**, **Countrows ALL de ResellerSales_USD**.

Crie uma nova Tabela Dinâmica e arraste a coluna datetime.[Calendar Year] para o painel rótulos de linha. Agora, adicione as medidas que você acabou de criar. Os resultados devem ser semelhantes aos seguintes:

RÓTULOS DE LINHA	COUNTROWS ALLNOBLANKROW DE RESELLERSALES_USD	COUNTROWS ALL DE RESELLERSALES_USD
2005	60856	60856
2006	60856	60856
2007	60856	60856
2008	60856	60856
	60856	60856

RÓTULOS DE LINHA	COUNTROWS ALLNOBLANKROW DE RESELLERSALES_USD	COUNTROWS ALL DE RESELLERSALES_USD
Total Geral	60856	60856

Agora, as duas medidas têm os mesmos resultados. Isso ocorre porque a função ALLNOBLANKROW não conta linhas em branco verdadeiramente em uma tabela, mas apenas manipula a linha em branco que é um caso especial gerado em uma tabela pai, quando uma ou mais das tabelas filho na relação contêm valores não correspondentes ou valores em branco.

Consulte também

[Funções de filtro \(DAX\)](#)

[Função ALL \(DAX\)](#)

[Função FILTER \(DAX\)](#)

ALLSELECTED

09/05/2020 • 8 minutes to read

Remove filtros de contexto de colunas e linhas na consulta atual, mantendo todos os outros filtros de contexto ou filtros explícitos.

A função ALLSELECTED obtém o contexto que representa todas as linhas e colunas na consulta, enquanto mantém filtros explícitos e contextos que não filtros de linha e coluna. Essa função pode ser usada para obter totais visuais em consultas.

Sintaxe

```
ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[,...]]]] )
```

Parâmetros

TERMO	DEFINIÇÃO
tableName	O nome de uma tabela existente, usando a sintaxe DAX padrão. Esse parâmetro não pode ser uma expressão. Esse parâmetro é opcional.
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão. Esse parâmetro é opcional.

Valor retornado

O contexto da consulta sem nenhum filtro de coluna e linha.

Comentários

Se houver um argumento, o argumento será *tableName* ou *columnName*. Se houver mais de um argumento, eles deverão ser colunas da mesma tabela.

Essa função é diferente de ALL() porque retém todos os filtros definidos explicitamente dentro da consulta e todos os filtros de contexto que não são filtros de linha e coluna.

Exemplo

O exemplo a seguir mostra como gerar diferentes níveis de totais visuais em um relatório de tabela usando expressões DAX. No relatório, dois (2) filtros anteriores foram aplicados aos dados de vendas do revendedor: um no Grupo de Território de Vendas = *Europa* e o outro no Tipo de Promoção = *Desconto por Volume*. Depois que os filtros forem aplicados, os totais visuais poderão ser calculados para todo o relatório, para Todos os Anos ou para Todas as Categorias de Produtos. Além disso, para fins de ilustração, o total geral para Todas as Vendas de Revendedores também é obtido, removendo todos os filtros no relatório. Avaliar a seguinte expressão DAX resulta em uma tabela com todas as informações necessárias para criar uma tabela com Totais Visuais.

```

define
measure 'Reseller Sales'[Reseller Sales Amount]=sum('Reseller Sales'[Sales Amount])
measure 'Reseller Sales'[Reseller Grand Total]=calculate(sum('Reseller Sales'[Sales Amount]), ALL('Reseller Sales'))
measure 'Reseller Sales'[Reseller Visual Total]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED())
measure 'Reseller Sales'[Reseller Visual Total for All of Calendar Year]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Date'[Calendar Year]))
measure 'Reseller Sales'[Reseller Visual Total for All of Product Category Name]=calculate(sum('Reseller Sales'[Sales Amount]), ALLSELECTED('Product Category'[Product Category Name]))
evaluate
CalculateTable(
    //CT table expression
    summarize(
    //summarize table expression
    crossjoin(distinct('Product Category'[Product Category Name]), distinct('Date'[Calendar Year]))
    //First Group by expression
    , 'Product Category'[Product Category Name]
    //Second Group by expression
    , 'Date'[Calendar Year]
    //Summary expressions
    , "Reseller Sales Amount", [Reseller Sales Amount]
    , "Reseller Grand Total", [Reseller Grand Total]
    , "Reseller Visual Total", [Reseller Visual Total]
    , "Reseller Visual Total for All of Calendar Year", [Reseller Visual Total for All of Calendar Year]
    , "Reseller Visual Total for All of Product Category Name", [Reseller Visual Total for All of Product Category Name]
    )
    //CT filters
    , 'Sales Territory'[Sales Territory Group]="Europe", 'Promotion'[Promotion Type]="Volume Discount"
    )
    order by [Product Category Name], [Calendar Year]

```

Depois de executar a expressão acima em SQL Server Management Studio no Modelo Tabular da AdventureWorks DW, você obtém os seguintes resultados:

[NOME DA CATEGORIA DE PRODUTO]	[ANO CIVIL]	[VALOR DAS VENDAS DO REVENDEDOR]	[TOTAL GERAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O ANO CIVIL]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O NOME DA CATEGORIA DE PRODUTO]
Acessórios	2000		80450596,98 23	877006,7987	38786,018	
Acessórios	2001		80450596,98 23	877006,7987	38786,018	
Acessórios	2002	625,7933	80450596,98 23	877006,7987	38786,018	91495,3104
Acessórios	2003	26037,3132	80450596,98 23	877006,7987	38786,018	572927,0136
Acessórios	2004	12122,9115	80450596,98 23	877006,7987	38786,018	212584,4747
Acessórios	2005		80450596,98 23	877006,7987	38786,018	

[NOME DA CATEGORIA DE PRODUTO]	[ANO CIVIL]	[VALOR DAS VENDAS DO REVENDEDOR]	[TOTAL GERAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O ANO CIVIL]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O NOME DA CATEGORIA DE PRODUTO]
Acessórios	2006		80450596,98 23	877006,7987	38786,018	
Bicicletas	2000		80450596,98 23	877006,7987	689287,7939	
Bicicletas	2001		80450596,98 23	877006,7987	689287,7939	
Bicicletas	2002	73778,938	80450596,98 23	877006,7987	689287,7939	91495,3104
Bicicletas	2003	439771,4136	80450596,98 23	877006,7987	689287,7939	572927,0136
Bicicletas	2004	175737,4423	80450596,98 23	877006,7987	689287,7939	212584,4747
Bicicletas	2005		80450596,98 23	877006,7987	689287,7939	
Bicicletas	2006		80450596,98 23	877006,7987	689287,7939	
Vestuário	2000		80450596,98 23	877006,7987	95090,7757	
Vestuário	2001		80450596,98 23	877006,7987	95090,7757	
Vestuário	2002	12132,4334	80450596,98 23	877006,7987	95090,7757	91495,3104
Vestuário	2003	58234,2214	80450596,98 23	877006,7987	95090,7757	572927,0136
Vestuário	2004	24724,1209	80450596,98 23	877006,7987	95090,7757	212584,4747
Vestuário	2005		80450596,98 23	877006,7987	95090,7757	
Vestuário	2006		80450596,98 23	877006,7987	95090,7757	
Componentes	2000		80450596,98 23	877006,7987	53842,2111	

[NOME DA CATEGORIA DE PRODUTO]	[ANO CIVIL]	[VALOR DAS VENDAS DO REVENDEDOR]	[TOTAL GERAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O ANO CIVIL]	[TOTAL DO VISUAL DO REVENDEDOR PARA TODO O NOME DA CATEGORIA DE PRODUTO]
Componentes	2001		80450596,98 23	877006,7987	53842,2111	
Componentes	2002	4958,1457	80450596,98 23	877006,7987	53842,2111	91495,3104
Componentes	2003	48884,0654	80450596,98 23	877006,7987	53842,2111	572927,0136
Componentes	2004		80450596,98 23	877006,7987	53842,2111	212584,4747
Componentes	2005		80450596,98 23	877006,7987	53842,2111	
Componentes	2006		80450596,98 23	877006,7987	53842,2111	

As colunas no relatório são:

Valor das Vendas do Revendedor

O valor real das Vendas do Revendedor para a categoria de ano e produto. Esse valor aparece em uma célula no centro do relatório, na interseção de ano e categoria.

Total do Visual do Revendedor para Todo o Ano Civil

O valor total para uma categoria de produto em todos os anos. Esse valor aparece no final de uma coluna ou linha de uma determinada categoria de produto e em todos os anos no relatório.

Total do Visual do Revendedor para Todo o Nome da Categoria de Produto

O valor total de um ano em todas as categorias de produto. Esse valor aparece no final de uma coluna ou linha de um determinado ano e em todas as categorias de produto no relatório.

Total do Visual do Revendedor

O valor total para todos os anos e categorias de produtos. Esse valor geralmente aparece no canto inferior direito da tabela.

Total geral do revendedor

Este é o total geral para todas as vendas do revendedor antes de qualquer filtro ter sido aplicado. Observe a diferença de [Total do Visual do Revendedor]. Você lembra que esse relatório inclui dois (2) filtros, um no grupo categoria de produto e outro no tipo de promoção.

NOTE

Se você tiver filtros explícitos em sua expressão, esses filtros também serão aplicados à expressão.

CALCULAR

08/05/2020 • 4 minutes to read

Avalia uma expressão em um contexto que é modificado pelos filtros especificados.

Sintaxe

```
CALCULATE(<expression>,<filter1>,<filter2>...)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Expressão a ser avaliada.
filter1, filter2,...	(opcional) Uma lista separada por vírgulas de expressão booliana ou uma expressão de tabela que define um filtro.

A expressão usada como o primeiro parâmetro é essencialmente a mesma que uma medida.

As seguintes restrições aplicam-se a expressões booleanas usadas como argumentos:

- A expressão não pode fazer referência a uma medida.
- A expressão não pode usar uma função CALCULATE aninhada.
- A expressão não pode usar nenhuma função que examina uma tabela ou retorna uma tabela, incluindo funções de agregação.

No entanto, uma expressão booleana pode usar qualquer função que pesquise um único valor ou que calcule um valor escalar.

Valor retornado

O valor que é o resultado da expressão.

Comentários

Se os dados tiverem sido filtrados, a função CALCULATE mudará o contexto no qual os dados são filtrados e avalia a expressão no novo contexto que você especificar. Para cada coluna usada em um argumento de filtro, todos os filtros existentes nessa coluna são removidos e o filtro usado no argumento de filtro é aplicado em vez disso.

Exemplo

Para calcular a taxa de vendas atuais do revendedor para todas as vendas do revendedor, você adiciona à Tabela Dinâmica uma medida que calcula a soma das vendas da célula atual (o numerador) e, em seguida, divide essa soma pelo total de vendas de todos os revendedores (o denominador). Para garantir que o denominador permaneça o mesmo,

independentemente de como a Tabela Dinâmica esteja filtrando ou agrupando os dados, a parte da fórmula que representa o denominador deve usar a função ALL para limpar todos os filtros e criar o total correto.

A tabela a seguir mostra os resultados quando a nova medida, chamada **Taxa de Vendas de Todos os Revendedores**, é criada usando a fórmula na seção de código.

Para ver como isso funciona, adicione o campo, CalendarYear, à área **Rótulos de Linha** da Tabela Dinâmica e adicione o campo, ProductCategoryName, à área **Rótulos de Coluna**. Em seguida, adicione a nova medida à área **Valores** da Tabela Dinâmica. Para exibir os números como porcentagens, aplique a formatação de número percentual à área da Tabela Dinâmica que contém a nova medida, **Taxa de Vendas de Todos os Revendedores**.

TODAS AS VENDAS DO REVENDEDOR	RÓTULOS DE COLUNA				
Rótulos de linha	Acessórios	Bicicletas	Vestuário	Componentes	Total Geral
2005	0,02%	9,10%	0,04%	0,75%	9,91%
2006	0,11%	24,71%	0,60%	4,48%	29,90%
2007	0,36%	31,71%	1,07%	6,79%	39,93%
2008	0,20%	16,95%	0,48%	2,63%	20,26%
Total Geral	0,70%	82,47%	2,18%	14,65%	100,00%

```
=( SUM('ResellerSales_USD'[SalesAmount_USD]))  
/CALCULATE( SUM('ResellerSales_USD'[SalesAmount_USD])  
 ,ALL('ResellerSales_USD'))
```

A expressão CALCULATE no denominador permite que a expressão sum inclua todas as linhas no cálculo. Isso substitui os filtros implícitos para CalendarYear e ProductCategoryName que existem para a parte de numerador da expressão.

Funções relacionadas

Enquanto a função CALCULATE requer como primeiro argumento uma expressão que retorne um único valor, a função CALCULATETABLE usa uma tabela de valores.

Consulte também

[Função CALCULATETABLE \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

CALCULATETABLE

08/05/2020 • 2 minutes to read

Avalia uma expressão de tabela em um contexto modificado pelos filtros especificados.

Sintaxe

```
CALCULATETABLE(<expression>,<filter1>,<filter2>,...)
```

Parâmetros

TERMO	DEFINIÇÃO
Expressão**	Expressão de tabela a ser avaliada
filter1, filter2,...	Uma expressão booliana ou uma expressão que define um filtro

A expressão usada como o primeiro parâmetro deve ser uma função que retorne uma tabela.

As seguintes restrições aplicam-se a expressões booleanas usadas como argumentos:

- A expressão não pode fazer referência a uma medida.
- A expressão não pode usar uma função CALCULATE aninhada.
- A expressão não pode usar nenhuma função que examina uma tabela ou retorna uma tabela, incluindo funções de agregação.

No entanto, uma expressão booleana pode usar qualquer função que pesquise um único valor ou que calcule um valor escalar.

Valor retornado

Uma tabela de valores.

Comentários

A função CALCULATETABLE altera o contexto no qual os dados são filtrados e avalia a expressão no novo contexto que você especificar. Para cada coluna usada em um argumento de filtro, todos os filtros existentes nessa coluna são removidos e o filtro usado no argumento de filtro é aplicado em vez disso.

Essa função é um sinônimo para a função RELATEDTABLE.

Exemplo

O exemplo a seguir usa a função CALCULATETABLE para obter a soma de vendas pela Internet para 2006. Esse valor é usado posteriormente para calcular a taxa de vendas pela Internet em comparação a todas as vendas do ano de 2006.

A seguinte fórmula:

```
=SUMX( CALCULATETABLE('InternetSales_USD', 'DateTime'[CalendarYear]=2006)
, [SalesAmount_USD])
```

Isso resulta no seguinte:

RÓTULOS DE LINHA	INTERNET SALESAMOUNT_USD	CALCULATETABLE 2006 INTERNET SALES	RAZÃO DE VENDAS PELA INTERNET PARA 2006
2005	US\$ 2.627.031,40	US\$ 5.681.440,58	0,46
2006	US\$ 5.681.440,58	US\$ 5.681.440,58	1,00
2007	US\$ 8.705.066,67	US\$ 5.681.440,58	1,53
2008	US\$ 9.041.288,80	US\$ 5.681.440,58	1,59
Total Geral	US\$ 26.054.827,45	US\$ 5.681.440,58	4,59

Consulte também

[Função RELATEDTABLE \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

CROSSFILTER

08/05/2020 • 4 minutes to read

Especifica a direção de filtragem cruzada a ser usada em um cálculo para uma relação existente entre duas colunas.

Sintaxe

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName1	O nome de uma coluna existente, usando a sintaxe do DAX padrão e totalmente qualificada, que geralmente representa o lado muitos da relação a ser usada; se os argumentos forem especificados na ordem inversa, a função os trocará antes de usá-los. Esse argumento não pode ser uma expressão.
columnName2	O nome de uma coluna existente, usando a sintaxe DAX padrão e totalmente qualificada, que geralmente representa o único lado ou o lado de pesquisa da relação a ser usado; se os argumentos forem especificados na ordem inversa, a função os trocará antes de usá-los. Esse argumento não pode ser uma expressão.
Direção	A direção do filtro cruzado a ser usada. Precisa ser uma das seguintes: nenhum Nenhuma filtragem cruzada ocorre nessa relação one – realiza a filtragem no lado um ou de pesquisa do lado da relação; filtra o lado muitos. both - Realiza a filtragem em qualquer lado; filtra o outro none – nenhuma filtragem cruzada ocorre nessa relação

Valor retornado

A função não retorna nenhum valor; a função define apenas a direção de filtragem cruzada para a relação indicada, durante o tempo da consulta.

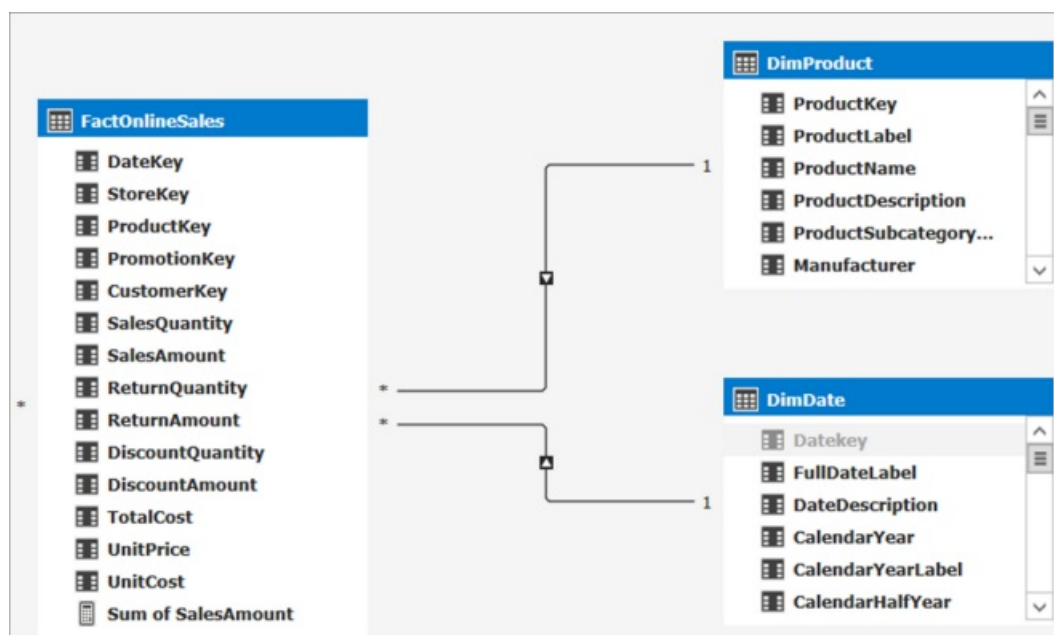
Comentários

- No caso de uma relação 1:1, não há nenhuma diferença entre uma direção e ambas as direções.
- CROSSFILTER só pode ser usado em funções que usam um filtro como argumento, por exemplo: Funções CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD e TOTALYTD.

- CROSSFILTER usa as relações existentes no modelo, identificando as relações pelas colunas de ponto final.
- Em CROSSFILTER, a configuração de filtragem cruzada de uma relação não é importante; ou seja, o fato de a relação estar definida para filtrar um ou ambos os sentidos no modelo não afeta o uso da função. CROSSFILTER substituirá qualquer configuração de filtragem cruzada existente.
- Um erro será retornado se uma das colunas nomeadas como argumento não fizer parte de uma relação ou se os argumentos pertencerem a relações diferentes.
- Se as expressões CALCULATE forem aninhadas e mais de uma expressão CALCULATE contiver uma função CROSSFILTER, o CROSSFILTER mais interno será aquele que prevalece em caso de conflito ou ambiguidade.

Exemplo

No diagrama de modelo a seguir, DimProduct e DimDate têm uma relação de direção única com FactOnlineSales.



Por padrão, não podemos obter a Contagem de Produtos vendidos por ano:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey
2005	\$3,266,373.66	606
2006	\$6,530,343.53	606
2007	\$9,791,060.30	606
2008	\$9,770,899.74	606
2009		606
2010		606
Grand Total	\$29,358,677.22	606

Há duas maneiras de obter a contagem de produtos por ano:

- Ative a filtragem cruzada bidirecional na relação. Isso alterará a forma como os filtros funcionam para todos os dados entre essas duas tabelas.
- Use a função CROSSFILTER para alterar a forma como as relações funcionam apenas para essa medida.

Ao usar o DAX, podemos usar a função CROSSFILTER para alterar a forma como a direção de filtro cruzado se comporta entre duas colunas definidas por uma relação. Nesse caso, a expressão DAX é parecida com esta:

```
BiDi:= CALCULATE([Distinct Count of ProductKey], CROSSFILTER(FactInternetSales[ProductKey],
DimProduct[ProductKey] , Both))
```

Ao usar a função CROSSFILTER em nossa expressão de medida, obteremos os resultados esperados:

Row Labels	Sum of SalesAmount	Distinct Count of ProductKey	BiDi
2005	\$3,266,373.66	606	25
2006	\$6,530,343.53	606	56
2007	\$9,791,060.30	606	133
2008	\$9,770,899.74	606	102
2009		606	
2010		606	
Grand Total	\$29,358,677.22	606	606

DISTINCT (coluna)

08/05/2020 • 4 minutes to read

Retorna uma tabela de coluna única que contém os valores distintos da coluna especificada. Em outras palavras, valores duplicados são removidos e apenas valores exclusivos são retornados.

NOTE

Esta função não pode ser usada para retornar valores em uma célula ou coluna em uma planilha; em vez disso, você aninha a função DISTINCT dentro de uma fórmula para obter uma lista de valores distintos que podem ser passados para outra função e, em seguida, contados, somados ou usados para outras operações.

Sintaxe

DISTINCT(<column>)

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna da qual os valores exclusivos serão retornados. Ou uma expressão que retorna uma coluna.

Valor retornado

Uma coluna de valores exclusivos.

Comentários

Os resultados de DISTINCT são afetados pelo contexto do filtro atual. Por exemplo, se você usar a fórmula no exemplo a seguir para criar uma medida, os resultados serão alterados sempre que a tabela for filtrada para mostrar apenas uma região ou um período específico.

Funções relacionadas

Há outra versão da função DISTINCT, [DISTINCT\(table\)](#), que retorna uma tabela removendo linhas duplicadas de outra tabela ou expressão.

A função VALUES é semelhante a DISTINCT; ela também pode ser usada para retornar uma lista de valores exclusivos e geralmente retornará exatamente os mesmos resultados que DISTINCT. No entanto, em alguns contextos, VALUES retornará um valor especial adicional. Para obter mais informações, confira [Função VALUES \(DAX\)](#).

Exemplo

A fórmula a seguir conta o número de clientes exclusivos que geraram pedidos no canal da Internet. A tabela a seguir ilustra os possíveis resultados quando a fórmula é adicionada a uma Tabela Dinâmica.


```
=COUNTROWS(DISTINCT(InternetSales_USD[CustomerKey])))
```

Não é possível colar a lista de valores que DISTINCT retorna diretamente para uma coluna. Em vez disso, você passa os resultados da função DISTINCT para outra função que conta, filtra ou agrega valores usando a lista. Para tornar o exemplo o mais simples possível, aqui a tabela de valores distintos foi passada para a função COUNTROWS.

CLIENTES ÚNICOS DA INTERNET	RÓTULOS DE COLUNA			
Rótulos de linha	Acessórios	Bicicletas	Vestuário	Total Geral
2005		1013		1013
2006		2677		2677
2007	6792	4875	2867	9309
2008	9435	5451	4196	11377
Total Geral	15114	9132	6852	18484

Além disso, observe que os resultados não são aditivos. Isso significa que o número total de clientes exclusivos em 2007 não é a soma de clientes exclusivos de *Acessórios*, *Bicicletas* e *Roupas* para aquele ano. O motivo é que um cliente pode ser contado em vários grupos.

Consulte também

[Funções de filtro \(DAX\)](#)

[Função FILTER \(DAX\)](#)

[Função RELATED \(DAX\)](#)

[Função VALUES \(DAX\)](#)

DISTINCT (tabela)

08/05/2020 • 2 minutes to read

Retorna uma tabela removendo linhas duplicadas de outra tabela ou expressão.

Sintaxe

```
DISTINCT(<table>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela da qual as linhas exclusivas devem ser retornadas. A tabela também pode ser uma expressão que resulta em uma tabela.

Valor retornado

Uma tabela que contém apenas linhas distintas.

Funções relacionadas

Há outra versão da função DISTINCT, [DISTINCT \(coluna\)](#), que usa um nome de coluna como parâmetro de entrada.

Exemplo

A consulta a seguir:

```
EVALUATE DISTINCT( { (1, "A"), (2, "B"), (1, "A") } )
```

Retorna a tabela:

[VALUE1]	[VALUE2]
1	A
2	B

Consulte também

[Funções de filtro \(DAX\)](#)

[DISTINCT \(coluna\) \(DAX\)](#)

[Função FILTER \(DAX\)](#)

[Função RELATED \(DAX\)](#)

[Função VALUES \(DAX\)](#)

EARLIER

09/05/2020 • 8 minutes to read

Retorna o valor atual da coluna especificada em uma etapa de avaliação externa da coluna mencionada.

A função EARLIER é útil para cálculos aninhados em que você deseja usar um determinado valor como entrada e produzir cálculos com base nessa entrada. No Microsoft Excel, você pode fazer esses cálculos somente dentro do contexto da linha atual. No DAX, no entanto, você pode armazenar o valor da entrada e, em seguida, efetuar cálculos usando dados de toda a tabela.

A função EARLIER é usada principalmente no contexto de colunas calculadas.

Sintaxe

```
EARLIER(<column>, <number>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma coluna ou expressão que é resolvida em uma coluna.
num	<p>(Opcional) Um número positivo para a etapa de avaliação externa.</p> <p>O próximo nível de avaliação externa será representado por 1. Dois níveis externos além serão representados por 2 e assim por diante.</p> <p>Quando omitido, assumirá o valor 1 como padrão.</p>

Valor da propriedade/valor retornado

O valor atual da linha, da **coluna**, no **número** de etapas de avaliação externas.

Exceções

Descrição de erros

Comentários

A função **EARLIER** terá sucesso se houver um contexto de linha antes do início do exame de tabela. Caso contrário, ela retornará um erro.

O desempenho da função **EARLIER** pode ser lento porque, teoricamente, ela poderá ter que executar um número de operações próximo ao número total de linhas (na coluna) vezes o mesmo número (dependendo da sintaxe da expressão). Por exemplo, se você tiver 10 linhas na coluna, poderão ser necessárias aproximadamente 100 operações; se você tiver 100 linhas, poderão ser executadas cerca de 10.000 operações.

NOTE

Na prática, o mecanismo de análise in-memory do VertiPaq executa otimizações para reduzir o número real de cálculos efetuados, mas você deve ter cuidado ao criar fórmulas que envolvem recursão.

Exemplo

Para ilustrar o uso da função EARLIER, é necessário criar um cenário que calcule um valor de classificação e, em seguida, use esse valor de classificação em outros cálculos.

O exemplo a seguir baseia-se nessa tabela simples, **ProductSubcategory**, que mostra o total de vendas de cada ProductSubcategory.

A tabela final, incluindo a coluna de classificação, é mostrada aqui.

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
18	Bretelles	US\$ 156.167,88	18
26	Racks de bicicleta	US\$ 220.720,70	14
27	Suportes de bicicleta	US\$ 35.628,69	30
28	Garrafas e compartimentos	US\$ 59.342,43	24
5	Suportes inferiores	US\$ 48.643,47	27
6	Freios	US\$ 62.113,16	23
19	Bonés	US\$ 47.934,54	28
7	Correntes	US\$ 8.847,08	35
29	Limpadores	US\$ 16.882,62	32
8	Pedaleiras	US\$ 191.522,09	15
9	Câmbios	US\$ 64.965,33	22
30	Para-choques	US\$ 41.974,10	29
10	Garfos	US\$ 74.727,66	21
20	Luvas	US\$ 228.353,58	12
4	Guidões	US\$ 163.257,06	17
11	Fones de ouvido	US\$ 57.659,99	25
31	Capacetes	US\$ 451.192,31	9
32	Conjuntos para hidratação	US\$ 96.893,78	20

PRODUCTSUBCATEGORYKEY	ENGLISHPRODUCTSUBCATEGORYNAME	TOTALSUBCATEGORYSALES	SUBCATEGORYRANKING
21	Camisas	US\$ 699.429,78	7
33	Luzes		36
34	Bloqueios	US\$ 15.059,47	33
1	Mountain bikes	US\$ 34.305.864,29	2
12	Quadros para mountain bikes	US\$ 4.511.170,68	4
35	Cestos		36
13	Pedais	US\$ 140.422,20	19
36	Bombas	US\$ 12.695,18	34
2	Bicicletas de estrada	US\$ 40.551.696,34	1
14	Quadros para bicicletas de estrada	US\$ 3.636.398,71	5
15	Selins	US\$ 52.526,47	26
22	Shorts	US\$ 385.707,80	10
23	Meias	US\$ 28.337,85	31
24	Calças	US\$ 189.179,37	16
37	Pneus e câmaras	US\$ 224.832,81	13
3	Bicicletas de passeio	US\$ 13.334.864,18	3
16	Quadros para bicicletas de passeio	US\$ 1.545.344,02	6
25	Coletes	US\$ 240.990,04	11
17	Rodas	US\$ 648.240,04	8

Criação de um valor de classificação

Uma maneira de obter um valor de classificação para um determinado valor em uma linha é contar o número de linhas na mesma tabela que contêm um valor maior (ou menor) do que a que está sendo comparada. Essa técnica retorna um valor em branco ou zero para o valor mais alto da tabela, enquanto valores iguais terão o mesmo valor de classificação e o próximo valor (após os valores iguais) terá um valor de classificação não consecutivo. Veja o exemplo abaixo.

Uma nova coluna calculada, **SubCategorySalesRanking**, é criada usando a fórmula a seguir.

```
= COUNTROWS(FILTER(ProductSubcategory, EARLIER(ProductSubcategory[TotalSubcategorySales])<ProductSubcategory[TotalSubcategorySales]))+1
```

As etapas a seguir descrevem o método de cálculo com mais detalhes.

1. A função **EARLIER** obtém o valor de *TotalSubcategorySales* para a linha atual da tabela. Nesse caso, como o processo está sendo iniciado, é a primeira linha da tabela
2. A função **EARLIER**([*TotalSubcategorySales*]) resulta em US\$ 156.167,88, a linha atual no loop externo.
3. A função **FILTER** agora retorna uma tabela em que todas as linhas têm um valor de *TotalSubcategorySales* maior que US\$ 156.167,88 (que é o valor atual da função **EARLIER**).
4. A função **COUNTROWS** conta as linhas da tabela filtrada e atribui esse valor à nova coluna calculada na linha atual mais 1. A adição de 1 é necessária para impedir que o valor de classificação superior se transforme em um espaço em branco.
5. A fórmula da coluna calculada passa para a próxima linha e repete as etapas 1 a 4. Essas etapas são repetidas até que o final da tabela seja alcançado.

A função **EARLIER** sempre obterá o valor da coluna antes da operação da tabela atual. Se você precisar obter um valor do loop antes disso, defina o segundo argumento como 2.

Consulte também

[Função EARLIEST \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

EARLIEST

09/05/2020 • 2 minutes to read

Retorna o valor atual da coluna especificada em uma etapa de avaliação externa da coluna especificada.

Sintaxe

```
EARLIEST(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma referência a uma coluna.

Valor da propriedade/valor retornado

Uma coluna com os filtros removidos.

Comentários

A função EARLIEST é semelhante à EARLIER, mas permite que você especifique um nível adicional de recursão.

Exemplo

Os dados de exemplo atuais não dão suporte a esse cenário.

```
=EARLIEST(<column>)
```

Consulte também

[Função EARLIER \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

FILTER

09/05/2020 • 4 minutes to read

Retorna uma tabela que representa um subconjunto de outra tabela ou expressão.

Sintaxe

```
FILTER(<table>,<filter>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela a ser filtrada. A tabela também pode ser uma expressão que resulta em uma tabela.
filter	Uma expressão booliana a ser avaliada para cada linha da tabela. Por exemplo, <code>[Amount] > 0</code> ou <code>[Region] = "France"</code>

Valor retornado

Uma tabela que contém apenas as linhas filtradas.

Comentários

Você pode usar FILTER para reduzir o número de linhas na tabela com a qual você está trabalhando e usar apenas dados específicos em cálculos. FILTER não é usado de maneira independente, mas como uma função que é inserida em outras funções que exigem uma tabela como um argumento.

Exemplo

O exemplo a seguir cria um relatório de vendas pela Internet fora dos Estados Unidos usando uma medida que filtra as vendas nos Estados Unidos e, em seguida, fatiando por ano civil e categorias de produtos. Para criar essa medida, filtre a tabela, Vendas pela Internet US\$, usando a Região de Vendas e, em seguida, usa a tabela filtrada em uma função SUMX.

Neste exemplo, a expressão

```
FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

 retorna uma tabela que é um subconjunto de Vendas pela Internet menos todas as linhas que pertencem à região de vendas Estados Unidos. A função RELATED é o que vincula a chave de território na tabela Vendas pela Internet para SalesTerritoryCountry na tabela SalesTerritory.

A tabela a seguir demonstra a prova de conceito da medida, Vendas pela Internet NÃO EUA, a fórmula para a qual é fornecida na seção de código abaixo. A tabela compara todas as vendas pela Internet com vendas da Internet não dos EUA, para mostrar que a expressão de filtro funciona, excluindo as vendas dos Estados Unidos do cálculo.

Para recriar essa tabela, adicione o campo, SalesTerritoryCountry, à área **Rótulos de Linha** da Tabela Dinâmica.

Tabela 1. Comparando as vendas totais para os EUA versus todas as outras regiões

RÓTULOS DE LINHA	VENDAS PELA INTERNET	VENDAS PELA INTERNET FORA DOS EUA
Austrália	US\$ 4.999.021,84	US\$ 4.999.021,84
Canadá	US\$ 1.343.109,10	US\$ 1.343.109,10
França	US\$ 2.490.944,57	US\$ 2.490.944,57
Alemanha	US\$ 2.775.195,60	US\$ 2.775.195,60
Reino Unido	US\$ 5.057.076,55	US\$ 5.057.076,55
Estados Unidos	US\$ 9.389.479,79	
Total Geral	US\$ 26.054.827,45	US\$ 16.665.347,67

A tabela de relatório final mostra os resultados quando você cria uma Tabela Dinâmica usando a medida, Vendas pela Internet NÃO EUA. Adicione o campo, CalendarYear, à área **Rótulos de Linha** da Tabela Dinâmica e adicione o campo, ProductCategoryName, à área **Rótulos de Coluna**.

Tabela 2. Comparando vendas não dos EUA por categorias de produtos

VENDAS PELA INTERNET FORA DOS EUA	RÓTULOS DE COLUNA			
Rótulos de linha	Acessórios	Bicicletas	Vestuário	Total Geral
2005		US\$ 1.526.481,95		US\$ 1.526.481,95
2006		US\$ 3.554.744,04		US\$ 3.554.744,04
2007	US\$ 156.480,18	US\$ 5.640.106,05	US\$ 70.142,77	US\$ 5.866.729,00
2008	US\$ 228.159,45	US\$ 5.386.558,19	US\$ 102.675,04	US\$ 5.717.392,68
Total Geral	US\$ 384.639,63	US\$ 16.107.890,23	US\$ 172.817,81	US\$ 16.665.347,67

```
SUMX(FILTER('InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States"), 'InternetSales_USD'[SalesAmount_USD])
```

Consulte também

- [Funções de filtro \(DAX\)](#)
- [Função ALL \(DAX\)](#)
- [Função ALLEXCEPT \(DAX\)](#)

FILTERS

09/05/2020 • 2 minutes to read

Retorna os valores que são aplicados diretamente como filtros para *columnName*.

Sintaxe

```
FILTERS(<columnName>)
```

Parâmetros

TERMO	DESCRIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

Os valores aplicados diretamente como filtros para *columnName*.

Comentários

Exemplo

O exemplo a seguir mostra como determinar o número de filtros diretos que uma coluna tem.

```
=COUNTROWS(FILTERS(ResellerSales_USD[ProductKey]))
```

O exemplo acima permite que você saiba quantos filtros diretos em ResellerSales_USD[ProductKey] foram aplicados ao contexto em que a expressão está sendo avaliada.

HASONEFILTER

09/05/2020 • 2 minutes to read

Retorna **TRUE** quando o número de valores filtrados diretamente em *columnName* é um; caso contrário, retorna **FALSE**.

Sintaxe

```
HASONEFILTER(<columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

TRUE quando o número de valores filtrados diretamente em *columnName* é um; caso contrário, retorna **FALSE**.

Comentários

1. Essa função é semelhante a HASONEVALUE() com a diferença de que HASONEVALUE() funciona com base em filtros cruzados, enquanto HASONEFILTER() funciona por um filtro direto.

Exemplo

O exemplo a seguir mostra como usar HASONEFILTER() para retornar o filtro para ResellerSales_USD[ProductKey] se houver um filtro ou para retornar BLANK se não houver filtros ou mais de um filtro em ResellerSales_USD[ProductKey]).

```
=IF(HASONEFILTER(ResellerSales_USD[ProductKey]),FILTERS(ResellerSales_USD[ProductKey]),BLANK())
```

HASONEVALUE

09/05/2020 • 3 minutes to read

Retorna **TRUE** quando o contexto para *columnName* foi filtrado para apenas um valor distinto. Caso contrário, será **FALSE**.

Sintaxe

```
HASONEVALUE(<columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

TRUE quando o contexto para *columnName* foi filtrado para apenas um valor distinto. Caso contrário, será **FALSE**.

Comentários

- Uma expressão equivalente para HASONEVALUE() é `COUNTROWS(VALUES(<columnName>)) = 1`.

Exemplo

No exemplo a seguir, você deseja criar uma fórmula que verifica se o contexto está sendo segmentado por um valor para calcular um percentual em relação a um cenário predefinido; nesse caso, você deseja comparar as vendas do revendedor com relação às vendas em 2007, então você precisa saber se o contexto é filtrado por um único ano. Além disso, se a comparação for inútil, você poderá retornar **BLANK**.

Crie uma medida chamada [ResellerSales em comparação com 2007] usando a seguinte expressão:

```
=IF(HASONEVALUE(DateTime[CalendarYear]),SUM(ResellerSales_USD[SalesAmount_USD])/CALCULATE(SUM(ResellerSales_USD[SalesAmount_USD]),DateTime[CalendarYear]=2007),BLANK())
```

- Depois de criar a medida, você deverá ter um resultado vazio em [ResellerSales em comparação com 2007]. A célula **BLANK** no resultado é porque você não tem filtros de ano único em nenhum lugar em seu contexto.
- Arraste `DateTime[CalendarYear]` para a caixa **Rótulos de Coluna**. Sua tabela deve ter a seguinte aparência:

	RÓTULOS DE COLUNA			
	2005	2006	2007	2008
ResellerSales em comparação com 2007	24,83%	74,88%	100,00%	50,73%

	RÓTULOS DE COLUNA			
--	-------------------	--	--	--

3. Arraste ProductCategory[ProductCategoryName] para a caixa **Rótulos de Linha** para ter algo assim:

RESELLERSALES EM COMPARAÇÃO COM 2007	RÓTULOS DE COLUNA			
Rótulos de Linha	2005	2006	2007	2008
Acessórios	6,74%	31,40%	100,00%	55,58%
Bicicletas	28,69%	77,92%	100,00%	53,46%
Vestuário	3,90%	55,86%	100,00%	44,92%
Componentes	11,05 %	65,99%	100,00%	38,65%
Total Geral	24,83%	74,88%	100,00%	50,73%

Você observou que **Totais Gerais** apareciam na parte inferior das colunas, mas não nas linhas? Isso ocorre porque o contexto de Totais Gerais em linhas implica mais de um ano; mas, para colunas, implica um único ano.

4. Arraste DateTime[CalendarYear] para a caixa **Segmentações Horizontais** e arraste SalesTerritory[SalesTerritoryGroup] para a caixa **Rótulos Horizontais**. Você deve ter um conjunto de resultados vazio, pois sua tabela contém dados para vários anos. Selecione **2006** na segmentação e sua tabela agora deve ter dados novamente. Tente outros anos para ver como os resultados são alterados.
5. Em suma, HASONEVALUE() permite que você identifique se a expressão está sendo avaliada no contexto de um único valor para *columnName*.

ISCROSSFILTERED

09/05/2020 • 2 minutes to read

Retorna TRUE quando *columnName* ou outra coluna na mesma tabela ou relacionada está sendo filtrada.

Sintaxe

```
ISCROSSFILTERED(<columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

TRUE quando *columnName* ou outra coluna na mesma tabela ou relacionada está sendo filtrada. Caso contrário, retorna **False**.

Comentários

- Uma coluna é considerada em filtro cruzado quando um filtro aplicado a outra coluna na mesma tabela ou em uma tabela relacionada afeta *columnName* filtrando-a. Uma coluna é considerada como filtrada *diretamente* quando o filtro ou os filtros se aplicam sobre a coluna.
- A função relacionada [ISFILTERED \(DAX\)](#) retorna TRUE quando *columnName* é filtrado diretamente.

Consulte também

[Função ISFILTERED \(DAX\)](#)

[Função FILTERS \(DAX\)](#)

[Função HASONEFILTER \(DAX\)](#)

[Função HASONEVALUE \(DAX\)](#)

ISFILTERED

09/05/2020 • 2 minutes to read

Retorna TRUE quando *columnName* está sendo filtrado diretamente. Se não houver nenhum filtro na coluna ou se a filtragem ocorrer porque uma coluna diferente da mesma tabela ou de uma tabela relacionada está sendo filtrada, então a função retornará FALSE.

Sintaxe

```
ISFILTERED(<columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

TRUE quando *columnName* estiver sendo filtrada diretamente.

Comentários

- columnName* é dita como sendo filtrada diretamente quando o(s) filtro(s) se aplica(m) sobre a coluna; uma coluna é dita sob filtro cruzado quando um filtro aplicado a outra coluna da mesma tabela ou de uma tabela relacionada afeta a coluna *columnName* filtrando-a também.
- A função relacionada [Função ISCROSSFILTERED \(DAX\)](#) retorna TRUE quando *columnName* ou outra coluna da mesma tabela ou de uma tabela relacionada está sendo filtrada.

Consulte também

[Função ISCROSSFILTERED \(DAX\)](#)

[Função FILTERS \(DAX\)](#)

[Função HASONEFILTER \(DAX\)](#)

[Função HASONEVALUE \(DAX\)](#)

KEEPFILTERS

09/05/2020 • 6 minutes to read

Modifica como os filtros são aplicados durante a avaliação de uma função CALCULATE ou CALCULATETABLE.

Sintaxe

```
KEEPFILTERS(<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Qualquer expressão.

Valor retornado

Uma tabela de valores.

Comentários

Use KEEPFILTERS nas funções de contexto CALCULATE e CALCULATETABLE, para substituir o comportamento padrão dessas funções.

Por padrão, os argumentos de filtro s em funções como CALCULATE são usados como o contexto para avaliar a expressão e, dessa forma, os argumentos de filtro da função CALCULATE substituem todos os filtros existentes nas mesmas colunas. O novo contexto efetivado pelo argumento de filtro para a função CALCULATE afeta apenas filtros existentes em colunas mencionadas como parte do argumento de filtro. Os filtros de colunas diferentes daquelas mencionados nos argumentos da função CALCULATE ou outras funções relacionadas permanecem em vigor e inalterados.

A função KEEPFILTERS permite que você modifique esse comportamento. Quando você usa KEEPFILTERS, todos os filtros existentes no contexto atual são comparados com as colunas dos argumentos de filtro e a interseção desses argumentos é usada como o contexto para avaliar a expressão. O efeito líquido em qualquer uma das colunas é que ambos os conjuntos de argumentos se aplicam: ambos os argumentos de filtro usados na função CALCULATE e os filtros usados nos argumentos da função KEEPFILTER. Em outras palavras, enquanto os filtros da CALCULATE substituem o contexto atual, a KEEPFILTERS adiciona filtros ao contexto atual.

Exemplo

O exemplo a seguir orienta você por alguns cenários comuns que demonstram o uso da função KEEPFILTERS como parte de uma fórmula CALCULATE ou CALCULATETABLE.

As três primeiras expressões obtêm dados simples a serem usados para comparações:

- Vendas pela Internet para o estado de Washington.
- Vendas pela Internet para os Estados de Washington e Oregon (ambos os Estados combinados).
- Vendas pela Internet para o estado de Washington e a província da Colúmbia Britânica (ambas as regiões combinadas).

A quarta expressão calcula as Vendas pela Internet para Washington e Oregon, enquanto o filtro para Washington e Colúmbia Britânica é aplicado.

A próxima expressão calcula as Vendas pela Internet para Washington e Oregon, mas usa a função `KEEPFILTERS`. O filtro para Washington e Colúmbia Britânica faz parte do contexto anterior.

```
EVALUATE ROW(  
  "$$ in WA"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
      , 'Geography'[State Province Code]="WA"  
    )  
  , "$$ in WA and OR"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
      , 'Geography'[State Province Code]="WA"  
      || 'Geography'[State Province Code]="OR"  
    )  
  , "$$ in WA and BC"  
    , CALCULATE('Internet Sales'[Internet Total Sales]  
      , 'Geography'[State Province Code]="WA"  
      || 'Geography'[State Province Code]="BC"  
    )  
  , "$$ in WA and OR ??"  
    , CALCULATE(  
      CALCULATE('Internet Sales'[Internet Total Sales]  
        , 'Geography'[State Province Code]="WA"  
        || 'Geography'[State Province Code]="OR"  
      )  
      , 'Geography'[State Province Code]="WA"  
      || 'Geography'[State Province Code]="BC"  
    )  
  , "$$ in WA !!"  
    , CALCULATE(  
      CALCULATE('Internet Sales'[Internet Total Sales]  
        , KEEPFILTERS('Geography'[State Province Code]="WA"  
          || 'Geography'[State Province Code]="OR"  
        )  
      )  
      , 'Geography'[State Province Code]="WA"  
      || 'Geography'[State Province Code]="BC"  
    )  
  )  
)
```

Quando essa expressão é avaliada com relação ao banco de dados de exemplo AdventureWorks DW, os resultados a seguir são obtidos.

COLUNA	VALOR
[\$\$ em WA]	US\$ 2.467.248,34
[\$\$ em WA e OR]	US\$ 3.638.239,88
[\$\$ em WA e BC]	US\$ 4.422.588,44
[\$\$ em WA e OR ??]	US\$ 3.638.239,88
[\$\$ em WA !!]	US\$ 2.467.248,34

NOTE

Os resultados acima foram formatados em uma tabela, em vez de uma única linha, para fins educacionais.

Primeiro, examine a expressão **[\$\$ em WA e OR ??]** . Você pode se perguntar como essa fórmula poderia retornar o valor de vendas para Washington e Oregon, já que a expressão CALCULATE externa inclui um filtro para Washington e Colúmbia Britânica. A resposta é que o comportamento padrão da função CALCULATE substitui os filtros externos em 'Geography'[State Province Code] e substitui seus próprios argumentos de filtro, pois os filtros se aplicam à mesma coluna.

Em seguida, examine a expressão **[\$\$ em WA !!]** . Você pode se perguntar como essa fórmula poderia retornar o valor de vendas para Washington e Oregon, já que o filtro do argumento inclui Oregon e a expressão CALCULATE externa inclui um filtro para Washington e Colúmbia Britânica. A resposta é que a função KEEPFILTERS modifica o comportamento padrão da função CALCULATE e inclui um filtro adicional. Como a interseção de filtros é usada, agora o filtro externo 'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="BC" é adicionado ao argumento de filtro 'Geography'[State Province Code]="WA" || 'Geography'[State Province Code]="OR" . Como ambos os filtros se aplicam à mesma coluna, o filtro resultante 'Geography'[State Province Code]="WA" é o filtro aplicado ao avaliar a expressão.

Consulte também

[Funções de filtro \(DAX\)](#)

[Função CALCULATE \(DAX\)](#)

[Função CALCULATETABLE \(DAX\)](#)

RELATED

08/05/2020 • 5 minutes to read

Retorna um valor relacionado de outra tabela.

Sintaxe

```
RELATED(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os valores que você deseja recuperar.

Valor retornado

Um único valor que está relacionado à linha atual.

Comentários

A função RELATED requer que exista uma relação entre a tabela atual e a tabela com informações relacionadas. Você especifica a coluna que contém os dados desejados e a função segue uma relação muitos para um existente para buscar o valor na coluna especificada na tabela relacionada. Se não existir uma relação, você precisará criar uma.

Quando a função RELATED executa uma pesquisa, ela examina todos os valores na tabela especificada, independentemente dos filtros que possam ter sido aplicados.

NOTE

A função RELATED precisa de um contexto de linha. Portanto, ela só pode ser usada na expressão de coluna calculada, em que o contexto da linha atual não é ambíguo, ou como uma função aninhada em uma expressão que usa uma função de verificação de tabela. Uma função de verificação de tabela, como SUMX, obtém o valor do valor da linha atual e examina outra tabela em busca de instâncias desse valor.

Exemplo

No exemplo a seguir, a medida de Vendas pela Internet fora dos EUA é criada para produzir um relatório de vendas que exclui as vendas no Estados Unidos. Para criar a medida, a tabela InternetSales_USD deve ser filtrada para excluir todas as vendas que pertencem ao Estados Unidos na tabela SalesTerritory. O Estados Unidos, como um país, aparecem cinco vezes na tabela SalesTerritory; uma vez para cada uma das seguintes regiões: Noroeste, Nordeste, Centro, Sudoeste e Sudeste.

A primeira abordagem para filtrar as Vendas pela Internet a fim de criar a medida pode ser adicionar uma expressão de filtro como a seguinte:

```
FILTER('InternetSales_USD', 'InternetSales_USD'[SalesTerritoryKey]<>1 && 'InternetSales_USD'[SalesTerritoryKey]<>2 && 'InternetSales_USD'[SalesTerritoryKey]<>3 && 'InternetSales_USD'[SalesTerritoryKey]<>4 && 'InternetSales_USD'[SalesTerritoryKey]<>5)
```

No entanto, essa abordagem é bem intuitiva, sujeita a erros de digitação e poderá não funcionar se alguma das regiões existentes for dividida no futuro.

Uma abordagem melhor seria usar a relação existente entre InternetSales_USD e SalesTerritory e declarar explicitamente que o país deve ser diferente dos Estados Unidos. Para fazer isso, crie uma expressão de filtro como a seguinte:

```
FILTER( 'InternetSales_USD', RELATED('SalesTerritory'[SalesTerritoryCountry])<>"United States")
```

Essa expressão usa a função RELATED para pesquisar o valor do país na tabela SalesTerritory, começando com o valor da coluna de chave, SalesTerritoryKey, na tabela InternetSales_USD. O resultado da pesquisa é usado pela função de filtro para determinar se a linha InternetSales_USD é filtrada ou não.

NOTE

Se o exemplo não funcionar, talvez seja necessário criar uma relação entre as tabelas.

```
= SUMX(FILTER( 'InternetSales_USD'
, RELATED('SalesTerritory'[SalesTerritoryCountry])
<>"United States"
)
,'InternetSales_USD'[SalesAmount_USD])
```

A tabela a seguir mostra apenas os totais de cada região, para provar que a expressão de filtro na medida, Vendas pela Internet fora dos EUA, funciona conforme o esperado.

RÓTULOS DE LINHA	VENDAS PELA INTERNET	VENDAS PELA INTERNET FORA DOS EUA
Austrália	US\$ 4.999.021,84	US\$ 4.999.021,84
Canadá	US\$ 1.343.109,10	US\$ 1.343.109,10
França	US\$ 2.490.944,57	US\$ 2.490.944,57
Alemanha	US\$ 2.775.195,60	US\$ 2.775.195,60
Reino Unido	US\$ 5.057.076,55	US\$ 5.057.076,55
Estados Unidos	US\$ 9.389.479,79	
Total Geral	US\$ 26.054.827,45	US\$ 16.665.347,67

A tabela a seguir mostra o relatório final que você poderá obter se tiver usado essa medida em uma Tabela Dinâmica:

VENDAS PELA INTERNET FORA DOS EUA	RÓTULOS DE COLUNA			
Rótulos de linha	Acessórios	Bicicletas	Vestuário	Total Geral
2005		US\$ 1.526.481,95		US\$ 1.526.481,95

VENDAS PELA INTERNET FORA DOS EUA	RÓTULOS DE COLUNA			
2006		US\$ 3.554.744,04		US\$ 3.554.744,04
2007	US\$ 156.480,18	US\$ 5.640.106,05	US\$ 70.142,77	US\$ 5.866.729,00
2008	US\$ 228.159,45	US\$ 5.386.558,19	US\$ 102.675,04	US\$ 5.717.392,68
Total Geral	US\$ 384.639,63	US\$ 16.107.890,23	US\$ 172.817,81	US\$ 16.665.347,67

Consulte também

[Função RELATEDTABLE \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

RELATEDTABLE

09/05/2020 • 2 minutes to read

Avalia uma expressão de tabela em um contexto modificado pelos filtros especificados.

Sintaxe

```
RELATEDTABLE(<tableName>)
```

Parâmetros

TERMO	DEFINIÇÃO
tableName	O nome de uma tabela existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

Uma tabela de valores.

Comentários

A função RELATEDTETABLE altera o contexto no qual os dados são filtrados e avalia a expressão no novo contexto que você especificar.

Essa função é um atalho para a função CALCULATETABLE sem nenhuma expressão lógica.

Exemplo

O exemplo a seguir usa a função RELATEDTABLE para criar uma coluna calculada com as Vendas pela Internet na tabela Categoria do Produto.

A tabela a seguir mostra os resultados do uso do código mostrado aqui.

Chave de Categoria do Produto	AlternateKey de Categoria do produto	Nome da categoria de produto	Vendas pela Internet
1	1	Bicicletas	US\$ 28.318.144,65
2	2	Componentes	
3	3	Vestuário	US\$ 339.772,61
4	4	Acessórios	US\$ 700.759,96

```
= SUMX( RELATEDTABLE('InternetSales_USD')  
    , [SalesAmount_USD])
```

Consulte também

[Função CALCULATETABLE \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

REMOVEFILTERS

09/05/2020 • 2 minutes to read

Limpar os filtros das tabelas ou colunas especificadas.

Sintaxe

```
REMOVEFILTERS([<table> | <column>[, <column>[, <column>[,...]]]])
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela da qual você deseja limpar os filtros.
coluna	A coluna da qual você deseja limpar os filtros.

Valor retornado

N/A. Veja os comentários.

Comentários

A função REMOVEFILTERS só pode ser usada para limpar filtros, mas não para retornar uma tabela.

Exemplo 1

Consulta DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales], CALCULATE([TotalSales],REMOVEFILTERS()))

EVALUATE
    SUMMARIZECOLUMNS(
        ROLLUPADDISSUBTOTAL(DimProductCategory[EnglishProductCategoryName], "IsGrandTotal"),
        "TotalSales", [TotalSales],
        "%Sales", [%Sales]
    )
ORDER BY
    [IsGrandTotal] DESC, [TotalSales] DESC
```

Retorna

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	[ISGRANDTOTAL]	[TOTALSALES]	[%SALES]
Row1	True	29.358.677,2207	1
Bicicletas	Falso	28.318.144,6507	0,964557920570538

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	[ISGRANDTOTAL]	[TOTALSALES]	[%SALES]
Acessórios	Falso	700.759,96	0,023868921434441
Vestuário	Falso	339.772,61	0,0115731579950215

Exemplo 2

Consulta DAX

```
DEFINE
MEASURE FactInternetSales[TotalSales] = SUM(FactInternetSales[SalesAmount])
MEASURE FactInternetSales[%Sales] = DIVIDE([TotalSales],
CALCULATE([TotalSales],REMOVEFILTERS(DimProductSubcategory[EnglishProductSubcategoryName])))

EVALUATE
SUMMARIZECOLUMNS(
    DimProductCategory[EnglishProductCategoryName],
    DimProductSubcategory[EnglishProductSubcategoryName],
    "TotalSales", [TotalSales],
    "%Sales", [%Sales]
)
ORDER BY
    DimProductCategory[EnglishProductCategoryName] ASC,
    DimProductSubcategory[EnglishProductSubcategoryName] ASC
```

Retorna

DIMPRODUCTCATEGORY[ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY[ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Acessórios	Racks de bicicleta	39.360	0,05616759
Acessórios	Suportes de bicicleta	39.591	0,05649723
Acessórios	Garrafas e compartimentos	56.798,19	0,08105228
Acessórios	Limpadores	7.218,6	0,0103011
Acessórios	Para-choques	46.619,58	0,06652717
Acessórios	Capacetes	225.335,6	0,3215589
Acessórios	Conjuntos para hidratação	40.307,67	0,05751994
Acessórios	Pneus e câmaras	245.529,32	0,35037578
Bicicletas	Mountain bikes	9.952.759,564	0,35146228
Bicicletas	Bicicletas de estrada	14520584,04	0,51276608
Bicicletas	Bicicletas de passeio	3.844.801,05	0,13577164

DIMPRODUCTCATEGORY [ENGLISHPRODUCTCATEGORYNAME]	DIMPRODUCTSUBCATEGORY [ENGLISHPRODUCTSUBCATEGORYNAME]	[TOTALSALES]	[%SALES]
Vestuário	Bonés	19.688,1	0,05794493
Vestuário	Luvas	35.020,7	0,10307099
Vestuário	Camisas	172.950,68	0,5090189
Vestuário	Shorts	71.319,81	0,20990453
Vestuário	Meias	5.106,32	0,01502864
Vestuário	Coletes	35.687	0,10503201

SELECTEDVALUE

09/05/2020 • 2 minutes to read

Retorna o valor quando o contexto para columnName foi filtrado para apenas um valor distinto. Caso contrário, retorna alternateResult.

Sintaxe

```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.
alternateResult	(Opcional) O valor retornado quando o contexto de columnName foi filtrado para zero ou mais de um valor distinto. Quando não fornecido, o valor padrão é BLANK().

Valor retornado

O valor quando o contexto para columnName foi filtrado para apenas um valor distinto. Caso contrário, alternateResult.

Comentários

Uma expressão equivalente para `SELECTEDVALUE(<columnName>, <alternateResult>)` é

```
IF(HASONEVALUE(<columnName>), VALUES(<columnName>), <alternateResult>).
```

Exemplo

A seguinte consulta DAX:

```
DEFINE MEASURE DimProduct[Selected Color] = SELECTEDVALUE(DimProduct[Color], "No Single Selection")
EVALUATE SUMMARIZECOLUMNS(ROLLUPADDSUBTOTAL(DimProduct[Color], "Is Total"), "Selected Color", [Selected Color])
ORDER BY [Is Total] ASC, [Color] ASC
```

Retorna o seguinte:

DIMPRODUCT[COLOR]	[É TOTAL]	[COR SELECIONADA]
Preto	FALSO	Preto
Azul	FALSO	Azul
Cinza	FALSO	Cinza

DIMPRODUCT[COLOR]	[É TOTAL]	[COR SELECIONADA]
Multi	FALSO	Multi
NA	FALSO	NA
Vermelho	FALSO	Vermelho
Prata	FALSO	Prata
Prata/Preto	FALSO	Prata/Preto
Branco	FALSO	Branco
Amarelo	FALSO	Amarelo
	VERDADEIRO	Nenhuma Seleção Única

SUBSTITUTEWITHINDEX

09/05/2020 • 4 minutes to read

Retorna uma tabela que representa uma semijunção à esquerda das duas tabelas fornecidas como argumentos. A semijunção é executada usando colunas comuns, determinadas por nomes de coluna comuns e tipo de dados comuns. As colunas que estão sendo unidas são substituídas por uma única coluna na tabela retornada, que é do tipo inteiro e contém um índice. O índice é uma referência à tabela de junção à direita, dada uma ordem de classificação especificada.

As colunas na segunda tabela à direita fornecida que não existem na primeira tabela à esquerda fornecida não estão incluídas na tabela retornada e não são usadas para junção.

O índice começa em 0 (baseado em 0) e é incrementado em um para cada linha adicional na segunda tabela de junção à direita fornecida. O índice é baseado na ordem de classificação especificada para a segunda tabela de junção à direita.

Sintaxe

```
SUBSTITUTEWITHINDEX(<table>, <indexColumnName>, <indexColumnsTable>, [<orderBy_expression>, [<order>]][, <orderBy_expression>, [<order>]]...)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Uma tabela a ser filtrada com a execução de uma semijunção à esquerda com a tabela especificada como o terceiro argumento (indexColumnsTable). Essa é a tabela no lado esquerdo da semijunção à esquerda, de modo que a tabela retornada inclui as mesmas colunas que essa tabela, exceto que todas as colunas comuns das duas tabelas serão substituídas por uma única coluna de índice na tabela retornada.
indexColumnName	Uma cadeia de caracteres que especifica o nome da coluna de índice que está substituindo todas as colunas comuns nas duas tabelas fornecidas como argumentos para essa função.
indexColumnsTable	A segunda tabela para a semijunção à esquerda. Esta é a tabela no lado direito da semijunção à esquerda. Somente os valores presentes nesta tabela serão retornados pela função. Além disso, as colunas desta tabela (com base em nomes de coluna) serão substituídas por uma única coluna de índice na tabela retornada por essa função.

TERMO	DEFINIÇÃO
orderBy_expression	Qualquer expressão DAX em que o valor de resultado é usado para especificar a ordem de classificação desejada da tabela indexColumnsTable para gerar valores de índice corretos. A ordem de classificação especificada para a tabela indexColumnsTable define o índice de cada linha na tabela e esse índice é usado na tabela retornada para representar combinações de valores no indexColumnsTable conforme eles aparecem na tabela fornecida como o primeiro argumento para essa função.
order	<p>(Opcional) Um valor que especifica como classificar valores orderBy_expression em ordem crescente ou decrescente:</p> <p>Valor: Desc. Valor alternativo: 0 (zero)/FALSE. Classifica em ordem decrescente de valores de orderBy_expression. É o valor padrão quando o parâmetro order é omitido.</p> <p>Valor: ASC. Valor alternativo: 1/TRUE. Classifica em ordem crescente de orderBy_expression.</p>

Valor retornado

Uma tabela que inclui apenas esses valores presentes na tabela indexColumnsTable e que tem uma coluna de índice em vez de todas as colunas presentes (por nome) na tabela indexColumnsTable.

Comentários

Essa função não garante nenhuma ordem de classificação de resultado.

USERELATIONSHIP

09/05/2020 • 5 minutes to read

Especifica a relação a ser usada em um cálculo específico como aquela que existe entre columnName1 e columnName2.

Sintaxe

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName1	O nome de uma coluna existente, usando a sintaxe do DAX padrão e totalmente qualificada, que geralmente representa o lado muitos da relação a ser usada; se os argumentos forem especificados na ordem inversa, a função os trocará antes de usá-los. Esse argumento não pode ser uma expressão.
columnName2	O nome de uma coluna existente, usando a sintaxe DAX padrão e totalmente qualificada, que geralmente representa o único lado ou o lado de pesquisa da relação a ser usado; se os argumentos forem especificados na ordem inversa, a função os trocará antes de usá-los. Esse argumento não pode ser uma expressão.

Valor retornado

A função não retornará nenhum valor, mas apenas habilitará a relação indicada durante o cálculo.

Comentários

- USERELATIONSHIP só pode ser usado em funções que usam um filtro como argumento, por exemplo: Funções CALCULATE, CALCULATETABLE, CLOSINGBALANCEMONTH, CLOSINGBALANCEQUARTER, CLOSINGBALANCEYEAR, OPENINGBALANCEMONTH, OPENINGBALANCEQUARTER, OPENINGBALANCEYEAR, TOTALMTD, TOTALQTD e TOTALYTD.
- USERELATIONSHIP não pode ser usado quando a segurança em nível de linha é definida para a tabela na qual a medida está incluída. Por exemplo,

```
CALCULATE(SUM([SalesAmount]), USERELATIONSHIP(FactInternetSales[CustomerKey], DimCustomer[CustomerKey]))
```

retornará um erro se a segurança em nível de linha for definida para DimCustomer.
- USERELATIONSHIP usa as relações existentes no modelo, identificando as relações pelas colunas de ponto final.
- No USERELATIONSHIP, o status de uma relação não é importante; ou seja, o fato da relação estar – ou não – ativa não afeta o uso da função. Mesmo que a relação esteja inativa, ela será usada e substituirá quaisquer outras relações ativas que possam estar presentes no modelo, mas não mencionadas nos argumentos da função.
- Um erro será retornado se uma das colunas nomeadas como argumento não fizer parte de uma relação ou

se os argumentos pertencerem a relações diferentes.

- Se forem necessárias várias relações para unir a tabela A à tabela B em um cálculo, cada relação deverá ser indicada em uma função USERELATIONSHIP diferente.
- Se as expressões CALCULATE forem aninhadas e mais de uma expressão CALCULATE contiver uma função USERELATIONSHIP, então a USERELATIONSHIP mais interna será aquela que prevalecerá em caso de conflito ou ambiguidade.
- Até 10 funções USERELATIONSHIP poderão ser aninhadas. No entanto, sua expressão poderá ter um nível mais profundo de aninhamento, ou seja, a seguinte expressão de exemplo é aninhada em 3 níveis de profundidade, mas somente 2 para USERELATIONSHIP:

```
=CALCULATE(CALCULATE( CALCULATE( &lt;anyExpression&gt;;, USERELATIONSHIP( t1[colA], t2[colB])), t99[colZ]=999), USERELATIONSHIP( t1[colA], t2[colA]))
```

Exemplo

O exemplo a seguir mostra como substituir a relação padrão ativa existente entre as tabelas InternetSales e DateTime. Existe uma relação padrão entre a coluna OrderDate, da tabela InternetSales e a coluna Date, na tabela DateTime.

Para calcular a soma das vendas pela Internet e permitir a segmentação por ShippingDate em vez da tradicional OrderDate, você precisará criar uma medida, [InternetSales by ShippingDate] usando a seguinte expressão:

```
=CALCULATE(SUM(InternetSales[SalesAmount]), USERELATIONSHIP(InternetSales[ShippingDate], DateTime[Date]))
```

Arraste a nova medida para a área de Valores no painel à direita, arraste a coluna InternetSales[ShippingDate] para a área rótulos de linha; agora você tem as vendas pela Internet segmentadas pela data de envio em vez da data do pedido, como geralmente é mostrado nestes exemplos.

Para este exemplo funcionar, as relações entre InternetSales[ShipmentDate] e DateTime[date] devem existir e não devem ser a relação ativa. Além disso, a relação entre InternetSales[OrderDate] e DateTime[Date] deve existir e ser a relação ativa.

VALUES

08/05/2020 • 6 minutes to read

Quando o parâmetro de entrada é um nome de coluna, retorna uma tabela de coluna única que contém os valores distintos da coluna especificada. Valores duplicados são removidos e apenas valores exclusivos são retornados. Um valor BLANK pode ser adicionado. Quando o parâmetro de entrada é um nome de tabela, retorna as linhas da tabela especificada. Linhas duplicadas são preservadas. Uma linha BLANK pode ser adicionada.

NOTE

Esta função não pode ser usada para retornar valores em uma célula ou coluna em uma planilha; em vez disso, você pode usá-la como uma função intermediária, aninhada em uma fórmula, para obter uma lista de valores distintos que podem ser contados ou usados para filtrar ou somar outros valores.

Sintaxe

```
VALUES(<TableNameOrColumnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
TableName ou ColumnName	Uma coluna da qual os valores únicos devem ser retornados ou uma tabela da qual as linhas devem ser retornadas.

Valor retornado

Quando o parâmetro de entrada é um nome de coluna, uma tabela de única coluna. Quando o parâmetro de entrada é um nome de tabela, uma tabela das mesmas colunas é retornada.

Comentários

Quando você usa a função VALUES em um contexto que foi filtrado, como em uma Tabela Dinâmica, os valores únicos retornados por VALUES são afetados pelo filtro. Por exemplo, se você filtrar por Região e retornar uma lista de valores para Cidade, a lista incluirá somente as cidades nas regiões permitidas pelo filtro. Para retornar todas as cidades, independentemente dos filtros existentes, você deve usar a função ALL para remover filtros da tabela. O segundo exemplo demonstra o uso de ALL com VALUES.

Funções relacionadas

Na maioria dos casos, quando o argumento é um nome de coluna, os resultados da função VALUES são idênticos aos da função DISTINCT. Ambas as funções removem duplicados e retornam uma lista dos valores possíveis na coluna especificada. No entanto, a função VALUES também pode retornar um valor em branco. Esse valor em branco é útil nos casos em que você está pesquisando valores distintos de uma tabela relacionada, mas um valor usado na relação está ausente em uma tabela. Na terminologia do banco de dados, isso é chamado de violação de integridade referencial. Tais incompatibilidades nos dados podem ocorrer quando uma tabela está sendo atualizada e a tabela relacionada não está.

Quando o argumento for um nome de tabela, o resultado da função VALUES retornará todas as linhas na tabela

especificada, além de uma linha em branco, se houver uma violação de integridade referencial. A função DISTINCT remove linhas duplicadas e retorna linhas exclusivas na tabela especificada.

NOTE

A função DISTINCT permite que um nome de coluna ou qualquer expressão de tabela válida seja seu argumento, mas a função VALUES aceita apenas um nome de coluna ou um nome de tabela como o argumento.

A tabela a seguir resume a incompatibilidade entre os dados que pode ocorrer em duas tabelas relacionadas quando a integridade referencial não é preservada.

TABELA MYORDERS	TABELA MYSALES
1º de junho	Vendas de 1º de junho
2 de junho	Vendas de 2 de junho
(nenhuma data de pedido foi inserida)	Vendas de 3 de junho

Se você tivesse usado a função DISTINCT para retornar uma lista de datas da Tabela Dinâmica que contém essas tabelas, somente duas datas seriam retornadas. No entanto, se você usar a função VALUES, a função retornará as duas datas mais um membro em branco adicional. Além disso, qualquer linha da tabela MySales que não tenha uma data correspondente na tabela MyOrders será "correspondida" a esse membro desconhecido.

Exemplo

A fórmula a seguir conta o número de faturas únicas (ordens de vendas) e produz os seguintes resultados quando usada em um relatório que inclui os Nomes de Categoria de Produto:

```
=COUNTROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

Retorna

RÓTULOS DE LINHA	CONTAR FATURAS
Acessórios	18.208
Bicicletas	15.205
Vestuário	7.461
Total Geral	27.659

Consulte também

[Função FILTER \(DAX\)](#)

[Função COUNTROWS \(DAX\)](#)

[Funções de filtro \(DAX\)](#)

Funções de informações

09/05/2020 • 2 minutes to read

As funções de informações DAX examinam a célula ou a linha fornecida como um argumento e informa se o valor corresponde ao tipo esperado. Por exemplo, a função ISERROR retornará TRUE se o valor referenciado contiver um erro.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
CONTAINS	Retornará true se os valores de todas as colunas referidas existirem ou estiverem contidas nessas colunas; caso contrário, a função retornará false.
CUSTOMDATA	Retorna o conteúdo da propriedade CustomData na cadeia de conexão.
ISBLANK	Verifica se um valor está em branco e retorna TRUE ou FALSE.
ISERROR	Verifica se um valor está errado e retorna TRUE ou FALSE.
ISEVEN	Retorna TRUE se o número é par ou FALSE se é ímpar.
ISINScope	Retorna true quando a coluna especificada é o nível em uma hierarquia de níveis.
ISLOGICAL	Verifica se um valor é lógico (TRUE ou FALSE) e retorna TRUE ou FALSE.
ISNONTEXT	Verifica se um valor não é texto (células em branco não são texto) e retorna TRUE ou FALSE.
ISNUMBER	Verifica se um valor é um número e retorna TRUE ou FALSE.
ISONORAFTER	Uma função booliana que emula o comportamento de uma cláusula Start At e retorna true para uma linha que atende a todos os parâmetros de condição.
ISTEXT	Verifica se um valor é texto e retorna TRUE ou FALSE.
LOOKUPVALUE	Retorna o valor em <i>result_columnName</i> para a linha que atende a todos os critérios especificados por <i>search_columnName</i> e <i>search_value</i> .
USERNAME	Retorna o nome de domínio e o nome de usuário das credenciais fornecidas ao sistema no momento da conexão.
USEROBJECTID	Retorna a SID ou ID de objeto do usuário atual.
USERPRINCIPALNAME	Retorna o nome principal do usuário.

CONTAINS

08/05/2020 • 2 minutes to read

Retornará true se os valores de todas as colunas referidas existirem ou estiverem contidas nessas colunas; caso contrário, a função retornará false.

Sintaxe

```
CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.
valor	Qualquer expressão DAX que retorna um único valor escalar, que deve ser procurado em <i>columnName</i> . A expressão deve ser avaliada exatamente uma vez e antes de ser passada para a lista de argumentos.

Valor retornado

Um valor igual a **TRUE** se cada *value* especificado puder ser encontrado na *columnName* correspondente ou se estiver contido nessas colunas; caso contrário, a função retornará **FALSE**.

Comentários

- Os argumentos *columnName* e *value* precisam ser fornecidos em pares; caso contrário, um erro será retornado.
- columnName* precisa pertencer à *table* especificada ou a uma tabela relacionada à *table*.
- Se *columnName* se referir a uma coluna de uma tabela relacionada, ele precisará ser totalmente qualificado; caso contrário, um erro será retornado.

Exemplo

O exemplo a seguir cria uma medida calculada que informa se houve vendas na Internet do produto 214 e para o cliente 11185 ao mesmo tempo.

```
=CONTAINS(InternetSales, [ProductKey], 214, [CustomerKey], 11185)
```

CUSTOMDATA

09/05/2020 • 2 minutes to read

Retorna o conteúdo da propriedade **CustomData** na cadeia de conexão.

Sintaxe

```
CUSTOMDATA()
```

Valor retornado

O conteúdo da propriedade **CustomData** na cadeia de conexão.

Em branco, se a propriedade **CustomData** não foi definida no momento da conexão.

Exceções

Comentários

Exemplo

O código DAX a seguir verifica se a propriedade CustomData foi definida como "OK" .

```
=IF(CUSTOMDATA()="OK", "Correct Custom data in connection string", "No custom data in connection string  
property or unexpected value")
```

Operador IN/função CONTAINSROW

09/05/2020 • 2 minutes to read

Retornará TRUE se uma linha de valores existir ou estiver contida em uma tabela; caso contrário, retornará FALSE. A sintaxe Except, o operador IN e a função CONTAINSROW são funcionalmente equivalentes.

Operador IN

Sintaxe

```
<scalarExpr> IN <tableExpr>  
( <scalarExpr1>, <scalarExpr2>, ... ) IN <tableExpr>
```

Função CONTAINSROW

Sintaxe

```
CONTAINSROW(<tableExpr>, <scalarExpr>[, <scalarExpr>, ...])
```

Parâmetros

TERMO	DEFINIÇÃO
scalarExprN	Qualquer expressão DAX válida que retorna um valor escalar.
tableExpr	Qualquer expressão DAX válida que retorna uma tabela de dados.

Valor retornado

TRUE ou FALSE.

Comentários

O número de scalarExprN deve corresponder ao número de colunas em tableExpr.

Ao contrário do operador =, o operador IN e a função CONTAINSROW executam uma comparação estrita. Por exemplo, o valor BLANK não corresponde a 0.

NOT IN não é um operador no DAX. Para executar a negação lógica do operador IN, coloque NOT na frente da expressão inteira. Por exemplo, NOT [Color] IN { "Vermelho", "Amarelo", "Azul" }.

Exemplos

Exemplo 1

As seguintes consultas DAX equivalentes:

```
EVALUATE FILTER(ALL(DimProduct[Color]), [Color] IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

e

```
EVALUATE FILTER(ALL(DimProduct[Color]), ([Color]) IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

e

```
EVALUATE FILTER(ALL(DimProduct[Color]), CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

Retornam a tabela a seguir com uma única coluna:

DIMPRODUCT[COLOR]	
Azul	
Vermelho	
Amarelo	

Exemplo 2

As seguintes consultas DAX equivalentes:

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), ([Color], [Size]) IN { ("Black", "L") })
```

e

```
EVALUATE FILTER(SUMMARIZE(DimProduct, [Color], [Size]), CONTAINSROW({ ("Black", "L") }, [Color], [Size]))
```

Retornam:

DIMPRODUCT[COLOR]	DIMPRODUCT[SIZE]
Preto	L

Exemplo 3

As seguintes consultas DAX equivalentes:

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT [Color] IN { "Red", "Yellow", "Blue" })
ORDER BY [Color]
```

e

```
EVALUATE FILTER(ALL(DimProduct[Color]), NOT CONTAINSROW({ "Red", "Yellow", "Blue" }, [Color]))
ORDER BY [Color]
```

Retornam a tabela a seguir com uma única coluna:

DIMPRODUCT[COLOR]	
Preto	
Cinza	
Multi	
NA	
Prata	
Prata\Preto	
Branco	

ISBLANK

08/05/2020 • 2 minutes to read

Verifica se um valor está em branco e retorna TRUE ou FALSE.

Sintaxe

```
ISBLANK(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor ou a expressão que você deseja testar.

Valor retornado

Um valor booliano de TRUE se o valor estiver em branco; caso contrário, FALSE.

Exemplo

Esta fórmula computa a taxa de aumento ou diminuição em vendas em comparação com o ano anterior. O exemplo usa a função IF para verificar o valor das vendas do ano anterior a fim de evitar um erro de divisão por zero.

```
//Sales to Previous Year Ratio

=IF( ISBLANK('CalculatedMeasures'[PreviousYearTotalSales])
, BLANK()
, ( 'CalculatedMeasures'[Total Sales]-'CalculatedMeasures'[PreviousYearTotalSales] )
/'CalculatedMeasures'[PreviousYearTotalSales])
```

Resultado

RÓTULOS DE LINHA	TOTAL DE VENDAS	TOTAL DE VENDAS NO ANO ANTERIOR	TAXA DE VENDAS PARA O ANO ANTERIOR
2005	US\$ 10.209.985,08		
2006	US\$ 28.553.348,43	US\$ 10.209.985,08	179,66%
2007	US\$ 39.248.847,52	US\$ 28.553.348,43	37,46%
2008	US\$ 24.542.444,68	US\$ 39.248.847,52	-37,47%
Total Geral	US\$ 102.554.625,71		

Consulte também

ISERROR

09/05/2020 • 2 minutes to read

Verifica se um valor está errado e retorna TRUE ou FALSE.

Sintaxe

```
ISERROR(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor que você deseja testar.

Valor retornado

Um valor booliano de TRUE se o valor for um erro; caso contrário, FALSE.

Exemplo

O exemplo a seguir calcula a taxa do total de vendas pela Internet para o total de vendas do revendedor. A função ISERROR é usada para verificar se há erros, como divisão por zero. Se houver um erro, um espaço em branco será retornado; caso contrário, a taxa será retornada.

```
= IF( ISERROR(  
    SUM('ResellerSales_USD'[SalesAmount_USD])  
    /SUM('InternetSales_USD'[SalesAmount_USD])  
    )  
    , BLANK()  
    , SUM('ResellerSales_USD'[SalesAmount_USD])  
    /SUM('InternetSales_USD'[SalesAmount_USD])  
    )
```

Consulte também

[Funções de informações \(DAX\)](#)

[Função IFERROR \(DAX\)](#)

[Função IF \(DAX\)](#)

ISEVEN

08/05/2020 • 2 minutes to read

Retorna TRUE se o número é par ou FALSE se é ímpar.

Sintaxe

```
ISEVEN(number)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Valor a ser testado. Se o número não for um inteiro, ele será truncado.

Valor retornado

Retorna TRUE se o número é par ou FALSE se é ímpar.

Comentários

Se o número for não numérico, ISEVEN retornará o valor de erro #VALUE!.

ISINSCOPE

09/05/2020 • 2 minutes to read

Retorna true quando a coluna especificada é o nível em uma hierarquia de níveis.

Sintaxe

```
ISINSCOPE(<columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.

Valor retornado

TRUE quando a coluna especificada é o nível em uma hierarquia de níveis.

Exemplo

```

DEFINE
MEASURE FactInternetSales[% of Parent] =
    SWITCH (TRUE(),
        ISINSCOPE(DimProduct[Subcategory]),
            DIVIDE(
                SUM(FactInternetSales[Sales Amount]),
                CALCULATE(
                    SUM(FactInternetSales[Sales Amount]),
                    ALLSELECTED(DimProduct[Subcategory]))
            ),
        ISINSCOPE(DimProduct[Category]),
            DIVIDE(
                SUM(FactInternetSales[Sales Amount]),
                CALCULATE(
                    SUM(FactInternetSales[Sales Amount]),
                    ALLSELECTED(DimProduct[Category]))
            ),
        1
    ) * 100
EVALUATE
    SUMMARIZECOLUMNS
    (
        ROLLUPADDISSUBTOTAL
    (
        DimProduct[Category], "Category Subtotal",
        DimProduct[Subcategory], "Subcategory Subtotal"
    ),
    TREATAS(
        {"Bike Racks", "Bike Stands", "Mountain Bikes", "Road Bikes", "Touring Bikes"},
        DimProduct[Subcategory]),
    "Sales", SUM(FactInternetSales[Sales Amount]),
    "% of Parent", [% of Parent]
    )
ORDER BY
    [Category Subtotal] DESC, [Category],
    [Subcategory Subtotal] DESC, [Subcategory]

```

Retorna

DIMPRODUCT[C ATEGORY]	DIMPRODUCT[SU BCATEGORY]	[SUBTOTAL DA CATEGORIA]	[SUBTOTAL DA SUBCATEGORIA]	[VENDAS]	[% DO PAI]
		VERDADEIRO	VERDADEIRO	28.397.095,65	100,00
Acessórios		FALSO	VERDADEIRO	78.951,00	0,28
Acessórios	Racks de bicicleta	FALSO	FALSO	39.360,00	49,85
Acessórios	Suportes de bicicleta	FALSO	FALSO	39.591,00	50,15
Bicicletas		FALSO	VERDADEIRO	28.318.144,65	99,72
Bicicletas	Mountain bikes	FALSO	FALSO	9.952.759,56	35,15
Bicicletas	Bicicletas de estrada	FALSO	FALSO	14.520.584,04	51,28
Bicicletas	Bicicletas de passeio	FALSO	FALSO	3.844.801,05	13,58

Consulte também

[Função SUMMARIZECOLUMNS \(DAX\)](#)

[Função CALCULATE \(DAX\)](#)

ISLOGICAL

09/05/2020 • 2 minutes to read

Verifica se um valor é lógico (TRUE ou FALSE) e retorna TRUE ou FALSE.

Sintaxe

```
ISLOGICAL(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor que você deseja testar.

Valor da propriedade/valor retornado

TRUE se o valor for um valor lógico; FALSE se for qualquer valor diferente de TRUE ou FALSE.

Exemplo

Os três exemplos a seguir mostram o comportamento da função ISLOGICAL.

```
//RETURNS: Is Boolean type or Logical
=IF(ISLOGICAL(true), "Is Boolean type or Logical", "Is different type")

//RETURNS: Is Boolean type or Logical
=IF(ISLOGICAL(false), "Is Boolean type or Logical", "Is different type")

//RETURNS: Is different type
=IF(ISLOGICAL(25), "Is Boolean type or Logical", "Is different type")
```

Consulte também

[Funções de informações \(DAX\)](#)

ISNONTEXT

08/05/2020 • 2 minutes to read

Verifica se um valor não é texto (células em branco não são texto) e retorna TRUE ou FALSE.

Sintaxe

```
ISNONTEXT(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor que você quer verificar.

Valor retornado

TRUE se o valor não for texto ou em branco; FALSE se o valor for texto.

Comentários

Uma cadeia de caracteres vazia é considerada texto.

Exemplo

Os exemplos a seguir mostram o comportamento da função ISNONTEXT.

```
//RETURNS: Is Non-Text
=IF(ISNONTEXT(1), "Is Non-Text", "Is Text")

//RETURNS: Is Non-Text
=IF(ISNONTEXT(BLANK()), "Is Non-Text", "Is Text")

//RETURNS: Is Text
=IF(ISNONTEXT(""), "Is Non-Text", "Is Text")
```

Consulte também

[Funções de informações \(DAX\)](#)

ISNUMBER

09/05/2020 • 2 minutes to read

Verifica se um valor é um número e retorna TRUE ou FALSE.

Sintaxe

```
ISNUMBER(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor que você deseja testar.

Valor da propriedade/valor retornado

TRUE se o valor for numérico; caso contrário, FALSE.

Exemplo

Os três exemplos a seguir mostram o comportamento da função ISNUMBER.

```
//RETURNS: Is number
=IF(ISNUMBER(0), "Is number", "Is Not number")

//RETURNS: Is number
=IF(ISNUMBER(3.1E-1), "Is number", "Is Not number")

//RETURNS: Is Not number
=IF(ISNUMBER("123"), "Is number", "Is Not number")
```

Consulte também

[Funções de informações \(DAX\)](#)

ISODD

09/05/2020 • 2 minutes to read

Retornará TRUE se o número for ímpar ou FALSE se o número for par.

Sintaxe

```
ISODD(number)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Valor a ser testado. Se o número não for um inteiro, ele será truncado.

Valor retornado

Retornará TRUE se o número for ímpar ou FALSE se o número for par.

Comentários

Se número for um valor não numérico, a função ISODD retornará o valor de erro #VALUE!.

ISONORAFTER

09/05/2020 • 2 minutes to read

Uma função booliana que emula o comportamento de uma cláusula “Start At” e retorna true para uma linha que atende a todos os parâmetros de condição.

Essa função usa um número variável de triplos, os dois primeiros valores de um triplo são as expressões a serem comparadas e o terceiro parâmetro indica a ordem de classificação. A ordem de classificação pode ser crescente (padrão) ou decrescente.

Com base na ordem de classificação, o primeiro parâmetro é comparado com o segundo. Se a ordem de classificação for crescente, a comparação a ser feita será de primeiro parâmetro maior ou igual ao segundo parâmetro. Se a ordem de classificação for decrescente, a comparação a ser feita será de segundo parâmetro inferior ou igual ao primeiro parâmetro.

Sintaxe

```
ISONORAFTER(<scalar_expression>, <scalar_expression>[, sort_order [, <scalar_expression>, <scalar_expression>[, sort_order]]...)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão escalar	Qualquer expressão que retorna um valor escalar, como uma referência de coluna ou um valor de número inteiro ou de cadeia de caracteres. Normalmente, o primeiro parâmetro é uma referência de coluna e o segundo parâmetro é um valor escalar.
sort order	(opcional) A ordem na qual a coluna é classificada. Pode ser crescente (ASC) ou decrescente (DEC). Por padrão, a ordem de classificação é crescente.

Valor retornado

True ou false.

Exemplo

Nome da tabela: 'Info'

PAÍS	ESTADO	CONTAGEM	TOTAL
IND	JK	20	800
IND`	MH	25	1000
IND	WB	10	900

PAÍS	ESTADO	CONTAGEM	TOTAL
EUA	AC	5	500
EUA	WA	10	900

`FILTER(Info, ISONORAFTER(Info[Country], "IND", ASC, Info[State], "MH", ASC))`

ISTEXT

08/05/2020 • 2 minutes to read

Verifica se um valor é texto e retorna TRUE ou FALSE.

Sintaxe

```
ISTEXT(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	O valor que você quer verificar.

Valor da propriedade/valor retornado

TRUE se o valor for texto; caso contrário, FALSE

Exemplo

Os exemplos a seguir mostram o comportamento da função ISTEXT.

```
//RETURNS: Is Text
=IF(ISTEXT("text"), "Is Text", "Is Non-Text")

//RETURNS: Is Text
=IF(ISTEXT(""), "Is Text", "Is Non-Text")

//RETURNS: Is Non-Text
=IF(ISTEXT(1), "Is Text", "Is Non-Text")

//RETURNS: Is Non-Text
=IF(ISTEXT(BLANK()), "Is Text", "Is Non-Text")
```

Consulte também

[Funções de informações \(DAX\)](#)

LOOKUPVALUE

08/05/2020 • 2 minutes to read

Retorna o valor em *result_columnName* para a linha que atende a todos os critérios especificados por *search_columnName* e *search_value*.

Sintaxe

```
LOOKUPVALUE( <result_columnName>, <search_columnName>, <search_value>[, <search_columnName>, <search_value>]...  
[, <alternateResult>])
```

Parâmetros

TERMO	DEFINIÇÃO
result_columnName	O nome de uma coluna existente que contém o valor que você deseja retornar. A coluna precisa ser nomeada usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.
search_columnName	O nome de uma coluna existente, na mesma tabela que result_columnName ou em uma tabela relacionada, na qual a pesquisa é executada. A coluna precisa ser nomeada usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.
search_value	Uma expressão escalar que não faz referência a nenhuma coluna na mesma tabela que está sendo pesquisada.
alternateResult	(Opcional) O valor retornado quando o contexto de result_columnName foi filtrado para zero ou mais de um valor distinto. Quando o valor não é fornecido, a função retorna BLANK() quando result_columnName é filtrado para um valor zero ou um erro quando há mais de um valor distinto.

Valor retornado

O valor de *result_column* na linha em que todos os pares de *search_column* e *search_value* têm uma correspondência.

Se não houver uma correspondência que satisfaça todos os valores de pesquisa, BLANK ou *alternateResult*, se fornecido, será retornado. Em outras palavras, a função não retornará um valor de pesquisa se apenas alguns dos critérios corresponderem.

Se várias linhas corresponderem aos valores de pesquisa e, em todos os casos, os valores de *result_column* forem idênticos, esse valor será retornado. No entanto, se *result_column* retornar valores diferentes, um erro ou *alternateResult*, se fornecido, será retornado.

Comentários

Exemplo

O exemplo a seguir retorna o SafetyStockLevel para o modelo de bicicleta "Mountain-400-W Silver, 46".

```
=LOOKUPVALUE(Product[SafetyStockLevel], [ProductName], " Mountain-400-W Silver, 46")
```

USERNAME

09/05/2020 • 2 minutes to read

Retorna o nome de domínio e o nome de usuário das credenciais fornecidas ao sistema no momento da conexão.

Sintaxe

```
USERNAME()
```

Parâmetros

Essa expressão não tem parâmetros.

Valor retornado

O nome de usuário das credenciais fornecidas ao sistema no momento da conexão

Exemplo

O código a seguir verifica se o logon do usuário faz parte da UsersTable.

```
=IF(CONTAINS(UsersTable,UsersTable[login], USERNAME()), "Allowed", BLANK())
```

USEROBJECTID

08/05/2020 • 2 minutes to read

Retorna a SID (ID de segurança) ou a ID de objeto do usuário atual do Azure AD.

Sintaxe

```
USEROBJECTID()
```

Parâmetros

Essa expressão não tem parâmetros.

Valor retornado

A ID de objeto do usuário atual do Azure AD para Power BI ou modelos de Azure Analysis Services ou SID para modelos do SQL Server Analysis Services.

USERPRINCIPALNAME

09/05/2020 • 2 minutes to read

Retorna o nome principal do usuário.

Sintaxe

```
USERPRINCIPALNAME()
```

Parâmetros

Essa expressão não tem parâmetros.

Valor retornado

A userprincipalname no momento da conexão.

Funções lógicas

08/05/2020 • 2 minutes to read

As funções lógicas agem sobre uma expressão para retornar informações sobre os valores ou conjuntos dela. Por exemplo, você pode usar a função IF para verificar o resultado de uma expressão e criar resultados condicionais.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
AND	Verifica se ambos os argumentos são TRUE e retorna TRUE se isso se confirmar.
COALESCE	Retorna a primeira expressão que não é avaliada como BLANK.
FALSE	Retorna o valor lógico FALSE.
IF	Verifica se uma condição fornecida como o primeiro argumento é atendida.
IF.EAGER	Usa execução adiantada para verificar se uma condição fornecida como o primeiro argumento é atendida.
IFERROR	Avalia uma expressão e retorna um valor especificado se a expressão retornar um erro
NOT	Altera FALSE para TRUE ou TRUE para FALSE.
OR	Verifica se um dos argumentos é TRUE para retornar TRUE.
SWITCH	Avalia uma expressão em relação a uma lista de valores e retorna uma das várias expressões de resultado possíveis.
TRUE	Retorna o valor lógico TRUE.

AND

08/05/2020 • 2 minutes to read

Verifica se ambos os argumentos são TRUE e retorna TRUE se isso se confirmar. Caso contrário, retorna false.

Sintaxe

```
AND(<logical1>,<logical2>)
```

Parâmetros

TERMO	DEFINIÇÃO
logical_1, logical_2	Os valores lógicos que você quer testar.

Valor retornado

Retorna true ou false dependendo da combinação de valores que você testa.

Comentários

A função **e** no DAX aceita apenas dois (2) argumentos. Se você precisar executar uma operação AND em várias expressões, poderá criar uma série de cálculos ou, melhor ainda, usar o operador AND (**&&**) para unir todas elas em uma expressão mais simples.

Exemplo 1

A fórmula a seguir mostra a sintaxe da função AND.

```
=IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

Como as duas condições, passadas como argumentos, para a função AND são true, a fórmula retorna "All True".

Exemplo 2

O exemplo a seguir usa a função AND com fórmulas aninhadas para comparar dois conjuntos de cálculos ao mesmo tempo. Para cada categoria de produto, a fórmula determina se as vendas do ano atual e as vendas do ano anterior do canal da Internet são maiores do que o canal do revendedor para os mesmos períodos. Se ambas as condições forem verdadeiras, para cada categoria, a fórmula retornará o valor "Sucesso na Internet".

```
= IF( AND( SUM( 'InternetSales_USD'[SalesAmount_USD])
           >SUM('ResellerSales_USD'[SalesAmount_USD])
           , CALCULATE(SUM('InternetSales_USD'[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey] ))
           >CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]), PREVIOUSYEAR('DateTime'[DateKey] ))
           )
  , "Internet Hit"
  , ""
  )
```

Retorna

FUNÇÃO AND	RÓTULOS DE COLUNA					
Rótulos de linha	2005	2006	2007	2008		Total Geral
Bretelles						
Racks de bicicleta						
Suportes de bicicleta				Sucesso na Internet		
Garrafas e compartimentos				Sucesso na Internet		
Suportes inferiores						
Freios						
Bonés						
Correntes						
Limpadores						
Pedaleiras						
Câmbios						
Para-choques				Sucesso na Internet		
Garfos						
Luvas						
Guidões						
Fones de ouvido						
Capacetes						
Conjuntos para hidratação						
Camisas						

FUNÇÃO AND	RÓTULOS DE COLUNA					
Luzes						
Bloqueios						
Mountain bikes						
Quadros para mountain bikes						
Cestos						
Pedais						
Bombas						
Bicicletas de estrada						
Quadros para bicicletas de estrada						
Selins						
Shorts						
Meias						
Calças						
Pneus e câmaras				Sucesso na Internet		
Bicicletas de passeio						
Quadros para bicicletas de passeio						
Coletes						
Rodas						
Total Geral						

Consulte também

COALESCE

26/05/2020 • 2 minutes to read

Retorna a primeira expressão que não é avaliada como BLANK. Se todas as expressões forem avaliadas como BLANK, haverá retorno de BLANK.

Sintaxe

```
COALESCE(<expression>, <expression>[, <expression>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorna um valor escalar.

Valor retornado

Um valor escalar proveniente de uma das expressões ou BLANK se todas as expressões forem avaliadas como BLANK.

Comentários

As expressões de entrada podem ser de tipos de dados diferentes.

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE { COALESCE(BLANK(), 10, DATE(2008, 3, 3)) }
```

Retorna **10**, que é a primeira expressão não avaliada como BLANK.

Exemplo 2

A seguinte expressão DAX:

```
= COALESCE(SUM(FactInternetSales[SalesAmount]), 0)
```

Retorna a soma de todos os valores da coluna SalesAmount da tabela FactInternetSales ou **0**. Isso pode ser usado para converter valores BLANK do total de vendas para **0**.

FALSO

09/05/2020 • 2 minutes to read

Retorna o valor lógico FALSE.

Sintaxe

```
FALSE()
```

Valor retornado

Sempre FALSE.

Comentários

A palavra FALSE também é interpretada como o valor lógico FALSE.

Exemplo

A fórmula retorna o valor lógico FALSE quando o valor na coluna, 'InternetSales_USD'[SalesAmount_USD], é inferior ou igual a 200000.

```
=IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

A tabela a seguir mostra os resultados quando a fórmula de exemplo é usada com 'ProductCategory'[ProductCategoryName] em rótulos de linha e 'DateTime'[CalendarYear] em rótulos de coluna.

VERDADEIRO-FALSO	RÓTULOS DE COLUNA					
Rótulos de linha	2005	2006	2007	2008		Total Geral
Acessórios	FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO
Bicicletas	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO
Vestuário	FALSO	FALSO	FALSO	FALSO	FALSO	VERDADEIRO
Componentes	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
Total Geral	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO

Consulte também

[Função TRUE \(DAX\)](#)

[Função NOT \(DAX\)](#)

[Função IF \(DAX\)](#)

[Referência de funções do DAX](#)

IF

08/05/2020 • 4 minutes to read

Verifica se uma condição fornecida como o primeiro argumento é atendida. Retornará um valor se uma condição for TRUE e retornará outro valor se a mesma condição for FALSE.

Sintaxe

```
IF(<logical_test>,<value_if_true>[, <value_if_false>])
```

Parâmetros

TERMO	DEFINIÇÃO
logical_test	Qualquer expressão ou valor que possa ser avaliado como TRUE ou FALSE.
value_if_true	O valor retornado se o teste lógico for TRUE.
value_if_false	O valor retornado se o teste lógico for FALSE. Se omitido, BLANK será retornado.

Valor retornado

Qualquer tipo de valor que possa ser retornado por uma expressão.

Comentários

Se o valor referenciado na expressão for uma coluna, IF retornará o valor correspondente à linha atual.

A função IF pode retornar o tipo de dados da variante se **value_if_true** e **value_if_false** forem de tipos de dados diferentes, mas a função tentará retornar um único tipo de dados se **value_if_true** e **value_if_false** forem de tipos de dados numéricos. No último caso, a função IF converterá implicitamente os tipos de dados para acomodar os dois valores. Por exemplo, a fórmula `IF(<condition>,TRUE(),0)` retorna TRUE ou 0, mas a fórmula `IF(<condition>,1.0,0)` retorna apenas valores decimais, embora **value_if_false** seja do tipo de dados número inteiro. Para obter mais informações sobre a conversão implícita de tipos de dados, confira [Tipos de dados](#).

Exemplo

O exemplo a seguir usa funções aninhadas IF que avaliam o número na coluna, Calls, da tabela FactCallCenter. A função atribui um rótulo da seguinte maneira: **low** (baixo) se o número de chamadas é menor que 200, **medium** (médio) se o número de chamadas é menor que 300, mas não menor que 200 e **high** (alto) para todos os outros valores.

```
=IF([Calls]<200,"low",IF([Calls]<300,"medium","high"))
```

Exemplo

O exemplo a seguir obtém uma lista de cidades que contêm clientes potenciais na área da Califórnia usando colunas da tabela ProspectiveBuyer. Já que a lista destina-se a planejar uma campanha que visará pessoas ou pessoas casadas com filhos em casa, a condição na função IF verifica o valor das colunas [MaritalStatus] e [NumberChildrenAtHome] e gera a cidade se uma das condições é atendida e se o cliente está na Califórnia. Caso contrário, ele gera a cadeia de caracteres vazia.

```
=IF([StateProvinceCode]= "CA" && ([MaritalStatus] = "M" || [NumberChildrenAtHome] >1),[City])
```

Observe que os parênteses são usados para controlar a ordem em que os operadores AND (&&) e OR (||) são usados. Observe também que nenhum valor foi especificado para **value_if_false**. Portanto, a função retorna o padrão, que é uma cadeia de caracteres vazia.

Consulte também

[Função TRUE \(DAX\)](#)

[Função FALSE \(DAX\)](#)

[Função NOT \(DAX\)](#)

[Função IF \(DAX\)](#)

[Referência de funções do DAX](#)

IF.EAGER

09/05/2020 • 2 minutes to read

Verifica se uma condição é atendida e retornará um valor, se for TRUE, e outro valor, se for FALSE. Usa execução adiantada.

Sintaxe

```
IF.EAGER ( <LogicalTest>, <ResultIfTrue> [, <ResultIfFalse>] )
```

Parâmetros

TERMO	DEFINIÇÃO
LogicalTest	Qualquer expressão ou valor que possa ser avaliado como TRUE ou FALSE.
ResultIfTrue	O valor retornado se o teste lógico for TRUE.
ResultIfFalse	(Opcional) O valor retornado se o teste lógico for FALSE; se omitido, BLANK será retornado.

Valor retornado

Qualquer tipo de valor que possa ser retornado por uma expressão.

IFERROR

09/05/2020 • 2 minutes to read

Avalia uma expressão e retorna um valor especificado se a expressão retorna um erro; caso contrário, retorna o valor da própria expressão.

Sintaxe

```
IFERROR(value, value_if_error)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	Qualquer valor ou expressão.
value_if_error	Qualquer valor ou expressão.

Valor retornado

Um escalar do mesmo tipo que **value**

Comentários

Você pode usar a função IFERROR para interceptar e manipular erros em uma expressão.

Se **value** ou **value_if_error** for uma célula vazia, IFERROR o tratará como um valor de cadeia de caracteres vazio ("").

A função IFERROR é baseada na função IF e usa as mesmas mensagens de erro, mas tem menos argumentos. A relação entre a função IFERROR e a função IF é a seguinte:

```
IFERROR(A,B) := IF(ISERROR(A), B, A)
```

Observe que os valores retornados para A e B devem ser do mesmo tipo de dados; portanto, a coluna ou expressão usada para **value** e o valor retornado para **value_if_error** devem ser do mesmo tipo de dados.

Exemplo

O exemplo a seguir retornará 9999 se a expressão 25/0 for avaliada como um erro. Se a expressão retornar um valor diferente de erro, esse valor será passado para a expressão de invocação.

```
=IFERROR(25/0,9999)
```

Consulte também

[Funções lógicas \(DAX\)](#)

IN

08/05/2020 • 2 minutes to read

Retornará True se o valor escalar aparecer em pelo menos uma linha da relação de entrada.

Sintaxe

IN

Parâmetros

TERMO	DEFINIÇÃO
expressão escalar	
expressão de tabela	

Valor retornado

True

Comentários

Nenhum

Exemplo

```
Filtered Sales:=CALCULATE (
    [Internet Total Sales], 'Product'[Color] IN { "Red", "Blue", "Black" }
)
```

Consulte também

NOT

08/05/2020 • 2 minutes to read

Altera FALSE para TRUE ou TRUE para FALSE.

Sintaxe

```
NOT(<logical>)
```

Parâmetros

TERMO	DEFINIÇÃO
logical	Uma expressão ou um valor que pode ser avaliado como TRUE ou FALSE.

Valor retornado

TRUE OU FALSE.

Exemplo

O exemplo a seguir recupera valores da coluna calculada que foi criada para ilustrar a função IF. Para esse exemplo, a coluna calculada foi nomeada usando o nome padrão, **Calculated Column1** e contém a seguinte fórmula: `=IF([Orders]<300,"true","false")`

A fórmula verifica o valor na coluna, [Orders], e retorna "true" se o número de pedidos é menor que 300.

Agora, crie uma nova coluna calculada, **Calculated Column2**, e digite a fórmula a seguir.

```
=NOT([CalculatedColumn1])
```

Para cada linha na **Calculated Column1**, os valores "true" e "false" são interpretados como os valores lógicos TRUE ou FALSE e a função NOT retorna o oposto lógico desse valor.

Consulte também

[Função TRUE \(DAX\)](#)

[Função FALSE \(DAX\)](#)

[Função IF \(DAX\)](#)

[Referência de funções do DAX](#)

OR

08/05/2020 • 3 minutes to read

Verifica se um dos argumentos é TRUE para retornar TRUE. A função retornará FALSE se os dois argumentos forem FALSE.

Sintaxe

```
OR(<logical1>,<logical2>)
```

Parâmetros

TERMO	DEFINIÇÃO
logical_1, logical_2	Os valores lógicos que você quer testar.

Valor retornado

Um valor booliano. O valor será TRUE se qualquer um dos dois argumentos for TRUE e será FALSE se os dois argumentos forem FALSE.

Comentários

A função **OR** no DAX aceita apenas dois (2) argumentos. Se precisar executar uma operação OR em várias expressões, você poderá criar uma série de cálculos ou, melhor ainda, usar o operador OR (||) para unir todas elas em uma expressão mais simples.

A função avalia os argumentos até o primeiro argumento TRUE e, em seguida, retorna TRUE.

Exemplo

O exemplo a seguir mostra como usar a função OR para obter os vendedores que pertencem ao Círculo de Excelência. O Círculo de Excelência reconhece as pessoas que alcançaram mais de um milhão de dólares em vendas de Bicicletas de Passeio ou vendas de mais de dois milhões e meio de dólares em 2007.

```
IF( OR( CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]),
'ProductSubcategory'[ProductSubcategoryName]="Touring Bikes") > 1000000
, CALCULATE(SUM('ResellerSales_USD'[SalesAmount_USD]), 'DateTime'[CalendarYear]=2007) > 2500000
)
, "Circle of Excellence"
, ""
)
```

Retorna

SALESPERSON FLAG	TRUE					

SALESPERSON FLAG	TRUE					
Função OR	Rótulos de Coluna					
Rótulos de linha	2005	2006	2007	2008		Total Geral
Abbas, Syed E						
Alberts, Amy E						
Ansman-Wolfe, Pamela O						
Blythe, Michael G	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência
Campbell, David R						
Carson, Jillian	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência
Ito, Shu K						
Jiang, Stephen Y						
Mensa-Annan, Tete A						
Mitchell, Linda C	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência
Pak, Jae B	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência
Reiter, Tsvi Michael						
Saraiva, José Edvaldo	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência
Tsoflias, Lynn N						
Valdez, Rachel B						
Vargas, Garrett R						

SALESPERSON FLAG	TRUE					
Varkey Chudukatil, Ranjit R						Círculo de Excelência
Total Geral	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência	Círculo de Excelência

Consulte também

[Funções lógicas \(DAX\)](#)

SWITCH

08/05/2020 • 2 minutes to read

Avalia uma expressão em relação a uma lista de valores e retorna uma das várias expressões de resultado possíveis.

Sintaxe

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).
valor	Um valor constante a ser correspondido com os resultados da <i>expressão</i> .
result	Qualquer expressão escalar a ser avaliada se os resultados da <i>expressão</i> corresponderem ao <i>valor</i> correspondente.
else	Qualquer expressão escalar a ser avaliada se os resultados da <i>expressão</i> não corresponderem a nenhum dos argumentos <i>value</i> .

Valor retornado

Um valor escalar proveniente de uma das expressões *result*, se houver uma correspondência com *value*, ou da expressão *else*, se não houver nenhuma correspondência com qualquer *value*.

Comentários

Todas as expressões de resultado e a expressão *else* devem ter o mesmo tipo de dados.

Exemplo

O exemplo a seguir cria uma coluna calculada de nomes de mês.

```
=SWITCH([Month], 1, "January", 2, "February", 3, "March", 4, "April",  
5, "May", 6, "June", 7, "July", 8, "August",  
9, "September", 10, "October", 11, "November", 12, "December",  
"Unknown month number" )
```

VERDADEIRO

09/05/2020 • 2 minutes to read

Retorna o valor lógico TRUE.

Sintaxe

```
TRUE()
```

Valor retornado

Sempre TRUE.

Comentários

A palavra TRUE também é interpretada como o valor lógico TRUE.

Exemplo

A fórmula retorna o valor lógico TRUE quando o valor da coluna 'InternetSales_USD'[SalesAmount_USD] é maior que 200.000.

```
= IF(SUM('InternetSales_USD'[SalesAmount_USD]) >200000, TRUE(), false())
```

A tabela a seguir mostra os resultados quando a fórmula de exemplo é usada em uma Tabela Dinâmica com 'ProductCategory'[ProductCategoryName] nos rótulos de linha e 'DateTime'[CalendarYear] nos rótulos de coluna.

VERDADEIRO- FALSO	RÓTULOS DE COLUNA					
Rótulos de linha	2005	2006	2007	2008		Total Geral
Acessórios	FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO
Bicicletas	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO
Vestuário	FALSO	FALSO	FALSO	FALSO	FALSO	VERDADEIRO
Componentes	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
	FALSO	FALSO	FALSO	FALSO	FALSO	FALSO
Total Geral	VERDADEIRO	VERDADEIRO	VERDADEIRO	VERDADEIRO	FALSO	VERDADEIRO

Consulte também

[Função FALSE \(DAX\)](#)

[Função NOT \(DAX\)](#)

[Função IF \(DAX\)](#)

[Referência de funções do DAX](#)

Funções matemáticas e trigonométricas

08/05/2020 • 5 minutes to read

As funções matemáticas e trigonométricas em DAX (Data Analysis Expressions) são muito semelhantes às funções matemáticas e trigonométricas do Excel. Esta seção lista as funções matemáticas fornecidas pelo DAX.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
ABS	Retorna o valor absoluto de um número.
ACOS	Retorna o arco cosseno ou cosseno inverso de um número.
ACOSH	Retorna o cosseno hiperbólico inverso de um número.
ACOT	Retorna o arco tangente ou cotangente inversa de um número.
ACOTH	Retorna a cotangente hiperbólica inversa de um número.
ASIN	Retorna o arco seno ou seno inverso de um número.
ASINH	Retorna o seno hiperbólico inverso de um número.
ATAN	Retorna o arco tangente ou a tangente inversa de um número.
ATANH	Retorna a tangente hiperbólica inversa de um número.
CEILING	Arredonda um número para cima, para o inteiro mais próximo ou para o múltiplo de significância mais próximo.
COMBIN	Retorna o número de combinações para um determinado número de itens.
COMBINA	Retorna o número de combinações (com repetições) para um determinado número de itens.
COS	Retorna o cosseno do ângulo especificado.
COSH	Retorna o cosseno hiperbólico de um número.
CURRENCY	Avalia o argumento e retorna o resultado como o tipo de dados de moeda.
DEGREES	Converte radianos em graus.
DIVIDE	Executa a divisão e retorna o resultado alternativo ou BLANK() na divisão por 0.

FUNÇÃO	DESCRIÇÃO
EVEN	Retorna o número arredondado para cima até o inteiro par mais próximo.
EXP	Retorna e elevado à potência de um determinado número.
FACT	Retorna o fatorial de um número, igual à série $1*2*3*...*$, terminando no número especificado.
FLOOR	Arredonda um número para baixo, em direção a zero, para o múltiplo de significância mais próximo.
GCD	Retorna o maior divisor comum de dois ou mais inteiros.
INT	Arredonda um número para baixo, para o inteiro mais próximo.
ISO.CEILING	Arredonda um número para cima, para o inteiro mais próximo ou para o múltiplo de significância mais próximo.
LCM	Retorna o mínimo múltiplo comum de inteiros.
LN	Retorna o logaritmo natural de um número.
LOG	Retorna o logaritmo de um número na base que você especificar.
LOG10	Retorna o logaritmo de base 10 de um número.
MROUND	Retorna um número arredondado para o múltiplo desejado.
ODD	Retorna o número arredondado para cima até o inteiro ímpar mais próximo.
PI	Retorna o valor de Pi, 3,14159265358979, com precisão de 15 dígitos.
POWER	Retorna o resultado de um número elevado a uma potência.
PRODUCT	Retorna o produto dos números em uma coluna.
PRODUCTX	Retorna o produto de uma expressão avaliada para cada linha de uma tabela.
QUOTIENT	Executa a divisão e retorna apenas a parte inteira do resultado da divisão.
RADIANS	Converte graus em radianos.
RAND	Retorna um número aleatório maior ou igual a 0 e menor que 1, distribuído uniformemente.

FUNÇÃO	DESCRIÇÃO
RANDBETWEEN	Retorna um número aleatório no intervalo entre dois números especificados por você.
ROUND	Arredonda um números conforme o número de dígitos especificado.
ROUNDDOWN	Arredonda um número para baixo, em direção a zero.
ROUNDUP	Arredonda um número para cima, afastando-o de 0 (zero).
SIGN	Determina o sinal de um número, o resultado de um cálculo ou um valor em uma coluna.
SQRT	Retorna a raiz quadrada de um número.
SUM	Adiciona todos os números de uma coluna.
SUMX	Retorna a soma de uma expressão avaliada para cada linha de uma tabela.
TRUNC	Trunca um número para um inteiro removendo a parte decimal ou fracionária do número.

ABS

08/05/2020 • 2 minutes to read

Retorna o valor absoluto de um número.

Sintaxe

```
ABS(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número para o qual você deseja o valor absoluto.

Valor retornado

Um número decimal.

Comentários

O valor absoluto de um número é um número decimal, inteiro ou decimal, sem seu sinal. Você pode usar a função ABS para garantir que apenas números não negativos sejam retornados de expressões quando aninhados em funções que exigem um número positivo.

Exemplo

O exemplo a seguir retorna o valor absoluto da diferença entre o preço de lista e o preço do revendedor, que você pode usar em uma nova coluna calculada, **DealerMarkup**.

```
=ABS([DealerPrice]-[ListPrice])
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função SIGN \(DAX\)](#)

ACOS

08/05/2020 • 2 minutes to read

Retorna o arco cosseno ou cosseno inverso de um número. O arco cosseno é o ângulo cujo cosseno é *número*. O ângulo retornado é fornecido em radianos no intervalo de 0 (zero) para pi.

Sintaxe

ACOS(number)

Parâmetros

TERMO	DEFINIÇÃO
Número	O cosseno do ângulo que você quer e deve ser entre -1 e 1.

Valor retornado

Retorna o arco cosseno ou cosseno inverso de um número.

Comentários

Se você quiser converter o resultado de radianos em graus, multiplique-o por 180/PI() ou use a função DEGREES.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ACOS(-0,5)	Arco cosseno de -0,5 em radianos, $2\pi/3$.	2,094395102
=ACOS(-0.5)*180/PI()	Arco cosseno de -0,5 em graus.	120

ACOSH

08/05/2020 • 2 minutes to read

Retorna o cosseno hiperbólico inverso de um número. O número deve ser maior ou igual a 1. O cosseno hiperbólico inverso é o valor cujo cosseno hiperbólico é *number*, portanto, ACOSH(COSH(number)) é igual a *number*.

Sintaxe

ACOSH(*number*)

Parâmetros

TERMO	DEFINIÇÃO
número	Qualquer número real igual ou maior que 1.

Valor retornado

Retorna o cosseno hiperbólico inverso de um número.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ACOSH(1)	Cosseno hiperbólico inverso de 1.	0
=ACOSH(10)	Cosseno hiperbólico inverso de 10.	2,993228

ACOT

09/05/2020 • 2 minutes to read

Retorna o principal valor do arco tangente ou a cotangente inversa de um número.

Sintaxe

```
ACOT(number)
```

Parâmetros

TERMO	DEFINIÇÃO
Número	O cosseno do ângulo que você deseja. Deve ser um número real.

Valor retornado

Um valor decimal único.

ACOT

09/05/2020 • 2 minutes to read

Retorna a cotangente hiperbólica inversa de um número.

Sintaxe

ACOTH(number)

Parâmetros

TERMO	DEFINIÇÃO
Número	O valor absoluto do Número deve ser maior que 1.

Valor retornado

Um valor decimal único.

ASIN

08/05/2020 • 2 minutes to read

Retorna o arco seno ou seno inverso de um número. O arco seno é o ângulo cujo seno é o *número*. O ângulo retornado é fornecido em radianos no intervalo de $-\pi/2$ a $\pi/2$.

Sintaxe

ASIN(number)

Parâmetros

TERMO	DEFINIÇÃO
número	O seno do ângulo que você quer e deve ser entre -1 e 1.

Valor retornado

Retorna o arco seno ou seno inverso de um número.

Comentários

Para expressar o arco seno em graus, multiplique o resultado por $180/\pi$ () ou use a função DEGREES.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ASIN(-0,5)	Arco seno de -0,5 em radianos, $-\pi/6$	-0,523598776
=ASIN(-0.5)*180/PI()	Arco seno de -0,5 em graus	-30
=DEGREES(ASIN(-0.5))	Arco seno de -0,5 em graus	-30

ASINH

09/05/2020 • 2 minutes to read

Retorna o seno hiperbólico inverso de um número. O seno hiperbólico inverso é o valor cujo seno hiperbólico é *number*, portanto, ASINH(SINH(number)) é igual a *number*.

Sintaxe

ASINH(number)

Parâmetros

TERMO	DEFINIÇÃO
número	Qualquer número real.

Valor retornado

Retorna o seno hiperbólico inverso de um número.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ASINH(-2.5)	Seno hiperbólico inverso de -2,5	-1,647231146
=ASINH(10)	Seno hiperbólico inverso de 10	2,99822295

ATAN

09/05/2020 • 2 minutes to read

Retorna o arco tangente ou a tangente inversa de um número. O arco tangente é o ângulo cuja tangente é *número*. O ângulo retornado é fornecido em radianos no intervalo de $-\pi/2$ a $\pi/2$.

Sintaxe

ATAN(*number*)

Parâmetros

TERMO	DEFINIÇÃO
número	A tangente do ângulo que você quer.

Valor retornado

Retorna a tangente hiperbólica inversa de um número.

Comentários

Para expressar o arco tangente em graus, multiplique o resultado por $180/\pi$ () ou use a função DEGREES.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ATAN(1)	Arco tangente de 1 em radianos, $\pi/4$	0,785398163
=ATAN(1)*180/PI()	Arco tangente de 1 em graus	45

ATANH

08/05/2020 • 2 minutes to read

Retorna a tangente hiperbólica inversa de um número. O número deve estar entre -1 e 1 (excluindo -1 e 1). A tangente hiperbólica inversa é o valor cuja tangente hiperbólica é *number*, portanto, ATANH(TANH(number)) é igual a *number*.

Sintaxe

ATANH(number)

Parâmetros

TERMO	DEFINIÇÃO
número	Qualquer número rela entre 1 e -1.

Valor retornado

Retorna a tangente hiperbólica inversa de um número.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ATANH(0.76159416)	Tangente hiperbólica inversa de 0,76159416	1,00000001
=ATANH(-0.1)		-0,100335348

Consulte também

[Função ATAN \(DAX\)](#)

CEILING

09/05/2020 • 3 minutes to read

Arredonda um número para cima, para o inteiro mais próximo ou para o múltiplo de significância mais próximo.

Sintaxe

```
CEILING(<number>, <significance>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja arredondar ou uma referência a uma coluna que contém números.
significância	O múltiplo de significância para o qual você deseja arredondar. Por exemplo, para arredondar para o inteiro mais próximo, digite 1.

Valor retornado

Um número arredondado conforme especificado.

Comentários

Há duas funções CEILING no DAX, com as seguintes diferenças:

- A função CEILING emula o comportamento da função CEILING no Excel.
- A função ISO.CEILING segue o comportamento definido por ISO para determinar o valor do teto.

As duas funções retornam o mesmo valor para números positivos, mas valores diferentes para números negativos. Ao usar um múltiplo de significância positivo, CEILING e ISO.CEILING arredondam números negativos para cima (em direção ao infinito positivo). Ao usar um múltiplo de significância negativo, CEILING arredonda os números negativos para baixo (em direção ao infinito negativo), enquanto ISO.CEILING arredonda números negativos para cima (em direção ao infinito positivo).

O tipo retornado geralmente é do mesmo tipo do argumento significativo, com as seguintes exceções:

- Se o tipo de argumento de número for moeda, o tipo retornado será moeda.
- Se o tipo de argumento de significância for booliano, o tipo retornado será inteiro.
- Se o tipo de argumento de significância for não numérico, o tipo retornado será real.

Exemplo

A fórmula a seguir retorna 4,45. Isso poderá ser útil se você quiser evitar o uso de unidades menores em seus preços. Se um produto existente tiver o preço de US\$ 4,42, você poderá usar CEILING para arredondar preços até a unidade mais próxima de cinco centavos.

```
=CEILING(4.42,0.05)
```

Exemplo

A fórmula a seguir retorna resultados semelhantes ao exemplo anterior, mas usa valores numéricos armazenados na coluna, **ProductPrice**.

```
=CEILING([ProductPrice],0.05)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função FLOOR \(DAX\)](#)

[Função ISO.CEILING \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

COMBIN

08/05/2020 • 2 minutes to read

Retorna o número de combinações para um determinado número de itens. Use COMBIN para determinar o número total possível de grupos para um determinado número de itens.

Sintaxe

COMBIN(number, number_chosen)

Parâmetros

TERMO	DEFINIÇÃO
número	O número de itens.
number_chosen	O número de itens em cada combinação.

Valor retornado

Retorna o número de combinações para um determinado número de itens.

Comentários

Os argumentos numéricos são truncados para inteiros.

Se um dos argumentos for não numérico, COMBIN retornará o valor de erro #VALUE!.

Se o número < 0, number_chosen < 0 ou número < number_chosen, COMBIN retornará o valor de erro #VALUE!.

Uma combinação é qualquer conjunto ou subconjunto de itens, independentemente de sua ordem interna. As combinações são diferentes de permutas, para as quais a ordem interna é significativa.

O número de combinações é o seguinte, em que number = n e number_chosen = k:

$$\binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

Em que

$$P_{k,n} = \frac{n!}{(n-k)!}$$

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=COMBIN(8,2)	Possíveis equipes de duas pessoas que podem ser formadas com oito candidatos.	28

COMBINA

09/05/2020 • 2 minutes to read

Retorna o número de combinações (com repetições) para um determinado número de itens.

Sintaxe

```
COMBINA(number, number_chosen)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Deve ser maior ou igual a 0 e maior ou igual a Number_chosen. Valores não inteiros são truncados.
number_chosen	Deve ser maior que ou igual a 0. Valores não inteiros são truncados.

Valor retornado

Retorna o número de combinações (com repetições) para um determinado número de itens.

Comentários

Se o valor de um dos argumentos estiver fora de suas restrições, COMBINA retornará o valor de erro #VALUE!.

Se um dos argumentos for um valor não numérico, COMBINA retornará o valor de erro #VALUE!.

A equação a seguir é usada, em que N é Número e M é Number_chosen:

$$\binom{N + M - 1}{N - 1}$$

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=COMBINA(4,3)	Retorna o número de combinações (com repetições) para 4 e 3.	20
=COMBINA(10,3)	Retorna o número de combinações (com repetições) para 10 e 3.	220

COS

08/05/2020 • 2 minutes to read

Retorna o cosseno do ângulo especificado.

Sintaxe

`COS(number)`

Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. O ângulo em radianos do qual você deseja obter o cosseno.

Valor retornado

Retorna o cosseno do ângulo especificado.

Comentários

Se o ângulo estiver em graus, multiplique o ângulo por $\text{PI()}/180$ ou use a função `RADIANS` para converter o ângulo em radianos.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
<code>=COS(1,047)</code>	Cosseno de 1,047 radianos	0,5001711
<code>=COS(60*PI()/180)</code>	Cosseno de 60 graus	0,5
<code>=COS(RADIANS(60))</code>	Cosseno de 60 graus	0,5

COSH

08/05/2020 • 2 minutes to read

Retorna o cosseno hiperbólico de um número.

Sintaxe

COSH(number)

Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. Qualquer número real do qual você deseja encontrar o cosseno hiperbólico.

Valor retornado

O cosseno hiperbólico de um número.

Comentários

A fórmula para o cosseno hiperbólico é:

$$\text{COSH}(z) = \frac{e^z + e^{-z}}{2}$$

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=COSH(4)	Cosseno hiperbólico igual a 4	27,308233
=COSH(EXP(1))	Cosseno hiperbólico da base do logaritmo natural.	7,6101251

CURRENCY

08/05/2020 • 2 minutes to read

Avalia o argumento e retorna o resultado como o tipo de dados de moeda.

Sintaxe

```
CURRENCY(<value>)
```

Parâmetros

TERMO	DEFINIÇÃO
valor	Qualquer expressão DAX que retorna um único valor escalar em que a expressão deve ser avaliada exatamente uma vez antes de todas as outras operações.

Valor retornado

O valor da expressão avaliada e retornada como um valor de tipo de moeda.

Comentários

- A função CURRENCY arredonda o 5º decimal significativo, em valor, para retornar o 4º dígito decimal; o arredondamento ocorrerá se o 5º decimal significativo for igual ou superior a 5. Por exemplo, se o valor for 3,66666666666666, a conversão em moeda retornará US\$ 3,6667; no entanto, se o valor for 3.0123456789, a conversão em moeda retornará US\$ 3,0123.
- Se o tipo de dados da expressão for TrueFalse, CURRENCY(<TrueFalse>) retornará US\$ 1 para valores True e US\$ 0 para valores False.
- Se o tipo de dados da expressão for Text, CURRENCY(<Text>) tentará converter o texto em um número; se a conversão for feita com êxito, o número será convertido em moeda, caso contrário, um erro será retornado.
- Se o tipo de dados da expressão for DateTime, CURRENCY(<DateTime>) converterá o valor de datetime em um número e esse número em moeda. Os valores de DateTime têm uma parte inteira que representa o número de dias entre a data especificada e 1900-03-01 e uma fração que representa a fração de um dia (em que 12 horas ou meio-dia é 0,5 dia). Se o valor da expressão não for um valor de DateTime adequado, um erro será retornado.

Exemplo

Converter o número 1.234,56 em um tipo de dados de moeda.

```
=CURRENCY(1234.56)
```

Retorna o valor US\$ 1.234,5600.

DEGREES

08/05/2020 • 2 minutes to read

Converte radianos em graus.

Sintaxe

```
DEGREES(angle)
```

Parâmetros

TERMO	DEFINIÇÃO
angle	Obrigatório. O ângulo em radianos que você quer converter.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=DEGREES(PI())	Graus de pi radianos	180

DIVIDE

08/05/2020 • 2 minutes to read

Executa a divisão e retorna o resultado alternativo ou BLANK() na divisão por 0.

Sintaxe

```
DIVIDE(<numerator>, <denominator> [,<alternateresult>])
```

Parâmetros

TERMO	DEFINIÇÃO
numerator	O dividendo ou o número a ser dividido.
denominator	O divisor ou o número pelo qual dividir.
alternateresult	(Opcional) O valor retornado quando a divisão por zero resulta em um erro. Quando não fornecido, o valor padrão é BLANK().

Valor retornado

Um número decimal.

Comentários

O resultado alternativo na divisão por 0 deve ser uma constante.

Exemplo

O exemplo a seguir retorna 2,5.

```
=DIVIDE(5,2)
```

Exemplo

O exemplo a seguir retorna BLANK.

```
=DIVIDE(5,0)
```

Exemplo

O exemplo a seguir retorna 1.

```
=DIVIDE(5,0,1)
```

Consulte também

[Função QUOTIENT \(DAX\)](#)

[Funções matemáticas e trigonométricas \(DAX\)](#)

Retorna o número arredondado para cima até o inteiro par mais próximo. Você pode usar essa função para processar itens que vêm em pares. Por exemplo, uma caixa de embalagem aceita linhas de um ou dois itens. A caixa está cheia quando o número de itens, arredondado para os dois mais próximos corresponde à capacidade da caixa.

Sintaxe

EVEN(number)

Parâmetros

TERMO	DEFINIÇÃO
número	O valor a ser arredondado.

Valor retornado

Retorna o número arredondado para cima até o inteiro par mais próximo.

Comentários

Se o número for não numérico, EVEN retornará o valor de erro #VALUE!.

Independentemente do sinal do número, um valor será arredondado para cima quando ajustado distanciando-se de zero. Se o número for um inteiro par, nenhum arredondamento ocorrerá.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=EVEN(1.5)	Arredonda 1,5 para o inteiro par mais próximo	2
=EVEN(3)	Arredonda 3 para o inteiro par mais próximo	4
=EVEN(2)	Arredonda 2 para o inteiro par mais próximo	2
=EVEN(-1)	Arredonda -1 para o inteiro par mais próximo	-2

EXP

09/05/2020 • 2 minutes to read

Retorna e elevado à potência de um determinado número. A constante e é igual a 2,71828182845904, a base do logaritmo natural.

Sintaxe

EXP(<number>)

Parâmetros

TERMO	DEFINIÇÃO
número	O expoente aplicado à base e. A constante e é igual a 2,71828182845904, a base do logaritmo natural.

Valor retornado

Um número decimal.

Exceções

Comentários

EXP é o inverso de LN, que é o logaritmo natural do número especificado.

Para calcular as potências de bases diferentes de e, use o operador de exponenciação (^). Para obter mais informações, confira [Referência do operador DAX](#).

Exemplo

A fórmula a seguir calcula e eleva a potência do número contido na coluna [Power] .

=EXP([Power])

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função LN \(DAX\)](#)

[Função EXP \(DAX\)](#)

[Função LOG \(DAX\)](#)

[Função LOG \(DAX\)](#)

FACT

09/05/2020 • 2 minutes to read

Retorna o fatorial de um número, igual à série $1*2*3*...*$, terminando no número especificado.

Sintaxe

FACT(<number>)

Parâmetros

TERMO	DEFINIÇÃO
número	O número não negativo para o qual você deseja calcular o fatorial.

Valor retornado

Um número decimal.

Comentários

Se o número não for um inteiro, ele será truncado e um erro será retornado. Se o resultado for muito grande, um erro será retornado.

Exemplo

A fórmula a seguir retorna o fatorial para a série de inteiros na coluna [Values] .

=FACT([Values])

A seguinte tabela mostra os resultados esperados:

VALORES	RESULTADOS
0	1
1	1
2	2
3	6
4	24
5	120
170	7.257415615308E+306

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função TRUNC \(DAX\)](#)

FLOOR

09/05/2020 • 2 minutes to read

Arredonda um número para baixo, em direção a zero, para o múltiplo de significância mais próximo.

Sintaxe

```
FLOOR(<number>, <significance>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O valor numérico que você quer arredondar.
significância	O múltiplo para o qual você quer arredondar. Os argumentos number e significance devem ser positivos ou negativos.

Valor retornado

Um número decimal.

Comentários

Se um dos argumentos for não numérico, FLOOR retornará o valor de erro **#VALUE!** .

Se número e significância tiverem sinais diferentes, FLOOR retornará o valor de erro **#NUM!** .

Independentemente do sinal do número, um valor é arredondado para baixo quando ajustado para fora de zero. Se o número for exatamente um múltiplo de significância, nenhum arredondamento ocorrerá.

Exemplo

A fórmula a seguir usa os valores na coluna [Custo Total do Produto] da tabela, InternetSales e arredonda para baixo até o múltiplo mais próximo de 0,1.

```
=FLOOR(InternetSales[Total Product Cost],.5)
```

A seguinte tabela mostra os resultados esperados para alguns valores de exemplo.

VALORES	RESULTADO ESPERADO
10,8423	10,8
8,0373	8
2,9733	2.9

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

GCD

09/05/2020 • 2 minutes to read

Retorna o maior divisor comum de dois ou mais inteiros. Ele é o maior inteiro que divide o número1 e o número2 sem resto.

Sintaxe

```
GCD(number1, [number2], ...)
```

Parâmetros

TERMO	DEFINIÇÃO
número1, número2, ...	Número1 é necessário, os números seguintes são opcionais. 1 a 255 valores. Se algum valor não for um inteiro, ele estará truncado.

Valor retornado

O maior divisor comum de dois ou mais inteiros.

Comentários

Se algum argumento for não numérico, GCD retornará o valor de erro #VALUE!.

Se algum argumento for menor que zero, GCD retornará o valor de erro #VALUE!.

Um divide qualquer valor uniformemente.

Um número primo tem apenas a si e um como divisores pares.

Se um parâmetro para GCD for $\geq 2^{53}$, GCD retornará o valor de erro #VALUE!.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= GCD (5, 2)	Maior divisor comum de 5 e 2.	1
= GCD (24, 36)	Maior divisor comum de 24 e 36.	12
= GCD (7, 1)	Maior divisor comum de 7 e 1.	1

INT

09/05/2020 • 2 minutes to read

Arredonda um número para baixo, para o inteiro mais próximo.

Sintaxe

```
INT(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja arredondar para baixo, para um inteiro

Valor retornado

Um número inteiro.

Comentários

TRUNC e INT são semelhantes, pois ambas retornam inteiros. TRUNC remove a parte fracionária do número. INT arredonda os números para baixo até o inteiro mais próximo com base no valor da parte fracionária do número. INT e TRUNC são diferentes somente ao usar números negativos: `TRUNC(-4.3)` retorna -4, mas `INT(-4.3)` retorna -5 porque -5 é o número mais baixo.

Exemplo

A seguinte expressão arredonda o valor para 1. Se você usar a função ROUND, o resultado será 2.

```
=INT(1.5)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

[Função MROUND \(DAX\)](#)

ISO.CEILING

08/05/2020 • 3 minutes to read

Arredonda um número para cima, para o inteiro mais próximo ou para o múltiplo de significância mais próximo.

Sintaxe

```
ISO.CEILING(<number>[, <significance>])
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja arredondar ou uma referência a uma coluna que contém números.
significância	(opcional) O múltiplo de significância para o qual você deseja arredondar. Por exemplo, para arredondar para o inteiro mais próximo, digite 1. Se a unidade de significância não for especificada, o número será arredondado para cima até o inteiro mais próximo.

Valor retornado

Um número, do mesmo tipo que o argumento *number*, arredondado conforme especificado.

Comentários

Há duas funções CEILING no DAX, com as seguintes diferenças:

- A função CEILING emula o comportamento da função CEILING no Excel.
- A função ISO.CEILING segue o comportamento definido por ISO para determinar o valor do teto.

As duas funções retornam o mesmo valor para números positivos, mas valores diferentes para números negativos. Ao usar um múltiplo de significância positivo, CEILING e ISO.CEILING arredondam números negativos para cima (em direção ao infinito positivo). Ao usar um múltiplo de significância negativo, CEILING arredonda os números negativos para baixo (em direção ao infinito negativo), enquanto ISO.CEILING arredonda números negativos para cima (em direção ao infinito positivo).

O tipo de resultado geralmente é o mesmo tipo de significância usado como argumento com as seguintes exceções:

- Se o primeiro argumento for do tipo de moeda, o resultado será o tipo de moeda.
- Se o argumento opcional não for incluído, o resultado será do tipo inteiro.
- Se o argumento de significância for do tipo booliano, o resultado será do tipo inteiro.
- Se o argumento de significância for do tipo não numérico, o resultado será do tipo real.

Exemplo: Números positivos

Descrição

A fórmula a seguir retorna 4,45. Isso poderá ser útil se você quiser evitar o uso de unidades menores em seus preços. Se um produto existente tiver o preço de US\$ 4,42, você poderá usar ISO.CEILING para arredondar preços até a unidade mais próxima de cinco centavos.

Código

```
=ISO.CEILING(4.42,0.05)
```

Exemplo: Números negativos

Descrição

A seguinte fórmula retorna o valor de teto ISO de -4,40.

Código

```
=ISO.CEILING(-4.42,0.05)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função FLOOR \(DAX\)](#)

[Função CEILING \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

LCM

08/05/2020 • 2 minutes to read

Retorna o mínimo múltiplo comum de inteiros. O mínimo múltiplo comum é o menor inteiro positivo que é múltiplo de todos os argumentos inteiros número1, número2 e assim por diante. Use LCM para adicionar frações com diferentes denominadores.

Sintaxe

```
LCM(number1, [number2], ...)
```

Parâmetros

TERMO	DEFINIÇÃO
número1, número2,...	Número1 é necessário, os números seguintes são opcionais. 1 a 255 valores cujo mínimo múltiplo comum você deseja encontrar. Se o valor não for um inteiro, ele será truncado.

Valor retornado

Retorna o mínimo múltiplo comum de inteiros.

Comentários

Se algum argumento for não numérico, LCM retornará o valor de erro #VALUE!.

Se algum argumento for menor que zero, LCM retornará o valor de erro #VALUE!.

Se $\text{LCM}(a,b) > 2^{53}$, LCM retornará o valor de erro #VALUE!.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=LCM(5, 2)	O mínimo múltiplo comum de 5 e 2.	10
=LCM(24, 36)	O mínimo múltiplo comum de 24 e 36.	72

LN

09/05/2020 • 2 minutes to read

Retorna o logaritmo natural de um número. Os logaritmos naturais se baseiam na constante e (2,71828182845904).

Sintaxe

```
LN(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número positivo cujo logaritmo natural você deseja obter.

Valor retornado

Um número decimal.

Comentários

A função LN é o inverso da função EXP.

Exemplo

O exemplo a seguir retorna o logaritmo natural do número da coluna `[Values]`.

```
=LN([Values])
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função EXP \(DAX\)](#)

LOG

09/05/2020 • 2 minutes to read

Retorna o logaritmo de um número na base que você especificar.

Sintaxe

LOG(<number>,<base>)

Parâmetros

TERMO	DEFINIÇÃO
número	O número positivo cujo logaritmo você deseja obter.
base	A base do logaritmo. Se omitida, a base será 10.

Valor retornado

Um número decimal.

Comentários

Você poderá ver um erro se o valor for grande demais para ser exibido.

A função LOG10 é semelhante, mas sempre retorna o logaritmo comum, ou seja, o logaritmo na base 10.

Exemplo

As fórmulas a seguir retornam o mesmo resultado, 2.

```
=LOG(100,10)  
=LOG(100)  
=LOG10(100)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função EXP \(DAX\)](#)

[Função LOG \(DAX\)](#)

[Função LOG \(DAX\)](#)

LOG10

09/05/2020 • 2 minutes to read

Retorna o logaritmo de base 10 de um número.

Sintaxe

```
LOG10(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número positivo cujo logaritmo de base 10 você deseja obter.

Valor retornado

Um número decimal.

Comentários

A função LOG permite alterar a base do logaritmo, em vez de usar a base 10.

Exemplo

As fórmulas a seguir retornam o mesmo resultado: 2.

```
=LOG(100,10)  
=LOG(100)  
=LOG10(100)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função EXP \(DAX\)](#)

[Função LOG \(DAX\)](#)

[Função LOG \(DAX\)](#)

MOD

09/05/2020 • 2 minutes to read

Retorna o resto após a divisão de um número por um divisor. O resultado sempre tem o mesmo sinal do divisor.

Sintaxe

```
MOD(<number>, <divisor>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número para o qual você deseja encontrar o resto, após a execução da divisão.
divisor	O número pelo qual você deseja dividir.

Valor retornado

Um número inteiro.

Comentários

Se o divisor for 0 (zero), o MOD retornará um erro. Não é possível dividir por 0.

A função MOD pode ser expressa em termos da função INT: $MOD(n, d) = n - d * INT(n/d)$

Exemplo

A fórmula a seguir retorna 1, o resto de 3 dividido por 2.

```
=MOD(3,2)
```

Exemplo

A fórmula a seguir retorna -1, o resto de 3 dividido por 2. Observe que o sinal é sempre o mesmo que o sinal do divisor.

```
=MOD(-3,-2)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

Função MROUND (DAX)

Função INT (DAX)

MROUND

09/05/2020 • 2 minutes to read

Retorna um número arredondado para o múltiplo desejado.

Sintaxe

```
MROUND(<number>, <multiple>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Número a ser arredondado.
multiple	O múltiplo de significância para o qual você deseja arredondar o número.

Valor retornado

Um número decimal.

Comentários

MROUND arredondará para cima, distanciando de zero, se o restante da divisão de **number** pelo **multiple** especificado for maior ou igual à metade do valor de **multiple**.

Exemplo: Casas Decimais

Descrição

A expressão a seguir arredonda 1,3 para o múltiplo mais próximo de 0,2. O resultado esperado é 1,4.

Código

```
=MROUND(1.3,0.2)
```

Exemplo: Números negativos

Descrição

A expressão a seguir arredonda -10 para o múltiplo mais próximo de -3. O resultado esperado é -9.

Código

```
=MROUND(-10,-3)
```

Exemplo: Erro

Descrição

A expressão a seguir retorna um erro, pois os números têm sinais diferentes.

Código

```
=MROUND(5, -2)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

[Função MROUND \(DAX\)](#)

[Função INT \(DAX\)](#)

ODD

09/05/2020 • 2 minutes to read

Retorna o número arredondado para cima até o inteiro ímpar mais próximo.

Sintaxe

ODD(number)

Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. O valor a ser arredondado.

Valor retornado

Retorna o número arredondado para cima até o inteiro ímpar mais próximo.

Comentários

Se o número for não numérico, ODD retornará o valor de erro #VALUE!.

Independentemente do sinal do número, um valor será arredondado para cima quando ajustado distanciando-se de zero. Se o número for um inteiro ímpar, nenhum arredondamento ocorrerá.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=ODD(1.5)	Arredonda 1,5 para cima até o inteiro ímpar mais próximo.	3
=ODD(3)	Arredonda 3 para cima até o inteiro ímpar mais próximo.	3
=ODD(2)	Arredonda 2 para cima até o inteiro ímpar mais próximo.	3
=ODD(-1)	Arredonda -1 para cima até o inteiro ímpar mais próximo.	-1
=ODD(-2)	Arredonda -2 para cima (distanciando-se de 0) até o inteiro ímpar mais próximo.	-3

PI

09/05/2020 • 2 minutes to read

Retorna o valor de Pi, 3,14159265358979, com precisão de 15 dígitos.

Sintaxe

```
PI()
```

Valor retornado

Um número decimal com o valor de Pi, 3,14159265358979, com precisão de 15 dígitos.

Comentários

Pi é uma constante matemática. No DAX, o Pi é representado como um número real com precisão de 15 dígitos, a mesmo que a do Excel.

Exemplo

A fórmula a seguir calcula a área de um círculo de acordo com o raio na coluna, `[Radius]`.

```
=PI()*([Radius]*2)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

POWER

09/05/2020 • 2 minutes to read

Retorna o resultado de um número elevado a uma potência.

Sintaxe

```
POWER(<number>, <power>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número base, que pode ser qualquer número real.
potência	O expoente ao qual a base é elevada.

Valor retornado

Um número decimal.

Exemplo

O exemplo a seguir retorna 25.

```
=POWER(5,2)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

PRODUTO

08/05/2020 • 2 minutes to read

Retorna o produto dos números em uma coluna.

Para retornar o produto de uma expressão avaliada para cada linha de uma tabela, use a [função PRODUCTX \(DAX\)](#).

Sintaxe

```
PRODUCT(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números para os quais o produto deve ser calculado.

Valor retornado

Um número decimal.

Comentários

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

PRODUCT(Table[Column]) é equivalente a PRODUCTX(Table, Table[Column])

Exemplo

O seguinte calcula o produto da coluna AdjustedRates em uma tabela de Anuidade:

```
=PRODUCT( Annuity[AdjustedRates] )
```

Consulte também

[Função PRODUCTX \(DAX\)](#)

PRODUCTX

09/05/2020 • 2 minutes to read

Retorna o produto de uma expressão avaliada para cada linha de uma tabela.

Para retornar o produto dos números em uma coluna, use [função PRODUCT \(DAX\)](#).

Sintaxe

```
PRODUCTX(<table>, <expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

Valor retornado

Um número decimal.

Comentários

A função PRODUCTX leva como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento é uma coluna que contém os números para os quais você deseja calcular o produto ou uma expressão avaliada como uma coluna.

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

Exemplo

O exemplo a seguir computa o valor futuro de um investimento:

```
= [PresentValue] * PRODUCTX( AnnuityPeriods, 1+[FixedInterestRate] )
```

Consulte também

[Função PRODUCT \(DAX\)](#)

QUOTIENT

08/05/2020 • 2 minutes to read

Executa a divisão e retorna apenas a parte inteira do resultado da divisão. Use essa função quando desejar descartar o restante da divisão.

Sintaxe

```
QUOTIENT(<numerator>, <denominator>)
```

Parâmetros

TERMO	DEFINIÇÃO
numerator	O dividendo ou o número a ser dividido.
denominator	O divisor ou o número pelo qual dividir.

Valor retornado

Um número inteiro.

Comentários

Se um dos argumentos for não numérico, QUOTIENT retornará o valor de erro **#VALUE!** .

Você pode usar uma referência de coluna em vez de um valor literal para qualquer um dos argumentos. No entanto, se a coluna que você referenciar contiver um 0 (zero), um erro será retornado para a coluna inteira de valores.

Exemplo

As fórmulas a seguir retornam o mesmo resultado, 2.

```
=QUOTIENT(5,2)  
=QUOTIENT(10/2,2)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

RADIANS

09/05/2020 • 2 minutes to read

Converte graus em radianos.

Sintaxe

RADIANS(*angle*)

Parâmetros

TERMO	DEFINIÇÃO
angle	Obrigatório. Um ângulo, em graus, que você quer converter.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=RADIANS(270)	270 graus como radianos (4,712389 ou $3\pi/2$ radianos)	4,712389

RAND

08/05/2020 • 2 minutes to read

Retorna um número aleatório maior ou igual a 0 e menor que 1, distribuído uniformemente. O número retornado é alterado cada vez que a célula que contém essa função é recalculada.

Sintaxe

```
RAND()
```

Valor retornado

Um número decimal.

Comentários

O recálculo depende de vários fatores, incluindo se a pasta de trabalho está configurada com o modo de recálculo **Manual** ou **Automático** e se os dados foram atualizados. Isso é diferente do Microsoft Excel, em que você pode controlar quando RAND gera um novo número aleatório desativando o recálculo.

RAND e outras funções voláteis que não têm valores fixos nem sempre são recalculadas. Por exemplo, a execução de uma consulta ou filtragem geralmente não fará com que essas funções sejam reavaliadas. No entanto, os resultados dessas funções serão recalculados quando a coluna inteira for recalculada. Essas situações incluem a atualização de uma fonte de dados externa ou a edição manual de dados que leva à reavaliação das fórmulas que contêm essas funções.

Além disso, RAND sempre será recalculado se a função for usada na definição de uma medida.

Além disso, nesses contextos, a função RAND não pode retornar um resultado igual a zero, para evitar erros como divisão por zero.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

Para gerar um número real aleatório entre dois outros números, você pode usar uma fórmula como a seguinte:

```
= RAND()*(int1-int2)+int1
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

RANDBETWEEN

09/05/2020 • 2 minutes to read

Retorna um número aleatório no intervalo entre dois números especificados por você.

Sintaxe

```
RANDBETWEEN(<bottom>,<top>)
```

Parâmetros

TERMO	DEFINIÇÃO
Inferior	O menor número inteiro que a função retornará.
Superior	O maior número inteiro que a função retornará.

Valor retornado

Um número inteiro.

Comentários

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

A fórmula a seguir retorna um número aleatório entre 1 e 10.

```
=RANDBETWEEN(1,10)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

ROUND

09/05/2020 • 2 minutes to read

Arredonda um números conforme o número de dígitos especificado.

Sintaxe

```
ROUND(<number>, <num_digits>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja arredondar.
num_digits	O número de dígitos para o qual você deseja arredondar. Um valor negativo arredonda os dígitos à esquerda do ponto decimal, ao passo que o valor igual a zero arredonda para o número inteiro mais próximo.

Valor retornado

Um número decimal.

Comentários

Se **num_digits** for maior que 0 (zero), o número será arredondado para o número especificado de casas decimais.

Se **num_digits** for 0, o número será arredondado para o número inteiro mais próximo.

Se **num_digits** for menor que 0, o número será arredondado à esquerda do ponto decimal.

Funções relacionadas

Para sempre arredondar para cima (distanciando-se do zero), use a função ROUNDUP.

Para sempre arredondar para baixo (aproximando-se de zero), use a função ROUNDDOWN.

Para arredondar um número para um múltiplo específico (por exemplo, para arredondar para o múltiplo mais próximo de 0,5), use a função MROUND.

Você pode usar as funções TRUNC e INT para obter a parte inteira do número.

Exemplo

A fórmula a seguir arredonda 2,15 para cima, para uma casa decimal. O resultado esperado é 2,2.

```
=ROUND(2.15,1)
```

Exemplo

A fórmula a seguir arredonda 21,5 para uma casa decimal à esquerda do ponto decimal. O resultado esperado é 20.

```
=ROUND(21.5, -1)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

[Função MROUND \(DAX\)](#)

[Função INT \(DAX\)](#)

[Função TRUNC \(DAX\)](#)

ROUNDDOWN

08/05/2020 • 2 minutes to read

Arredonda um número para baixo, em direção a zero.

Sintaxe

```
ROUNDDOWN(<number>, <num_digits>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Um número real que você deseja arredondar para baixo.
num_digits	O número de dígitos para o qual você deseja arredondar. Arredondamentos negativos para a esquerda do ponto decimal; zero para o número inteiro mais próximo.

Valor retornado

Um número decimal.

Comentários

Se **num_digits** for maior que 0 (zero), o valor em **number** será arredondado para baixo o número especificado de casas decimais.

Se **num_digits** for 0, o valor em **number** será arredondado para baixo para o número inteiro mais próximo.

Se **num_digits** for menor que 0, o valor em **number** será arredondado para baixo para a esquerda do ponto decimal.

Funções relacionadas

ROUNDDOWN se comporta como ROUND, exceto pelo fato de que sempre arredonda o número para baixo. A função INT também arredonda para baixo, mas com INT o resultado sempre é um inteiro, enquanto com ROUNDDOWN você pode controlar a precisão do resultado.

Exemplo

O exemplo a seguir arredonda 3,14159 para baixo até três casas decimais. O resultado esperado é 3,141.

```
=ROUNDDOWN(3.14159,3)
```

Exemplo

O exemplo a seguir arredonda o valor de 31415,92654 para baixo até 2 casas decimais à esquerda do decimal. O resultado esperado é 31400.

=ROUNDDOWN(31415.92654, -2)

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

[Função MROUND \(DAX\)](#)

[Função INT \(DAX\)](#)

ROUNDUP

09/05/2020 • 2 minutes to read

Arredonda um número para cima, afastando-o de 0 (zero).

Sintaxe

```
ROUNDUP(<number>, <num_digits>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Um número real que você deseja arredondar para cima.
num_digits	O número de dígitos para o qual você deseja arredondar. Um valor negativo para num_digits arredonda para a esquerda do ponto decimal; se num_digits for zero ou for omitido, número será arredondado para o número inteiro mais próximo.

Valor retornado

Um número decimal.

Comentários

ROUNDUP se comporta como ROUND, exceto pelo fato de que sempre arredonda um número para cima.

- Se **num_digits** for maior que 0 (zero), o número será arredondado para cima até o número especificado de casas decimais.
- Se **num_digits** for 0, o número será arredondado para o número inteiro mais próximo.
- Se **num_digits** for menor que 0, o número será arredondado para cima à esquerda do ponto decimal.

Funções relacionadas

ROUNDUP se comporta como ROUND, exceto pelo fato de que sempre arredonda um número para cima.

Exemplo

A fórmula a seguir arredonda PI para quatro casas decimais. O resultado esperado é 3,1416.

```
=ROUNDUP(PI(),4)
```

Exemplo: decimais como segundo argumento

Descrição

A fórmula a seguir Arredonda 1,3 para o múltiplo mais próximo de 0,2. O resultado esperado é 2.

Código

```
=ROUNDUP(1.3,0.2)
```

Exemplo: Número negativo como segundo argumento

Descrição

A fórmula a seguir Arredonda o valor na coluna, **FreightCost**, com os resultados esperados mostrados na tabela a seguir:

Código

```
=ROUNDUP([Values],-1)
```

Comentários

Quando **num_digits** é menor que zero, o número de casas à esquerda do sinal decimal é aumentado pelo valor especificado por você.

FREIGHTCOST	RESULTADO ESPERADO
13.25	20
2.45	10
25.56	30
1.34	10
345.01	350

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDDOWN \(DAX\)](#)

[Função MROUND \(DAX\)](#)

[Função INT \(DAX\)](#)

SIGN

09/05/2020 • 2 minutes to read

Determina o sinal de um número, o resultado de um cálculo ou um valor em uma coluna. A função retornará um (1) se o número for positivo, zero (0) se o número for zero ou menos um (-1) se o número for negativo.

Sintaxe

```
SIGN(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	Qualquer número real, uma coluna que contenha números ou uma expressão que seja avaliada como um número.

Valor retornado

Um número inteiro. Os valores de retorno possíveis são 1, 0 e -1.

VALOR RETORNADO	DESCRIÇÃO
1	O número é positivo
0	O número é zero
-1	O número é negativo

Exemplo

A fórmula a seguir retorna o sinal do resultado da expressão que calcula o preço de venda menos o custo.

```
=SIGN( ([Sale Price] - [Cost]) )
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

SQRT

09/05/2020 • 2 minutes to read

Retorna a raiz quadrada de um número.

Sintaxe

```
SQRT(<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número para o qual você deseja a raiz quadrada, uma coluna que contém números ou uma expressão que é avaliada como um número.

Valor retornado

Um número decimal.

Comentários

Se o número for negativo, a função SQRT retornará um erro.

Exemplo

A fórmula a seguir retorna 5.

```
=SQRT(25)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

SUM

09/05/2020 • 2 minutes to read

Adiciona todos os números de uma coluna.

Sintaxe

```
SUM(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números a serem somados.

Valor retornado

Um número decimal.

Comentários

Se você quiser filtrar os valores que está somando, poderá usar a função SUMX e especificar uma expressão para efetuar a soma.

Exemplo

O exemplo a seguir soma todos os números contidos na coluna AMT da tabela Sales.

```
=SUM(Sales[Amt])
```

Consulte também

[Função SUMX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

SUMX

09/05/2020 • 2 minutes to read

Retorna a soma de uma expressão avaliada para cada linha de uma tabela.

Sintaxe

```
SUMX(<table>, <expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

Valor retornado

Um número decimal.

Comentários

A função SUMX leva como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento é uma coluna que contém os números que você deseja somar ou uma expressão avaliada como uma coluna.

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

Para ver alguns exemplos mais complexos de SUMX em fórmulas, confira [Função ALL \(DAX\)](#) e [Função CALCULATE \(DAX\)](#).

Exemplo

O exemplo a seguir primeiro filtra a tabela, InternetSales, na expressão, `ShippingTerritoryID = 5`. Em seguida, retorna a soma de todos os valores na coluna, Freight. Em outras palavras, a expressão retorna a soma dos encargos de frete apenas para a área de vendas especificada.

```
=SUMX(FILTER(InternetSales, InternetSales[SalesTerritoryID]=5),[Freight])
```

Se você não precisar filtrar a coluna, use a função SUM. A função SUM é semelhante à função do Excel de mesmo nome, exceto pelo fato de que usa uma coluna como referência.

Consulte também

[Função SUM \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

TRUNC

09/05/2020 • 2 minutes to read

Trunca um número para um inteiro removendo a parte decimal ou fracionária do número.

Sintaxe

```
TRUNC(<number>,<num_digits>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja truncar.
num_digits	Um número que especifica a precisão do truncamento; se omitido, assumirá 0 (zero)

Valor retornado

Um número inteiro.

Comentários

TRUNC e INT são semelhantes, pois ambas retornam inteiros. TRUNC remove a parte fracionária do número. INT arredonda os números para baixo até o inteiro mais próximo com base no valor da parte fracionária do número. INT e TRUNC são diferentes apenas ao usar números negativos: `TRUNC(-4.3)` retorna -4, mas `INT(-4.3)` retorna -5, pois -5 é o número menor.

Exemplo

A fórmula a seguir retorna 3, a parte inteira de PI.

```
=TRUNC(PI())
```

Exemplo

A fórmula a seguir retorna -8, a parte inteira de -8,9.

```
=TRUNC(-8.9)
```

Consulte também

[Funções matemáticas e trigonométricas \(DAX\)](#)

[Função ROUND \(DAX\)](#)

[Função ROUNDUP \(DAX\)](#)

[Função ROUNDOWN \(DAX\)](#)

Função MROUND (DAX)

Função INT (DAX)

Outras funções

08/05/2020 • 2 minutes to read

Essas funções executam ações exclusivas que não podem ser definidas por nenhuma das categorias.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
CONVERT	Converte uma expressão de um tipo de dados em outro.
DATATABLE	Fornecer um mecanismo para declarar um conjunto embutido de valores de dados.
ERROR	Gera um erro com uma mensagem de erro.
EXCEPT	Retorna as linhas de uma tabela que não aparecem em outra tabela.
GENERATESERIES	Retorna uma tabela de coluna única que contém os valores de uma série aritmética.
GROUPBY	Semelhante à função SUMMARIZE, GROUPBY não faz um uso explícito de CALCULATE em nenhuma coluna de extensão que adiciona.
INTERSECT	Retorna a interseção de linha entre duas tabelas, retendo as duplicatas.
ISEMPTY	Verifica se uma tabela está vazia.
ISSELECTEDMEASURE	Usada por expressões para itens de cálculo a fim de determinar se a medida que está no contexto é uma das especificadas em uma lista de medidas.
NATURALINNERJOIN	Executa uma junção interna de uma tabela com outra tabela.
NATURALLEFTOUTERJOIN	Executa uma junção interna de uma tabela com outra tabela.
SELECTEDMEASURE	Usada por expressões de itens de cálculo para referenciar a medida que está no contexto.
SELECTEDMEASUREFORMATSTRING	Usado por expressões para itens de cálculo para recuperar a cadeia de caracteres de formato da medida que está no contexto.
SELECTEDMEASURENAME	Usada por expressões de itens de cálculo para determinar a medida que está no contexto por nome.
SUMMARIZECOLUMNS	Retorna uma tabela de resumo por um conjunto de grupos.

FUNÇÃO	DESCRIÇÃO
Construtor de tabela (DAX)	Retorna uma tabela de uma ou mais colunas.
TREATAS	Aplica o resultado de uma expressão de tabela como filtros a colunas de uma tabela não relacionada.
UNION	Cria uma tabela de união (junção) de um par de tabelas.

CONVERT

08/05/2020 • 2 minutes to read

Converte uma expressão de um tipo de dados em outro.

Sintaxe

```
CONVERT(<Expression>, <Datatype>)
```

Parâmetros

TERMO	DEFINIÇÃO
Expressão	Qualquer expressão válida.
Datatype	Uma enumeração que inclui: INTEGER(Whole Number), DOUBLE(Decimal Number), STRING(Text), BOOLEAN(True/False), CURRENCY(Fixed Decimal Number), DATETIME(Date, Time etc).

Valor retornado

Retorna o valor de , convertido em .

Comentários

A função retorna um erro quando um valor não pode ser convertido no tipo de dados especificado.

As colunas calculadas DAX precisam ser de um único tipo de dados. Como as funções MEDIAN e MEDIANX em uma coluna de inteiros retornam tipos de dados mistos, integer ou double, a seguinte expressão de coluna calculada retornará um erro como resultado: `MedianNumberCarsOwned = MEDIAN(DimCustomer[NumberCarsOwned])`

Para evitar tipos de dados mistos, altere a expressão para sempre retornar o tipo de dados double, por exemplo:

```
MedianNumberCarsOwned = MEDIANX(DimCustomer, CONVERT([NumberCarsOwned], DOUBLE))
```

Exemplo

Consulta DAX

```
EVALUATE { CONVERT(DATE(1900, 1, 1), INTEGER) }
```

Retorna

[VALOR]
2

DATATABLE

08/05/2020 • 2 minutes to read

Fornece um mecanismo para declarar um conjunto embutido de valores de dados.

Sintaxe

```
DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}}...)
```

Parâmetros

TERMO	DEFINIÇÃO
ColumnName	Qualquer expressão DAX que retorna uma tabela.
DataType	Uma enumeração que inclui: INTEGER, DOUBLE, STRING, BOOLEAN, CURRENCY, DATETIME
Valor	<p>Um único argumento usando a sintaxe do Excel para uma constante de matriz dimensional, aninhada para fornecer uma matriz de matrizes. Esse argumento representa o conjunto de valores de dados que estarão na tabela</p> <p>Por exemplo, { {valores em row1}, {valores em row2}, {valores em row3} etc. } Em que {valores em row1} é um conjunto delimitado por vírgula de expressões constantes, ou seja, uma combinação de constantes, combinada com várias funções básicas, incluindo DATE, TIME e BLANK, bem como um operador de adição entre DATE a TIME e um operador de subtração unário, de modo que os valores negativos possam ser expressos.</p> <p>Os seguintes valores são todos os valores válidos: 3, -5, BLANK(), "2009-04-15 02:45:21". Os valores podem não se referir a nada fora da expressão imediata e não podem se referir a colunas, tabelas, relações ou a qualquer outra coisa.</p> <p>Um valor ausente será tratado de forma idêntica a BLANK(). Por exemplo, as seguintes expressões são as mesmas: {1,2,BLANK(),4} {1,2,,4}</p>

Valor retornado

Uma tabela que declara um conjunto embutido de valores.

Comentários

Ao contrário de DATATABLE, o [Construtor de Tabela](#) permite qualquer expressão escalar como valor de entrada.

Exemplo

```
=DataTable("Name", STRING,  
          "Region", STRING  
          ,{  
              {" User1", "East"},  
              {" User2", "East"},  
              {" User3", "West"},  
              {" User4", "West"},  
              {" User4", "East"}  
          }  
      )
```

ERRO

09/05/2020 • 2 minutes to read

Gera um erro com uma mensagem de erro.

Sintaxe

```
ERROR(<text>)
```

Parâmetros

TERMO	DEFINIÇÃO
texto	Uma cadeia de texto que contém uma mensagem de erro.

Valor retornado

Nenhum

Comentários

A função ERROR pode ser colocada em uma expressão DAX em qualquer lugar em que um valor escalar é esperado.

Exemplos

Exemplo 1

A seguinte consulta DAX:

```
DEFINE
MEASURE DimProduct[Measure] =
    IF(
        SELECTEDVALUE(DimProduct[Color]) = "Red",
        ERROR("red color encountered"),
        SELECTEDVALUE(DimProduct[Color])
    )
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]
```

Falha e gera a mensagem de erro contendo "cor vermelha encontrada".

Exemplo 2

A seguinte consulta DAX:

```

DEFINE
MEASURE DimProduct[Measure] =
    IF(
        SELECTEDVALUE(DimProduct[Color]) = "Magenta",
        ERROR("magenta color encountered"),
        SELECTEDVALUE(DimProduct[Color])
    )
EVALUATE SUMMARIZECOLUMNS(DimProduct[Color], "Measure", [Measure])
ORDER BY [Color]

```

Retorna a seguinte tabela:

DIMPRODUCT[COLOR]	[MEASURE]
Preto	Preto
Azul	Azul
Cinza	Cinza
Multi	Multi
NA	NA
Vermelho	Vermelho
Prata	Prata
Prata\Preto	Prata\Preto
Branco	Branco
Amarelo	Amarelo

Como o magenta não é uma das cores do produto, a função ERROR não é executada.

EXCEPT

09/05/2020 • 2 minutes to read

Retorna as linhas de uma tabela que não aparecem em outra tabela.

Sintaxe

```
EXCEPT(<table_expression1>, <table_expression2>)
```

Parâmetros

TERMO	DEFINIÇÃO
Table_expression	Qualquer expressão DAX que retorna uma tabela.

Valor retornado

Uma tabela que contém as linhas de uma tabela menos todas as linhas de outra tabela.

Comentários

Se uma linha aparecer em ambas as tabelas, ela e suas duplicatas não estarão presentes no conjunto de resultados. Se uma linha for exibida somente em table_expression1, ela e suas duplicatas serão exibidas no conjunto de resultados.

Os nomes de coluna corresponderão aos nomes de coluna em table_expression1.

A tabela retornada tem linhagem com base nas colunas em table_expression1, independentemente da linhagem das colunas na segunda tabela. Por exemplo, se a primeira coluna da primeira table_expression tiver linhagem na coluna base C1 no modelo, Except reduzirá as linhas com base na disponibilidade dos valores na primeira coluna do segundo table_expression e manterá a linhagem na coluna base C1 intacta.

As duas tabelas devem ter o mesmo número de colunas.

As colunas são comparadas com base no posicionamento e na comparação de dados sem coerção de tipo.

O conjunto de linhas retornado depende da ordem das duas expressões.

A tabela retornada não inclui colunas de tabelas relacionadas a table_expression1.

Exemplo

States1

ESTADO
A
B
B

ESTADO
B
C
D
D

States2

ESTADO
B
C
D
D
D
E
E
E

Except(States1, States2)

ESTADO
A

Except(States2, States1)

ESTADO
E
E
E

GENERATESERIES

09/05/2020 • 2 minutes to read

Retorna uma tabela de coluna única contendo os valores de uma série aritmética, ou seja, uma sequência de valores em que cada uma difere da anterior por uma quantidade constante. O nome da coluna retornada é Valor.

Sintaxe

```
GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])
```

Parâmetros

TERMO	DEFINIÇÃO
startValue	O valor inicial usado para gerar a sequência.
endValue	O valor final usado para gerar a sequência.
incrementValue	(Opcional) O valor de incremento da sequência. Quando não fornecido, o valor padrão é 1.

Valor retornado

Uma tabela de coluna única que contém os valores de uma série aritmética. O nome da coluna é Valor.

Comentários

Quando startValue é menor que endValue, uma tabela vazia é retornada.

incrementValue deve ser um valor positivo.

A sequência é interrompida no último valor inferior ou igual a endValue.

Exemplos

Exemplo 1

A seguinte consulta DAX:

```
EVALUATE GENERATESERIES(1, 5)
```

Retorna a tabela a seguir com uma única coluna:

[VALOR]	
1	
2	

[VALOR]	
3	
4	
5	

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE GENERATESERIES(1.2, 2.4, 0.4)
```

Retorna a tabela a seguir com uma única coluna:

[VALOR]	
1.2	
1.6	
2	
2,4	

Exemplo 3

A seguinte consulta DAX:

```
EVALUATE GENERATESERIES(CURRENCY(10), CURRENCY(12.4), CURRENCY(0.5))
```

Retorna a tabela a seguir com uma única coluna:

[VALOR]	
10	
10,5	
11	
11,5	
12	

GROUPBY

09/05/2020 • 6 minutes to read

A função GROUPBY é semelhante à função SUMMARIZE. No entanto, GROUPBY não faz um CÁLCULO implícito para nenhuma coluna de extensão que ele adiciona. GROUPBY permite que uma nova função, CURRENTGROUP(), seja usada dentro das funções de agregação nas colunas de extensão que ele adiciona. GROUPBY tenta reutilizar os dados que foram agrupados, tornando-os com desempenho superior.

Sintaxe

```
GROUPBY (<table>, [<groupBy_columnName1>], [<name>, <expression>]... )
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
groupBy_columnName	O nome de uma coluna existente na tabela (ou em uma tabela relacionada) pela qual os dados serão agrupados. Esse parâmetro não pode ser uma expressão.
Nome	O nome dado a uma nova coluna que está sendo adicionada à lista de colunas GroupBy, entre aspas duplas.
expressão	<p>Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada para cada conjunto de valores GroupBy.</p> <p>Observação: A expressão usada em GroupBy pode incluir qualquer uma das funções de agregação "X", como SUMX, AVERAGEX, MINX, MAXX etc. e quando uma dessas funções é usada dessa forma, permitimos que o argumento de tabela (que normalmente deve ser uma expressão de tabela) seja substituído por uma função CURRENTGROUP() especial, conforme descrito em outro lugar neste documento.</p> <p>Restrições na expressão:</p> <ul style="list-style-type: none">– A função CALCULATE (e, portanto, medidas) não é permitida na expressão.– A função do CURRENTGROUP só pode ser usada no nível superior das verificações de tabela na expressão. Ou seja, SUMX(<table>,SUMX(CURRENTGROUP(...), ...)) não é permitido. ABS(SUMX(CURRENTGROUP(), [Column])) é permitido, pois o ABS não executa uma verificação.

Valor retornado

Uma tabela com as colunas selecionadas para os argumentos groupBy_columnName e as colunas agrupadas por designadas pelos argumentos de nome.

Comentários

A função GROUPBY faz o seguinte:

1. Comece com a tabela especificada (e todas as tabelas relacionadas na direção "para um").
2. Crie um agrupamento usando todas as colunas GroupBy (que devem existir na tabela da etapa #1.).
3. Cada grupo é uma linha no resultado, mas representa um conjunto de linhas na tabela original.
4. Para cada grupo, avalie as colunas de extensão que estão sendo adicionadas. Ao contrário da função SUMMARIZE, um CALCULATE implícito não é executado e o grupo não é colocado no contexto do filtro.

Observações:

- Cada coluna para a qual você define um nome deve ter uma expressão correspondente; caso contrário, um erro será retornado. O primeiro argumento, name, define o nome da coluna nos resultados. O segundo argumento, expression, define o cálculo executado para obter o valor de cada linha nessa coluna.
- groupBy_columnName deve estar em table ou em uma tabela relacionada.
- Cada nome deve ser colocado entre aspas duplas.
- A função agrupa um conjunto selecionado de linhas em um conjunto de linhas de resumo pelos valores de uma ou mais colunas de groupBy_columnName. Uma linha é retornada para cada grupo.

Opções

CURRENTGROUP()

CURRENTGROUP só pode ser usado em uma expressão que define uma coluna dentro da função GROUPBY. Em vigor, CURRENTGROUP retorna um conjunto de linhas do argumento "table" de GROUPBY que pertence à linha atual do resultado GROUPBY. A função CURRENTGROUP não usa argumentos e só tem suporte como o primeiro argumento para uma das seguintes funções de agregação: AverageX, CountAX, CountX, GeoMeanX, MaxX, MinX, ProductX, StDevX.S, StDevX.P, SumX, VarX.S, VarX.P.

Exemplo

Suponha que um modelo de dados tenha quatro tabelas: Vendas, Cliente, Produto, Geografia, em que Vendas está no lado "muitos" de uma relação com cada uma das outras três tabelas.

```
GROUPBY (  
    Sales,  
    Geography[Country],  
    Product[Category],  
    "Total Sales", SUMX( CURRENTGROUP(), Sales[Price] * Sales[Qty])  
)
```

Isso será iniciado com a tabela Vendas, estendida com todas as colunas de todas as tabelas relacionadas. Em seguida, ela criará um resultado com três colunas.

- A primeira coluna é cada um dos países para os quais há uma venda.
- A segunda coluna é cada categoria de produto para a qual há uma venda nesse país.
- A terceira coluna é a soma do valor das vendas (conforme calculado do preço * qtd) para o país e a categoria de produto selecionados.

Suponha que nós criamos o resultado anterior. Podemos usar GROUPBY novamente para localizar o número máximo de vendas da categoria dentro de cada país, conforme mostrado aqui.

```
DEFINE
VAR SalesByCountryAndCategory =
GROUPBY (
Sales,
Geography[Country],
Product[Category],
"Total Sales", SUMX( CURRENTGROUP(), Sales[Price] * Sales[Qty])
)

Evaluate GROUPBY (
SalesByCountryAndCategory,
Geography[Country],
"Max Sales", MAXX( CURRENTGROUP(), [Total Sales])
)
```

Consulte também

[Função SUMMARIZE \(DAX\)](#)

[Função SUMMARIZECOLUMNS \(DAX\)](#)

INTERSECT

09/05/2020 • 2 minutes to read

Retorna a interseção de linha entre duas tabelas, retendo as duplicatas.

Sintaxe

```
INTERSECT(<table_expression1>, <table_expression2>)
```

Parâmetros

TERMO	DEFINIÇÃO
Table_expression	Qualquer expressão DAX que retorna uma tabela.

Valor retornado

Uma tabela que contém todas as linhas de table_expression1 que também estão em table_expression2

Exceções

Comentários

A interseção não é comutativa. Em geral, Intersect(T1, T2) terá um conjunto de resultados diferente de Intersect(T2, T1).

Linhas duplicadas serão preservadas. Se uma linha aparecer em table_expression1 e table_expression2, ela e todas as duplicatas de table_expression_1 serão incluídas no conjunto de resultados.

Os nomes de coluna corresponderão aos nomes de coluna em table_expression1.

A tabela retornada tem linhagem com base nas colunas em table_expression1, independentemente da linhagem das colunas na segunda tabela. Por exemplo, se a primeira coluna da primeira table_expression tiver linhagem na coluna de base C1 do modelo, a interseção reduzirá as linhas com base na interseção na primeira coluna da segunda table_expression e manterá a linhagem na coluna de base C1 intacta.

As colunas são comparadas com base no posicionamento e na comparação de dados sem coerção de tipo.

A tabela retornada não inclui colunas de tabelas relacionadas a table_expression1.

Exemplo

States1

ESTADO
A
A

ESTADO
B
B
B
C
D
D

States2

ESTADO
B
C
D
D
D
E

Intersect(States1, States2)

ESTADO
B
B
B
C
D
D

Intersect(States2, States1)

ESTADO
B

ESTADO
C
D
D
D

ISEMPTY

09/05/2020 • 2 minutes to read

Verifica se uma tabela está vazia.

Sintaxe

```
ISEMPTY(<table_expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
table_expression	Uma referência de tabela ou uma expressão DAX que retorna uma tabela.

Valor retornado

True se a tabela estiver vazia (não tem linhas); caso contrário, False.

Exemplo

Para a tabela abaixo denominada 'Info':

PAÍS	ESTADO	MUNICÍPIO	TOTAL
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
EUA	AC	5	500
EUA	WA	10	900

```
EVALUATE  
ROW("Any countries with count > 25?", NOT(ISEMPTY(FILTER(Info, [Count]>25))))
```

Valor retornado: FALSE

ISSELECTEDMEASURE

08/05/2020 • 2 minutes to read

Usada por expressões para itens de cálculo a fim de determinar se a medida que está no contexto é uma das especificadas em uma lista de medidas.

NOTE

Essa função atualmente se aplica somente ao [SQL Server 2019 Analysis Services CTP 2.3](#) (e versões posteriores).

Sintaxe

```
ISSELECTEDMEASURE( M1, M2, ... )
```

Parâmetros

TERMO	DEFINIÇÃO
M1, M2, ...	Uma lista de medidas.

Valor retornado

Um booleano que indica se a medida que está atualmente no contexto é uma das especificadas na lista de parâmetros.

Comentários

Só pode ser referenciada na expressão para um item de cálculo.

Exemplo

A expressão de item de cálculo a seguir verifica se a medida atual é uma das especificadas na lista de parâmetros. Se as medidas forem renomeadas, a correção de fórmulas refletirá as alterações de nome da expressão.

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Consulte também

[SELECTEDMEASURE](#)

[SELECTEDMEASURENAME](#)

NATURALINNERJOIN

09/05/2020 • 2 minutes to read

Executa uma junção interna de uma tabela com outra tabela. As tabelas são unidas em colunas comuns (por nome) nas duas tabelas. Se as duas tabelas não tiverem nomes de coluna comuns, um erro será retornado.

Sintaxe

```
NATURALINNERJOIN(<leftJoinTable>, <rightJoinTable>)
```

Parâmetros

TERMO	DEFINIÇÃO
leftJoinTable	Uma expressão de tabela que define a tabela no lado esquerdo da junção.
rightJoinTable	Uma expressão de tabela que define a tabela no lado direito da junção.

Valor retornado

Uma tabela que inclui somente linhas para as quais os valores nas colunas comuns especificadas também estão presentes nas duas tabelas. A tabela retornada terá as colunas comuns da tabela esquerda e as outras colunas das duas tabelas.

Comentários

Não há nenhuma garantia de ordem de classificação para os resultados.

As colunas que estão sendo unidas devem ter o mesmo tipo de dados nas duas tabelas.

Somente as colunas da mesma tabela de origem (com a mesma linhagem) são unidas. Por exemplo, Products[ProductID], WebSales[ProductID], StoreSales[ProductID], com relações muitos para um entre WebSales e StoreSales e a tabela Products com base na coluna ProductID e nas tabelas WebSales e StoreSales são unidas em [ProductID].

Uma semântica de comparação estrita é usada durante a junção. Não há coerção de tipo; por exemplo, 1 não é igual a 1.0.

NATURALLEFTOUTERJOIN

09/05/2020 • 2 minutes to read

Executa uma junção interna de uma tabela com outra tabela. As tabelas são unidas em colunas comuns (por nome) nas duas tabelas. Se as duas tabelas não tiverem nomes de coluna comuns, um erro será retornado.

Sintaxe

```
NATURALLEFTOUTERJOIN(<leftJoinTable>, <rightJoinTable>)
```

Parâmetros

TERMO	DEFINIÇÃO
leftJoinTable	Uma expressão de tabela que define a tabela no lado esquerdo da junção.
rightJoinTable	Uma expressão de tabela que define a tabela no lado direito da junção.

Valor retornado

Uma tabela que inclui somente linhas de rightJoinTable para as quais os valores nas colunas comuns especificadas também estão presentes em leftJoinTable. A tabela retornada terá as colunas comuns da tabela esquerda e as outras colunas das duas tabelas.

Comentários

Não há nenhuma garantia de ordem de classificação para os resultados.

As colunas que estão sendo unidas devem ter o mesmo tipo de dados nas duas tabelas.

Somente as colunas da mesma tabela de origem (com a mesma linhagem) são unidas. Por exemplo, Products[ProductID], WebSales[ProductID], StoreSales[ProductID], com relações muitos para um entre WebSales e StoreSales e a tabela Products com base na coluna ProductID e nas tabelas WebSales e StoreSales são unidas em [ProductID].

Uma semântica de comparação estrita é usada durante a junção. Não há coerção de tipo; por exemplo, 1 não é igual a 1.0.

SELECTEDMEASURE

08/05/2020 • 2 minutes to read

Usada por expressões de itens de cálculo para referenciar a medida que está no contexto.

NOTE

Essa função atualmente se aplica somente ao [SQL Server 2019 Analysis Services CTP 2.3](#) (e versões posteriores).

Sintaxe

```
SELECTEDMEASURE()
```

Parâmetros

Nenhum

Valor retornado

Uma referência à medida que está atualmente no contexto quando o item de cálculo é avaliado.

Comentários

Só pode ser referenciada na expressão para um item de cálculo.

Exemplo

A expressão do item de cálculo a seguir calcula o acumulado do ano até a data atual, seja qual for a medida em contexto.

```
CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))
```

Consulte também

[SELECTEDMEASURENAME](#)

[ISSELECTEDMEASURE](#)

SELECTEDMEASUREFORMATSTRING

09/05/2020 • 2 minutes to read

Usado por expressões para itens de cálculo para recuperar a cadeia de caracteres de formato da medida que está no contexto.

NOTE

Essa função atualmente se aplica somente ao [SQL Server 2019 Analysis Services CTP 2.3](#) (e versões posteriores).

Sintaxe

```
SELECTEDMEASUREFORMATSTRING()
```

Parâmetros

Nenhum

Valor retornado

Uma cadeia de caracteres que contém a cadeia de caracteres de formato da medida que está atualmente no contexto quando o item de cálculo é avaliado.

Comentários

Esta função só pode ser referenciada em expressões para itens de cálculo em grupos de cálculo. Ele foi projetado para ser usado pela propriedade de **expressão de cadeia de caracteres de formato** de itens de cálculo.

Exemplo

A expressão a seguir é avaliada pela propriedade de expressão de cadeia de caracteres de formato de um item de cálculo. Se houver uma única moeda no contexto de filtro, a cadeia de caracteres de formato será recuperada da coluna DimCurrency [FormatString]; caso contrário, a cadeia de caracteres de formato da medida no contexto será usada.

```
SELECTEDVALUE( DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING() )
```

Consulte também

[SELECTEDMEASURE](#)

[ISSELECTEDMEASURE](#)

SELECTEDMEASURENAME

08/05/2020 • 2 minutes to read

Usada por expressões de itens de cálculo para determinar a medida que está no contexto por nome.

NOTE

Essa função atualmente se aplica somente ao [SQL Server 2019 Analysis Services CTP 2.3](#) (e versões posteriores).

Sintaxe

```
SELECTEDMEASURENAME()
```

Parâmetros

Nenhum

Valor retornado

Um valor de cadeia de caracteres que contém o nome da medida que está atualmente no contexto quando o item de cálculo é avaliado.

Comentários

Só pode ser referenciada na expressão para um item de cálculo.

Essa função é geralmente usada para fins de depuração durante a criação de grupos de cálculo.

Exemplo

A expressão de item de cálculo a seguir verifica se a medida atual é a taxa de despesas e aplica condicionalmente a lógica de cálculo. Como a verificação se baseia em uma comparação de cadeia de caracteres, ela não está sujeita à correção de fórmulas e não se beneficiará do fato da renomeação de objeto ser refletida automaticamente. Para ver uma comparação semelhante que se beneficiaria da correção de fórmulas, confira a função `ISSLECTEDMEASURE`.

```
IF (
    SELECTEDMEASURENAME = "Expense Ratio",
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```

Consulte também

[SELECTEDMEASURE](#)

[ISSLECTEDMEASURE](#)

SUMMARIZECOLUMNS

08/05/2020 • 11 minutes to read

Retorna uma tabela de resumo por um conjunto de grupos.

Sintaxe

```
SUMMARIZECOLUMNS( <groupBy_columnName> [, <groupBy_columnName >]..., [<filterTable>]...[, <name>, <expression>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
groupBy_columnName	Uma referência de coluna totalmente qualificada (Tabela[Coluna]) para uma tabela de base para a qual os valores distintos são incluídos na tabela retornada. Cada coluna de groupBy_columnName é unida de forma cruzada (tabelas diferentes) ou de existência automática (na mesma tabela) com as colunas especificadas posteriores.
filterTable	Uma expressão de tabela que é adicionada ao contexto de filtro de todas as colunas especificadas como argumentos de groupBy_columnName. Os valores presentes na tabela de filtro são usados para filtrar antes que a junção cruzada/existência automática seja executada.
Nome	Uma cadeia de caracteres que representa o nome da coluna a ser usada para a expressão posterior especificada.
expressão	Qualquer expressão DAX que retorna um único valor (não é uma tabela).

Valor retornado

Uma tabela que inclui combinações de valores das colunas fornecidas, com base no agrupamento especificado. Somente linhas para as quais pelo menos uma das expressões fornecidas retorna um valor não em branco são incluídas na tabela retornada. Se todas as expressões forem avaliadas como em branco/nulo para uma linha, essa linha não será incluída na tabela retornada.

Comentários

SUMMARIZECOLUMNS não garante nenhuma ordem de classificação para os resultados.

Não é possível especificar uma coluna mais de uma vez no parâmetro groupBy_columnName. Por exemplo, as fórmulas a seguir são inválidas.

```
SUMMARIZECOLUMNS( Sales[StoreId], Sales[StoreId] )
```

Contexto de filtro

Considere a consulta a seguir:

```
SUMMARIZECOLUMNS ( 'Sales Territory'[Category], FILTER('Customer', 'Customer' [First Name] = "Alicia") )
```

Nessa consulta, sem uma medida, as colunas groupBy não contêm nenhuma coluna da expressão de filtro (ou seja, da tabela Cliente). O filtro não é aplicado às colunas groupBy. A Região de Vendas e a tabela Cliente podem estar indiretamente relacionadas por meio da tabela de fatos de vendas do Revendedor. Como elas não estão diretamente relacionadas, a expressão de filtro não é operacional e as colunas groupBy não são afetadas.

No entanto, com esta consulta:

```
SUMMARIZECOLUMNS ( 'Sales Territory'[Category], 'Customer' [Education], FILTER('Customer', 'Customer'[First Name] = "Alicia") )
```

As colunas groupBy contêm uma coluna que é afetada pelo filtro e esse filtro é aplicado aos resultados de groupBy.

Opções

IGNORE

A sintaxe IGNORE() pode ser usada para modificar o comportamento da função SUMMARIZECOLUMNS omitindo expressões específicas da avaliação BLANK/NULL. As linhas para as quais todas as expressões que não usam IGNORE retornam BLANK/NULL serão excluídas independentemente de as expressões que usam IGNORE serem avaliadas como BLANK/NULL ou não.

Sintaxe

```
IGNORE(<expression>)
```

Com SUMMARIZECOLUMNS

```
SUMMARIZECOLUMNS(<groupBy_columnName>[, <groupBy_columnName >]..., [<filterTable>]...[, <name>, IGNORE(...)]...)
```

Parâmetros

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorna um único valor (não é uma tabela).

Valor retornado

Essa função não retorna um valor.

Comentários

IGNORE pode ser usado como um argumento de expressão para SUMMARIZECOLUMNS.

Exemplo

```
SUMMARIZECOLUMNS( Sales[CustomerId], "Total Qty", IGNORE( SUM( Sales[Qty] ) ), "BlankIfTotalQtyIsNot3", IF( SUM( Sales[Qty] )=3, 3 ) )
```

Isso acumula a coluna Sales[CustomerId], criando um subtotal para todos os clientes no agrupamento determinado. Sem IGNORE, o resultado é:

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
A	5	
B	3	3
C	3	3

Com IGNORE

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
B	3	3
C	3	3

Toda a expressão ignorada

```
SUMMARIZECOLUMNS( Sales[CustomerId], "Blank", IGNORE( Blank() ), "BlankIfTotalQtyIsNot5", IGNORE( IF( SUM( Sales[Qty] )=5, 5 ) ) )
```

Embora ambas as expressões retornem em branco para algumas linhas, elas são incluídas, já que não há nenhuma expressão não ignorada que retorne em branco.

CUSTOMERID	TOTALQTY	BLANKIFTOTALQTYISNOT3
A		5
B		
C		

NONVISUAL

Marca um filtro de valor na função SUMMARIZECOLUMNS como sem afetar os valores da medida, mas aplicando-se apenas às colunas agrupar por.

Sintaxe

```
NONVISUAL(<expression>)
```

Valor retornado

Uma tabela de valores.

Exemplo


```

DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    NONVISUAL(TREATAS({2007, 2008}, DimDate[CalendarYear])),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales], ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]

```

Resultado

Retorna o resultado em que [Vendas Totais Visuais] é o total em todos os anos:

DIMDATE[CALENDARYEAR]	[VENDAS]	[VENDAS TOTAIS VISUAIS]
2007	9.791.060,30	29.358.677,22
2008	9.770.899,74	29.358.677,22

Por outro lado, a mesma consulta sem a função NONVISUAL:

```

DEFINE
MEASURE FactInternetSales[Sales] = SUM(FactInternetSales[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    TREATAS({2007, 2008}, DimDate[CalendarYear]),
    "Sales", [Sales],
    "Visual Total Sales", CALCULATE([Sales], ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear]

```

Resultado

Retorna o resultado em que [Vendas Totais Visuais] é o total nos dois anos selecionados:

DIMDATE[CALENDARYEAR]	[VENDAS]	[VENDAS TOTAIS VISUAIS]
2007	9.791.060,30	19.561.960,04
2008	9.770.899,74	19.561.960,04

ROLLUPADDISSUBTOTAL()

A adição da sintaxe ROLLUPADDISSUBTOTAL() modifica o comportamento da função SUMMARIZECOLUMNS acrescentando linhas acumuladas/de subtotal ao resultado com base nas colunas `groupBy_columnName`.

Sintaxe

```

ROLLUPADDISSUBTOTAL ( [<filter> ..., ] <groupBy_columnName>, <isSubtotal_columnName>[, <filter> ...][,
<groupBy_columnName >, <isSubtotal_columnName>[, <filter> ...]] )

```

Parâmetros

TERMO	DEFINIÇÃO
groupBy_columnName	O nome qualificado de uma coluna existente a ser usada para criar grupos de resumo com base nos valores encontrados nela. Esse parâmetro não pode ser uma expressão.
isSubtotal_columnName	O nome da coluna booliana a ser adicionada ao resultado, indicando se uma linha é ou não um subtotal da coluna groupBy (ou colunas quando usada com ROLLUPGROUP). Esse valor é calculado usando a função ISSUBTOTAL.
filter	Uma expressão de tabela que é adicionada ao contexto de filtro no nível de agrupamento atual. Um filtro antes que a primeira coluna group-by seja aplicada ao nível total geral.

Valor retornado

A função não retorna um valor. Ele especifica apenas o conjunto de colunas a ser subtotalizado.

Exemplo

Subtotal único

```

DEFINE
VAR vCategoryFilter =
  TREATAS({"Accessories", "Clothing"}, Product[Category])
VAR vSubcategoryFilter =
  TREATAS({"Bike Racks", "Mountain Bikes"}, Product[Subcategory])
EVALUATE
  SUMMARIZECOLUMNS
  (
    ROLLUPADDISSUBTOTAL
    (
      Product[Category], "IsCategorySubtotal", vCategoryFilter,
      Product[Subcategory], "IsSubcategorySubtotal", vSubcategoryFilter
    ),
    "Total Qty", SUM(Sales[Qty])
  )
ORDER BY
  [IsCategorySubtotal] DESC, [Category],
  [IsSubcategorySubtotal] DESC, [Subcategory]

```

Resultado

CATEGORIA	SUBCATEGORIA	ISCATEGORYSUBTOTAL	ISSUBCATEGORYSUBTOTAL	TOTAL QTY
		True	True	60398
Acessórios		Falso	True	36092
Acessórios	Racks de bicicleta	Falso	Falso	328
Bicicletas	Mountain bikes	Falso	Falso	4970
Vestuário		Falso	True	9101

Vários subtotais

```
SUMMARIZECOLUMNS ( Regions[State], ROLLUPADDISSUBTOTAL ( Sales[CustomerId], "IsCustomerSubtotal" ),  
ROLLUPADDISSUBTOTAL ( Sales[Date], "IsDateSubtotal"), "Total Qty", SUM( Sales[Qty] ) )
```

As vendas são agrupadas por estado, por cliente, por data, com subtotais para 1. Vendas por estado, por data 2. Vendas por estado, por cliente 3. Acumulado tanto no cliente quanto na data que levam às vendas por estado.

Resultado

CUSTOMERID	ISCUSTOMERSUBTOTAL	ESTADO	TOTAL QTY	DATA	ISDATESUBTOTAL
A	FALSO	WA	5	10/7/2014	
B	FALSO	WA	1	10/7/2014	
B	FALSO	WA	2	11/7/2014	
C	FALSO	OR	2	10/7/2014	
C	FALSO	OR	1	11/7/2014	
	VERDADEIRO	WA	6	10/7/2014	
	VERDADEIRO	WA	2	11/7/2014	
	VERDADEIRO	OR	2	10/7/2014	
	VERDADEIRO	OR	1	11/7/2014	
A	FALSO	WA	5		VERDADEIRO
B	FALSO	WA	3		VERDADEIRO
C	FALSO	OR	3		VERDADEIRO
	VERDADEIRO	WA	8		VERDADEIRO
	VERDADEIRO	OR	3		VERDADEIRO

ROLLUPGROUP()

Assim como com a função SUMMARIZE, ROLLUPGROUP pode ser usado junto com ROLLUPADDISSUBTOTAL para especificar quais grupos/granularidades de resumo (subtotais) incluir (reduzindo o número de linhas de subtotal retornadas).

Sintaxe

```
ROLLUPGROUP(<groupBy_columnName>, <groupBy_columnName>)
```

Com ROLLUPADDISSUBTOTAL

```
ROLLUPADDISSUBTOTAL( ROLLUPGROUP(...), isSubtotal_columnName[, <groupBy_columnName>...] )
```

Parâmetros

TERMO	DEFINIÇÃO
groupBy_columnName	O nome qualificado de uma coluna existente a ser usada para criar grupos de resumo com base nos valores encontrados nela. O conjunto de colunas de agrupamento fornecido para a função ROLLUPGROUP definirá a granularidade com a qual retornar as linhas de subtotal (mesmo comportamento de quando ROLLUP e ROLLUPGROUP são usados com a função SUMMARIZE). Esse parâmetro não pode ser uma expressão.

Valor retornado

A função não retorna um valor. Ela marca um conjunto de colunas a ser agrupado durante a subtotalização por ROLLUPADDISSUBTOTAL.

Comentários

ROLLUPGROUP só pode ser usado como um argumento groupBy_columnName para ROLLUPADDISSUBTOTAL ou para a função resumir.

Exemplo

Vários subtotais

```
SUMMARIZECOLUMNS( ROLLUPADDISSUBTOTAL( Sales[CustomerId], "IsCustomerSubtotal" ),
ROLLUPADDISSUBTOTAL(ROLLUPGROUP(Regions[City], Regions[State]), "IsCityStateSubtotal"), "Total Qty", SUM(
Sales[Qty] ) )
```

Ainda agrupados por cidade e estado, mas são combinados ao relatar um subtotal.

Resultado

ESTADO	CUSTOMERID	ISCUSTOMERSUB TOTAL	TOTAL QTY	CIDADE	ISCITYSTATESUBT OTAL
WA	A	FALSO	2	Bellevue	FALSO
WA	B	FALSO	2	Bellevue	FALSO
WA	A	FALSO	3	Redmond	FALSO
WA	B	FALSO	1	Redmond	FALSO
OR	C	FALSO	3	Portland	FALSO
WA		VERDADEIRO	4	Bellevue	FALSO
WA		VERDADEIRO	4	Redmond	FALSO
OR		VERDADEIRO	3	Portland	FALSO
	A	FALSO	5		FALSO
	B	FALSO	3		VERDADEIRO
	C	FALSO	3		VERDADEIRO
		VERDADEIRO	11		VERDADEIRO

Consulte também

Função SUMMARIZE (DAX)

Construtor de Tabela

08/05/2020 • 2 minutes to read

Retorna uma tabela de uma ou mais colunas.

Sintaxe

```
{ <scalarExpr1>, <scalarExpr2>, ... }  
{ ( <scalarExpr1>, <scalarExpr2>, ... ), ( <scalarExpr1>, <scalarExpr2>, ... ), ... }
```

Parâmetros

TERMO	DEFINIÇÃO
scalarExprN	Qualquer expressão DAX que retorna um valor escalar.

Valor retornado

Uma tabela com uma ou mais colunas. Quando há apenas uma coluna, seu nome é Value. Quando há N colunas em que $N > 1$, os nomes das colunas, da esquerda para a direita, são Value1, Value2,..., ValueN.

Comentários

A primeira sintaxe retorna uma tabela com uma única coluna. A segunda sintaxe retorna uma tabela com uma ou mais colunas.

O número de expressões escalares deve ser o mesmo para todas as linhas.

Quando os tipos de dados dos valores de uma coluna são diferentes em linhas diferentes, todos os valores são convertidos em um tipo de dados comum.

Exemplos

Exemplo 1

As seguintes consultas DAX:

```
EVALUATE { 1, 2, 3 }
```

e

```
EVALUATE { (1), (2), (3) }
```

Retornam a tabela a seguir com uma única coluna:

[VALOR]	
1	

[VALOR]	
2	
3	

Exemplo 2

A seguinte consulta DAX:

```
EVALUATE
{
    (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),
    (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),
    (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")
}
```

[VALUE1]	[VALUE2]	[VALUE3]	[VALUE4]
1.5	01/01/2017	199,99	A
Row2	02/01/2017	249,99	B
Row3	3/1/2017	299,99	C

Exemplo 3

A seguinte consulta DAX:

```
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }
```

Retorna a tabela a seguir de uma única coluna do tipo de dados String:

[VALOR]	
1	
01/01/2017	
VERDADEIRO	
A	

TREATAS

26/05/2020 • 2 minutes to read

Aplica o resultado de uma expressão de tabela como filtros a colunas de uma tabela não relacionada.

Sintaxe

```
TREATAS(table_expression, <column>[, <column>[, <column>[,...]]])
```

Parâmetros

TERMO	DEFINIÇÃO
table_expression	Uma expressão que resulta em uma tabela.
coluna	Uma ou mais colunas existentes. Não pode ser uma expressão.

Valor retornado

Uma tabela que contém todas as linhas em colunas que também estão em table_expression.

Comentários

O número de colunas especificado deve corresponder ao número de colunas na expressão de tabela e estar na mesma ordem.

Se um valor retornado na expressão de tabela não existir na coluna, ele será ignorado. Por exemplo, TREATAS({"Red", "Green", "Yellow"}, DimProduct[Color]) define um filtro na coluna DimProduct[Color] com três valores "Red", "Green" e "Yellow". Se "Yellow" não existir em DimProduct[Color], os valores de filtro reais serão "Red" e "Green".

Funciona melhor quando não há uma relação entre as tabelas. Se houver várias relações entre as tabelas envolvidas, considere usar [USERELATIONSHIP](#) em vez disso.

Exemplos

No exemplo a seguir, o modelo contém duas tabelas de produtos não relacionadas. Se um usuário aplicar um filtro a DimProduct1[ProductCategory] selecionando Bikes, Seats, Tires, o mesmo filtro Bikes, Seats, Tires será aplicado a DimProduct2[ProductCategory].

```
CALCULATE(  
    SUM(Sales[Amount]),  
    TREATAS(VALUES(DimProduct1[ProductCategory]), DimProduct2[ProductCategory])  
)
```

Consulte também

[Função INTERSECT](#)

[Função FILTER](#)

UNION

08/05/2020 • 2 minutes to read

Cria uma tabela de união (junção) de um par de tabelas.

Sintaxe

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
table_expression	Qualquer expressão DAX que retorna uma tabela.

Valor retornado

Uma tabela que contém todas as linhas de cada uma das duas expressões de tabela.

Comentários

As duas tabelas devem ter o mesmo número de colunas.

As colunas são combinadas por posição em suas respectivas tabelas.

Os nomes de coluna na tabela retornada corresponderão aos nomes de coluna em table_expression1.

Linhas duplicadas serão preservadas.

A tabela retornada tem linhagem sempre que possível. Por exemplo, se a primeira coluna de cada table_expression tiver linhagem na mesma coluna de base C1 no modelo, a primeira coluna no resultado de UNION terá a linhagem C1. No entanto, se as colunas combinadas tiverem linhagem em colunas de base diferentes ou se houver uma coluna de extensão, a coluna resultante em UNION não terá nenhuma linhagem.

Quando os tipos de dados forem diferentes, o tipo de dados resultante será determinado com base nas regras de coerção de tipo de dados.

A tabela retornada não conterá colunas de tabelas relacionadas.

Exemplo

A expressão a seguir cria uma união combinando a tabela `USAINVENTORY` e a tabela `INDINVENTORY` em uma única tabela:

```
UNION(UsaInventory, IndInventory)
```

USAINVENTORY

PAÍS	ESTADO	CONTAGEM	TOTAL
EUA	AC	5	500
EUA	WA	10	900

INDInventory

PAÍS	ESTADO	CONTAGEM	TOTAL
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Tabela retornada

PAÍS	ESTADO	CONTAGEM	TOTAL
EUA	AC	5	500
EUA	WA	10	900
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900

Funções pai e filho

09/05/2020 • 2 minutes to read

Essas funções gerenciam dados que são apresentados como hierarquias pai/filho. Para saber mais, confira [Compreensão das funções para hierarquias pai-filho no DAX](#).

Nesta categoria

FUNÇÃO	DESCRIÇÃO
PATH	Retorna uma cadeia de caracteres de texto delimitada que contém os identificadores de todos os pais do identificador atual.
PATHCONTAINS	Retornará TRUE se o <i>item</i> especificado existir no <i>path</i> especificado.
PATHITEM	Retorna o item na <i>posição</i> especificada de uma cadeia de caracteres resultante da avaliação de uma função PATH.
PATHITEMREVERSE	Retorna o item na <i>posição</i> especificada de uma cadeia de caracteres resultante da avaliação de uma função PATH.
PATHLENGTH	Retorna o número de pais para o item especificado em um determinado resultado de PATH, incluindo self.

Compreensão das funções para hierarquias pai-filho no DAX

09/05/2020 • 6 minutes to read

O DAX fornece cinco funções para ajudar os usuários a gerenciar dados apresentados como uma hierarquia pai-filho em seus modelos. Com essas funções, um usuário pode obter a linhagem inteira de pais de uma linha, quantos níveis têm a linhagem até o primeiro pai, quem são os níveis *n* pai acima da linha atual, quem é o descendente *n* da parte superior da hierarquia de linhas atual e se determinado pai é pai na hierarquia de linhas atual.

Funções pai-filho no DAX

A tabela a seguir contém uma hierarquia pai-filho nas colunas: **EmployeeKey** e **ParentEmployeeKey** usadas em todos os exemplos de função.

EMPLOYEEKEY	PARENTEMPLOYEEKEY
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

Na tabela acima, você pode ver que o funcionário 112 não tem nenhum pai definido, o funcionário 14 tem o funcionário 112 como gerente (ParentEmployeeKey), o funcionário 3 tem o funcionário 14 como gerente e funcionários 11, 13 e 162 têm o funcionário 3 como gerente. A tabela acima ajuda a entender que o funcionário 112 não tem gerente acima dele e ele é o gerente principal de todos os funcionários mostrados aqui. Além disso, o funcionário 3 é subordinado ao funcionário 14 e os funcionários 11, 13, 162 ao 3.

A tabela a seguir apresenta as funções disponíveis, uma breve descrição da função e um exemplo da função em relação aos mesmos dados mostrados acima.

Função PATH (DAX): retorna um texto delimitado com os identificadores de todos os pais da linha atual, começando com o mais antigo ou mais alto até o mais atual.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	CAMINHO
112		112
14	112	112 14
3	14	112 14 3
11	3	112 14 3 11
13	3	112 14 3 13
162	3	112 14 3 162
117	162	112 14 3 162 117
221	162	112 14 3 162 221
81	162	112 14 3 162 81

Função PATHLENGTH (DAX): retorna o número de níveis em um determinado PATH(), começando no nível atual até o nível pai mais antigo ou superior. No exemplo a seguir, a coluna PathLength é definida como "
 =PATHLENGTH([Path])". O exemplo inclui todos os dados do exemplo Path() para ajudar a entender como essa função funciona.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	CAMINHO	PATHLENGTH
112		112	1
14	112	112 14	2
3	14	112 14 3	3
11	3	112 14 3 11	4
13	3	112 14 3 13	4
162	3	112 14 3 162	4
117	162	112 14 3 162 117	5
221	162	112 14 3 162 221	5
81	162	112 14 3 162 81	5

Função PATHITEM (DAX): retorna o item na posição especificada de um resultado semelhante a PATH(), contando da esquerda para a direita. Na coluna de exemplo a seguir, o PathItem – 4ª da esquerda é definido como "
 =PATHITEM([Path], 4)". Esse exemplo retorna o EmployeeKey na quarta posição na cadeia de caracteres Path começando da esquerda, usando os mesmos dados de exemplo do exemplo Path().

EMPLOYEEKEY	PARENTEMPLOYEEKEY	CAMINHO	PATHITEM – 4º COMEÇANDO DA ESQUERDA
112		112	
14	112	112 14	
3	14	112 14 3	
11	3	112 14 3 11	11
13	3	112 14 3 13	13
162	3	112 14 3 162	162
117	162	112 14 3 162 117	162
221	162	112 14 3 162 221	162
81	162	112 14 3 162 81	162

Função PATHITEMREVERSE (DAX): retorna o item na *posição* de um resultado de função semelhante à PATH(), contando de forma inversa da direita para a esquerda.

Na coluna de exemplo a seguir, o PathItemReverse – 3ª da direita é definido como "`=PATHITEMREVERSE([Path], 3)`". Esse exemplo retorna o EmployeeKey na terceira posição na cadeia de caracteres Path começando da direita, usando os mesmos dados de exemplo do exemplo Path().

EMPLOYEEKEY	PARENTEMPLOYEEKEY	CAMINHO	PATHITEMREVERSE – 3ª COMEÇANDO DA DIREITA
112		112	
14	112	112 14	
3	14	112 14 3	112
11	3	112 14 3 11	14
13	3	112 14 3 13	14
162	3	112 14 3 162	14
117	162	112 14 3 162 117	3
221	162	112 14 3 162 221	3
81	162	112 14 3 162 81	3

Função PATHCONTAINS (DAX): retornará **TRUE** se o *item* especificado existir no *caminho* especificado. Na coluna de exemplo a seguir, PathContains – Funcionário 162 é definido como "`=PATHCONTAINS([Path], "162")`". Esse exemplo retornará **TRUE** se o caminho fornecido contiver o funcionário 162. Este exemplo usa os resultados do exemplo Path() acima.

EMPLOYEEKEY	PARENTEMPLOYEEKEY	CAMINHO	PATHCONTAINS – FUNCIONÁRIO 162
112		112	FALSO
14	112	112 14	FALSO
3	14	112 14 3	FALSO
11	3	112 14 3 11	FALSO
13	3	112 14 3 13	FALSO
162	3	112 14 3 162	VERDADEIRO
117	162	112 14 3 162 117	VERDADEIRO

PATH

08/05/2020 • 4 minutes to read

Retorna uma cadeia de caracteres de texto delimitado com os identificadores de todos os pais do identificador atual, começando com o mais antigo até o mais atual.

Sintaxe

```
PATH(<ID_columnName>, <parent_columnName>)
```

Parâmetros

TERMO	DEFINIÇÃO
ID_columnName	O nome de uma coluna existente que contém o identificador exclusivo das linhas na tabela. Não pode ser uma expressão. O tipo de dados do valor em <i>ID_columnName</i> deve ser texto ou inteiro e também deve ser o mesmo tipo de dados que a coluna referenciada em <i>parent_columnName</i> .
parent_columnName	O nome de uma coluna existente que contém o identificador exclusivo do pai da linha atual. Não pode ser uma expressão. O tipo de dados do valor em <i>parent_columnName</i> deve ser texto ou inteiro e deve ser o mesmo tipo de dados que o valor em <i>parent_columnName</i> .

Valor retornado

Uma cadeia de caracteres de texto delimitada que contém os identificadores de todos os pais do identificador atual.

Comentários

Essa função é usada em tabelas que têm algum tipo de hierarquia interna, para retornar os itens relacionados ao valor da linha atual. Por exemplo, em uma tabela de Funcionários que contém funcionários, gerentes de funcionários e gerentes de gerentes, você pode retornar o caminho que conecta um funcionário ao seu gerente.

O caminho não fica restrito a um único nível de relações pai-filho; ele pode retornar linhas relacionadas que estão vários níveis acima da linha inicial especificada.

- O delimitador usado para separar os ascendentes é a barra vertical, "|".
- Os valores em *ID_columnName* e *parent_columnName* devem ter o mesmo tipo de dados, texto ou inteiro.
- Os valores em *parent_columnName* devem estar presentes em *ID_columnName*. Ou seja, você não poderá pesquisar um pai se não houver nenhum valor no nível filho.
- Se *parent_columnName* estiver em branco, PATH() retornará o valor de *ID_columnName*. Ou seja, se você procurar o gerente de um funcionário, mas a coluna *parent_columnName* não tiver dados, a função PATH retornará apenas a ID do funcionário.
- Se *ID_columnName* tiver duplicatas e *parent_columnName* for igual para essas duplicatas, PATH() retornará

o valor comum *parent_columnName*. No entanto, se o valor *parent_columnName* for diferente para essas duplicatas, PATH() retornará um erro. Em outras palavras, se você tiver duas listagens para a mesma ID de funcionário e elas tiverem a mesma ID de gerente, a função PATH retornará a ID desse gerente. No entanto, se houver duas IDs de funcionário idênticas que tenham IDs de gerentes diferentes, a função PATH retornará um erro.

- Se *ID_columnName* estiver em branco, PATH() retornará BLANK.
- Se *ID_columnName* contiver uma barra vertical "|", PATH() retornará um erro.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir cria uma coluna calculada que lista todos os gerentes de cada funcionário.

```
=PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey])
```

PATHCONTAINS

09/05/2020 • 2 minutes to read

Retornará **TRUE** se o *item* especificado existir no *path* especificado.

Sintaxe

```
PATHCONTAINS(<path>, <item>)
```

Parâmetros

path

Uma cadeia de caracteres criada como o resultado da avaliação de uma função PATH.

item

Uma expressão de texto a ser procurada no resultado de path.

Valor retornado

Um valor de **TRUE** se *item* existir em *path*; caso contrário **FALSE**.

Comentários

Se *item* for um número inteiro, ele será convertido em texto e a função será avaliada. Se a conversão falhar, a função retornará um erro.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir cria uma coluna calculada que usa uma ID de gerente e verifica um conjunto de funcionários. Se a ID do gerente estiver entre a lista de gerentes retornada pela função PATH, a função PATHCONTAINS retornará true; caso contrário, retornará false.

```
=PATHCONTAINS(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), "23")
```

PATHITEM

09/05/2020 • 2 minutes to read

Retorna o item na *posição* especificada de uma cadeia de caracteres resultante da avaliação de uma função PATH. As posições são contadas da esquerda para a direita.

Sintaxe

```
PATHITEM(<path>, <position>[, <type>])
```

Parâmetros

path

Uma cadeia de texto na forma dos resultados de uma função PATH.

position

Uma expressão de inteiro com a posição do item a ser retornado.

type

(Opcional) Uma enumeração que define o tipo de dados do resultado:

Enumeração	Enumeração alternativa	Descrição
TEXT	0	Os resultados são retornados com o tipo de dados de texto. (padrão).
INTEGER	1	Os resultados são retornados como inteiros.

Valor retornado

O identificador retornado pela função PATH na posição especificada na lista de identificadores. Os itens retornados pela função PATH são ordenados do mais distante para o atual.

Comentários

- Essa função pode ser usada para retornar um nível específico de uma hierarquia retornada por uma função PATH. Por exemplo, você pode retornar apenas os gerentes de salto de nível para todos os funcionários.
- Se você especificar um número para *position* que seja menor que um (1) ou maior que o número de elementos em *path*, a função PATHITEM retornará BLANK
- Se *type* não for um elemento de enumeração válido, um erro será retornado.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir retorna o gerente de terceira camada do funcionário atual. Ele usa as IDs de funcionário e de

gerente como a entrada para uma função PATH que retorna uma cadeia de caracteres com a hierarquia de pais até o funcionário atual. Com base nessa cadeia de caracteres, PATHITEM retorna a terceira entrada como um inteiro.

```
=PATHITEM(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

PATHITEMREVERSE

09/05/2020 • 3 minutes to read

Retorna o item na *posição* especificada de uma cadeia de caracteres resultante da avaliação de uma função PATH. As posições são contadas para trás, da direita para a esquerda.

Sintaxe

```
PATHITEMREVERSE(<path>, <position>[, <type>])
```

Parâmetros

path

Uma cadeia de texto resultante da avaliação de uma função PATH.

position

Uma expressão de inteiro com a posição do item a ser retornado. A posição é contada para trás da direita para a esquerda.

type

(Opcional) Uma enumeração que define o tipo de dados do resultado:

Enumeração	Enumeração alternativa	Descrição
TEXT	0	Os resultados são retornados com o tipo de dados de texto. (padrão)
INTEGER	1	Os resultados são retornados com o tipo de dados inteiro.

Valor retornado

O ascendente de n posições no caminho fornecido, contando do atual para o mais antigo.

Comentários

- Essa função pode ser usada para obter um item individual de uma hierarquia resultante de uma função PATH.
- Essa função reverte a ordem padrão da hierarquia, de modo que os itens mais próximos são listados primeiro. Por exemplo, se a função PATH retornar uma lista de gerentes acima de um funcionário em uma hierarquia, a função PATHITEMREVERSE retornará o gerente imediato do funcionário na posição 2, porque a posição 1 conterá a ID do funcionário.
- Se o número especificado para *position* for menor que um (1) ou maior que o número de elementos em *path*, a função PATHITEM retornará BLANK.
- Se *type* não for um elemento de enumeração válido, um erro será retornado.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre

limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir usa uma coluna de ID do funcionário como a entrada para uma função PATH e reverte a lista de elementos avô que são retornados. A posição especificada é 3 e o tipo de retorno é 1; Portanto, a função PATHITEMREVERSE retorna um inteiro que representa o gerente dois níveis acima do funcionário.

```
=PATHITEMREVERSE(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]), 3, 1)
```

PATHLENGTH

09/05/2020 • 2 minutes to read

Retorna o número de pais para o item especificado em um determinado resultado de PATH, incluindo self.

Sintaxe

```
PATHLENGTH(<path>)
```

Parâmetros

TERMO	DEFINIÇÃO
path	Uma expressão de texto resultante da avaliação de uma função PATH.

Valor retornado

O número de itens que são pais para o item especificado em um determinado resultado de PATH, incluindo o item especificado.

Comentários

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir usa uma ID de funcionário como entrada para uma função PATH e retorna uma lista dos gerentes acima desse funcionário na hierarquia, a função PATHLENGTH usa esse resultado e conta os diferentes níveis de funcionários e de gerentes, incluindo o funcionário com o qual você começou.

```
=PATHLENGTH(PATH(Employee[EmployeeKey], Employee[ParentEmployeeKey]))
```


Funções estatísticas

26/05/2020 • 9 minutes to read

O DAX (Data Analysis Expressions) fornece muitas funções para criar agregações como somas, contagens e médias. Essas funções são muito semelhantes às funções de agregação usadas pelo Microsoft Excel. Esta seção lista as funções estatísticas e de agregação fornecidas pelo DAX.

Nesta categoria

FUNÇÃO	DESCRIÇÃO
ADDCOLUMNS	Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.
APPROXIMATEDISTINCTCOUNT	Retorna o número <i>aproximado</i> de linhas que contêm valores distintos em uma coluna.
AVERAGE	Retorna a média aritmética de todos os números de uma coluna.
AVERAGEA	Retorna a média aritmética dos valores de uma coluna.
AVERAGEX	Calcula a média aritmética de um conjunto de expressões avaliadas de uma tabela.
BETA.DIST	Retorna a distribuição beta.
BETA.INV	Retorna o inverso da função de densidade de probabilidade cumulativa beta (BETA.DIST).
CHISQ.DIST	Retorna a distribuição qui-quadrada.
CHISQ.DIST. RT	Retorna a probabilidade de cauda direita da distribuição qui-quadrada.
CHISQ.INV	Retorna o inverso da probabilidade de cauda esquerda da distribuição qui-quadrada.
CHISQ.INV. RT	Retorna o inverso da probabilidade de cauda direita da distribuição qui-quadrada.
CONFIDENCE.NORM	O intervalo de confiança é um intervalo de valores.
CONFIDENCE.T	Retorna o intervalo de confiança para uma média populacional, usando uma distribuição t de Student.
COT	Retorna o cotangente de um ângulo especificado em radianos.
COTH	Retorna a cotangente hiperbólica de um ângulo hiperbólico.

FUNÇÃO	DESCRIÇÃO
COUNT	Conta o número de células de uma coluna que contém números.
COUNTA	Conta o número de células de uma coluna que não estão vazias.
COUNTAX	Conta os resultados sem valor em branco ao avaliar o resultado de uma expressão em uma tabela.
COUNTBLANK	Conta o número de células em branco em uma coluna.
COUNTROWS	Conta o número de linhas na tabela especificada ou em uma tabela definida por uma expressão.
COUNTX	Conta o número de linhas que contêm um número ou uma expressão que resulta em um número, ao avaliar uma expressão em uma tabela.
CROSSJOIN	Retorna uma tabela que contém o produto cartesiano de todas as linhas de todas as tabelas nos argumentos.
DATATABLE	Fornecer um mecanismo para declarar um conjunto embutido de valores de dados.
DISTINCTCOUNT	Conta o número de valores distintos de uma coluna.
DISTINCTCOUNTNOBLANK	Conta o número de valores distintos de uma coluna.
EXPON.DIST	Retorna a distribuição exponencial.
GENERATE	Retorna uma tabela com o produto cartesiano entre cada linha em <i>table1</i> e a tabela resultante da avaliação de <i>table2</i> no contexto da linha atual de <i>table1</i> .
GENERATEALL	Retorna uma tabela com o produto cartesiano entre cada linha em <i>table1</i> e a tabela resultante da avaliação de <i>table2</i> no contexto da linha atual de <i>table1</i> .
GEOMEAN	Retorna a média geométrica dos números em uma coluna.
GEOMEANX	Retorna a média geométrica de uma expressão avaliada para cada linha de uma tabela.
MAX	Retorna o maior valor numérico de uma coluna ou entre duas expressões escalares.
MAXA	Retorna o maior valor de uma coluna.
MAXX	Avalia uma expressão para cada linha de uma tabela e retorna o maior valor numérico.
MEDIAN	Retorna a mediana dos números de uma coluna.

FUNÇÃO	DESCRIÇÃO
MEDIANX	Retorna o número mediano de uma expressão avaliada para cada linha de uma tabela.
MIN	Retorna o menor valor numérico de uma coluna ou entre duas expressões escalares.
MINA	Retorna o menor valor de uma coluna, incluindo quaisquer valores lógicos e números representados como texto.
MINX	Retorna o menor valor numérico resultante da avaliação de uma expressão para cada linha de uma tabela.
NORM.DIST	Retorna a distribuição normal para a média especificada e o desvio padrão.
NORM.INV	O inverso da distribuição cumulativa normal para a média especificada e o desvio padrão.
NORM.S.DIST	Retorna a distribuição normal padrão (tem uma média igual a zero e um desvio padrão de um).
NORM.S.INV	Retorna o inverso da distribuição cumulativa normal padrão.
PERCENTILE.EXC	Retorna o k-ésimo percentil de valores em um intervalo, em que k está no intervalo de 0..1, exclusivo.
PERCENTILE.INC	Retorna o k-ésimo percentil de valores em um intervalo, em que k está no intervalo de 0..1, inclusivo.
PERCENTILEX.EXC	Retorna o número de percentil de uma expressão avaliada para cada linha de uma tabela.
PERCENTILEX.INC	Retorna o número de percentil de uma expressão avaliada para cada linha de uma tabela.
POISSON.DIST	Retorna a distribuição de Poisson.
RANK.EQ	Retorna a classificação de um número em uma lista de números.
RANKX	Retorna a classificação de um número em uma lista de números para cada linha no argumento <i>table</i> .
ROW	Retorna uma tabela com uma única linha contendo valores que resultam das expressões fornecidas para cada coluna.
SAMPLE	Retorna uma amostra de N linhas da tabela especificada.
SELECTCOLUMNS	Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.
SIN	Retorna o seno do ângulo determinado.

FUNÇÃO	DESCRIÇÃO
SINH	Retorna o seno hiperbólico de um número.
STDEV.P	Retorna o desvio padrão da população inteira.
STDEV.S	Retorna o desvio padrão de uma amostra de população.
STDEVX.P	Retorna o desvio padrão da população inteira.
STDEVX.S	Retorna o desvio padrão de uma amostra de população.
SQRTPI	Retorna a raiz quadrada de (número * pi).
SUMMARIZE	Retorna uma tabela de resumo para os totais solicitados sobre um conjunto de grupos.
T.DIST	Retorna a distribuição t caudal esquerda do Student.
T.DIST.2T	Retorna a distribuição t bicaudal do Student.
T.DIST.RT	Retorna a distribuição t de cauda direita do Student.
T.INV	Retorna o inverso da distribuição t de cauda esquerda do Student.
T.INV.2t	Retorna o inverso bicaudal da distribuição t do Student.
TAN	Retorna a tangente do ângulo determinado.
TANH	Retorna a tangente hiperbólica de um número.
TOPN	Retorna as N linhas superiores da tabela especificada.
VAR.P	Retorna a variância da população inteira.
VAR.S	Retorna a variância de uma amostra da população.
VARX.P	Retorna a variância da população inteira.
VARX.S	Retorna a variância de uma amostra da população.
XIRR	Retorna a taxa interna de retorno de um agendamento de fluxos de caixa que não é necessariamente periódico.
XNPV	Retorna o valor atual de um agendamento de fluxos de caixa que não é necessariamente periódico.

ADDCOLUMNS

09/05/2020 • 2 minutes to read

Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.

Sintaxe

```
ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
Nome	O nome dado à coluna, entre aspas duplas.
expressão	Qualquer expressão DAX que retorne uma expressão escalar, avaliada para cada linha da <i>tabela</i> .

Valor retornado

Uma tabela com todas as suas colunas originais e as adicionadas.

Comentários

Exemplo

O exemplo a seguir retorna uma versão estendida da tabela de Categoria de produto, que inclui os valores totais de vendas do canal revendedor e das vendas pela Internet.

```
ADDCOLUMNS(ProductCategory,
    , "Internet Sales", SUMX(RELATEDTABLE(InternetSales_USD), InternetSales_USD[SalesAmount_USD])
    , "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

A tabela a seguir mostra uma visualização dos dados conforme eles seriam recebidos por qualquer função que espera receber uma tabela:

ProductCategory[ProductCategoryName]	ProductCategory[ProductCategoryAlternateKey]	ProductCategory[ProductCategoryKey]	[Internet Sales]	[Vendas do Revendedor]
Bicicletas	1	1	25107749,77	63084675,04
Componentes	2	2		11205837,96
Vestuário	3	3	306157,5829	1669943,267

Acessórios	4	4	640920,1338	534301,9888

APPROXIMATEDISTINCTCOUNT

08/05/2020 • 2 minutes to read

Retorna o número *aproximado* de linhas que contêm valores distintos em uma coluna. Essa função pode consultar grandes quantidades de dados com um desempenho potencialmente melhor do que DISTINCTCOUNT, com um pequeno desvio do resultado exato.

Sintaxe

```
APPROXIMATEDISTINCTCOUNT(<columnName>)
```

Parâmetros

TERMO	DESCRIÇÃO
coluna	A coluna que contém os valores a serem contados. Não pode ser uma expressão.

Valor retornado

O número aproximado de valores distintos em *coluna*.

Comentários

O único argumento para essa função é uma coluna. Você pode usar colunas que contenham qualquer tipo de dados. Quando a função não encontra nenhuma linha para contagem, ela retorna um espaço em branco; caso contrário, retorna a contagem de valores distintos.

Essa função é uma **versão prévia**. As seguintes limitações se aplicam:

- Essa função atualmente dá suporte a conexões **DirectQuery** somente para as seguintes fontes de dados: Banco de Dados SQL do Azure e SQL Data Warehouse do Azure.
- Este recurso ainda não está disponível no IntelliSense.

AVERAGE

09/05/2020 • 3 minutes to read

Retorna a média aritmética de todos os números de uma coluna.

Sintaxe

```
AVERAGE(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números para os quais você deseja a média.

Valor retornado

Retorna um número decimal que representa a média aritmética dos números na coluna.

Comentários

Essa função usa a coluna especificada como um argumento e localiza a média dos valores nessa coluna. Se você quiser localizar a média de uma expressão avaliada como um conjunto de números, use a função AVERAGEX em vez disso.

Os valores não numéricos na coluna são tratados da seguinte maneira:

- Se a coluna contiver texto, nenhuma agregação poderá ser executada e as funções retornarão espaços em branco.
- Se a coluna contiver valores lógicos ou células vazias, esses valores serão ignorados.
- As células com o valor zero são incluídas.
- Ao calcular a média de células, você deve ter em mente a diferença entre uma célula vazia e uma célula que contém o valor 0 (zero). Quando uma célula contém 0, ela é adicionada à soma dos números e a linha é contada entre o número de linhas usadas como o divisor. No entanto, quando uma célula contém um espaço em branco, a linha não é contada.

Sempre que não houver linhas para agregação, a função retornará um valor em branco. No entanto, se houver linhas, mas nenhuma delas atender aos critérios especificados, a função retornará 0. O Excel também retornará um zero se não for encontrada nenhuma linha que atenda às condições.

Exemplo

A fórmula a seguir retorna a média dos valores na coluna, ExtendedSalesAmount, na tabela, InternetSales.

```
=AVERAGE(InternetSales[ExtendedSalesAmount])
```


Funções relacionadas

A função AVERAGEX pode usar como argumento uma expressão avaliada para cada linha em uma tabela. Isso permite que você execute cálculos e, em seguida, use a média dos valores calculados.

A função AVERAGEA usa uma coluna como argumento, mas, caso contrário, é como a função do Excel de mesmo nome. Usando a função AVERAGEA, você pode calcular uma média em uma coluna que contém valores vazios.

AVERAGEA

08/05/2020 • 2 minutes to read

Retorna a média aritmética dos valores de uma coluna. Manipula texto e valores não numéricos.

Sintaxe

```
AVERAGEA(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os valores para os quais você deseja a média.

Valor retornado

Um número decimal.

Comentários

A função AVERAGEA usa uma coluna e calcula a média dos números nela, mas também trata os tipos de dados não numéricos de acordo com as seguintes regras:

- Valores avaliados como TRUE Count como 1.
- Valores avaliados como FALSE contam como 0 (zero).
- Valores que contêm texto não numérico contam como 0 (zero).
- Texto vazio ("") contam como 0 (zero).

Se você não quiser incluir valores lógicos e representações de texto de números em uma referência como parte do cálculo, use a função AVERAGE.

Sempre que não houver linhas para agregação, a função retornará um valor em branco. No entanto, se houver linhas, mas nenhuma delas atender aos critérios especificados, a função retornará 0. O Microsoft Excel também retornará um zero se não for encontrada nenhuma linha que atenda às condições.

Exemplo

O exemplo a seguir retorna a média de células não vazias na coluna referenciada, dada a tabela a seguir. Se você tivesse usado a função AVERAGE, a média seria 21/2; com a função AVERAGEA, o resultado seria 22/5.

ID DA TRANSAÇÃO	QUANTIDADE	RESULTADO
0000123	1	Conta como 1
0000124	20	Conta como 20

ID DA TRANSAÇÃO	QUANTIDADE	RESULTADO
0000125	n/a	Conta como 0
0000126		Conta como 0
0000126	VERDADEIRO	Conta como 1

=AVERAGEA([Amount])

Consulte também

[Função AVERAGE \(DAX\)](#)

[Função AVERAGEX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

AVERAGEX

09/05/2020 • 2 minutes to read

Calcula a média aritmética de um conjunto de expressões avaliadas de uma tabela.

Sintaxe

```
AVERAGEX(<table>,<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Nome de uma tabela ou uma expressão que especifica a tabela sobre a qual a agregação pode ser executada.
expressão	Uma expressão com um resultado escalar, que será avaliado para cada linha da tabela no primeiro argumento.

Valor retornado

Um número decimal.

Comentários

A função AVERAGEX permite avaliar expressões para cada linha de uma tabela e, em seguida, pegar o conjunto de valores resultante e calcular sua média aritmética. Portanto, a função usa uma tabela como o primeiro argumento e uma expressão como o segundo argumento.

Em todos os outros aspectos, AVERAGEX segue as mesmas regras que a AVERAGE. Você não pode incluir células não numéricas ou nulas. Os argumentos de tabela e de expressão são obrigatórios.

Quando não houver linhas para agregação, a função retornará um valor em branco. Quando houver linhas, mas nenhuma delas atender aos critérios especificados, a função retornará 0.

Exemplo

O exemplo a seguir calcula o frete médio e o imposto em cada pedido na tabela InternetSales, somando primeiro Frete mais TaxAmt em cada linha e, em seguida, calculando a média dessas somas.

```
=AVERAGEX(InternetSales, InternetSales[Freight]+ InternetSales[TaxAmt])
```

Se você usar várias operações na expressão usada como o segundo argumento, deverá usar parênteses para controlar a ordem dos cálculos. Para obter mais informações, confira [Referência de Sintaxe DAX](#).

Consulte também

[Função AVERAGE \(DAX\)](#)

[Função AVERAGEA \(DAX\)](#)

BETA.DIST

08/05/2020 • 2 minutes to read

Retorna a distribuição beta. A distribuição beta geralmente é usada para estudar o percentual de algo usando amostras, como a fração diária que as pessoas passam assistindo televisão.

Sintaxe

```
BETA.DIST(x,alpha,beta,cumulative,[A],[B])
```

Parâmetros

TERMO	DEFINIÇÃO
x	O valor entre A e B pelo qual avaliar a função
Alfa	Um parâmetro da distribuição.
Beta	Um parâmetro da distribuição.
A	Opcional. Um limite inferior para o intervalo de x.
B	Opcional. Um limite superior para o intervalo de x.

Valor retornado

Retorna a distribuição beta.

Comentários

Se algum argumento for não numérico, BETA.DIST retornará o valor de erro #VALUE!.

Se qualquer argumento não for um inteiro, ele será arredondado.

Se $\alpha \leq 0$ ou $\beta \leq 0$, BETA.DIST retornará o valor de erro #VALUE!.

Se $x < A$, $x > B$ ou $A = B$, BETA.DIST retornará o valor de erro #VALUE!.

Se você omitir valores para A e B, BETA.DIST usará a distribuição beta cumulativa padrão, de modo que $A = 0$ e $B = 1$.

BETA.INV

08/05/2020 • 2 minutes to read

Retorna o inverso da função de densidade de probabilidade cumulativa beta (BETA.DIST).

Se $\text{probability} = \text{BETA.DIST}(x, \dots, \text{TRUE})$, then $\text{BETA.INV}(\text{probability}, \dots) = x$. A distribuição beta pode ser usada no planejamento do projeto para modelar tempos de conclusão prováveis dado um tempo de conclusão e uma variabilidade esperados.

Sintaxe

`BETA.INV(probability,alpha,beta,[A],[B])`

Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	Uma probabilidade associada à distribuição de beta.
Alfa	Um parâmetro da distribuição.
Beta	Um parâmetro da distribuição.
A	Opcional. Um limite inferior para o intervalo de x.
B	Opcional. Um limite superior para o intervalo de x.

Valor retornado

Retorna o inverso da função de densidade de probabilidade cumulativa beta (BETA.DIST).

Comentários

Se algum argumento for não numérico, BETA.INV retornará o valor de erro #VALUE!.

Se qualquer argumento não for um inteiro, ele será arredondado.

Se $\alpha \leq 0$ ou $\beta \leq 0$, BETA.INV retornará o valor de erro #VALUE!.

Se $\text{probabilidade} \leq 0$ ou $\text{probabilidade} > 1$, BETA.INV retornará o valor de erro #VALUE!.

Se você omitir valores para A e B, BETA.INV usará a distribuição beta cumulativa padrão, de modo que $A = 0$ e $B = 1$.

CHISQ.DIST

15/05/2020 • 2 minutes to read

Retorna a distribuição qui-quadrada.

Ela é geralmente usada para estudar o percentual de algo usando amostras, como a fração diária que as pessoas passam assistindo televisão.

Sintaxe

```
CHISQ.DIST(<x>, <deg_freedom>, <cumulative>)
```

Parâmetros

TERMO	DEFINIÇÃO
x	O valor no qual você deseja avaliar a distribuição.
Deg_freedom	O número de graus de liberdade.
cumulative	Um valor lógico que determina a forma da função. Se cumulative for TRUE, CHISQ.DIST retornará a função de distribuição cumulativa; se for FALSE, retornará a função de densidade de probabilidade.

Valor retornado

A distribuição qui-quadrada.

Comentários

Se x ou deg_freedom não for numérico, um erro será retornado.

Se deg_freedom não for um inteiro, ele será arredondado.

Se $x < 0$, um erro é retornado.

Se $\text{deg_freedom} < 1$ ou $\text{deg_freedom} > 10^{10}$, um erro é retornado.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { CHISQ.DIST(2, 2, TRUE) }
```

Retorna

[VALOR]

0.632120558828558

CHISQ.DIST.RT

15/05/2020 • 2 minutes to read

Retorna a probabilidade de cauda direita da distribuição qui-quadrada.

A distribuição qui-quadrada é associada a um teste qui-quadrado. Use o teste qui-quadrado para comparar os valores observados e esperados. Por exemplo, um experimento genético pode criar a hipótese de que a próxima geração de plantas exibirá um determinado conjunto de cores. Ao comparar os resultados observados com os esperados, você pode decidir se a hipótese original é válida.

Sintaxe

```
CHISQ.DIST.RT(<x>, <deg_freedom>)
```

Parâmetros

TERMO	DEFINIÇÃO
x	O valor no qual você deseja avaliar a distribuição.
Deg_freedom	O número de graus de liberdade.

Valor retornado

A probabilidade de cauda direita da distribuição qui-quadrada.

Comentários

Se x ou deg_freedom não for numérico, um erro será retornado.

Se deg_freedom não for um inteiro, ele será arredondado.

Se $x < 0$, um erro é retornado.

Se $\text{deg_freedom} < 1$ ou $\text{deg_freedom} > 10^{10}$, um erro é retornado.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { CHISQ.DIST.RT(2, 5) }
```

Retorna

[VALOR]

0.84914503608461

Retorna o inverso da probabilidade de cauda esquerda da distribuição qui-quadrada.

Ela é geralmente usada para estudar o percentual de algo usando amostras, como a fração diária que as pessoas passam assistindo televisão.

Sintaxe

```
CHISQ.INV(probability,deg_freedom)
```

Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	Uma probabilidade associada à distribuição qui-quadrada.
Deg_freedom	O número de graus de liberdade.

Valor retornado

Retorna o inverso da probabilidade de cauda esquerda da distribuição qui-quadrada.

Comentários

Se o argumento for não numérico, CHISQ.INV retornará o valor de erro #VALUE!.

Se probabilidade < 0 ou probabilidade > 1, CHISQ.INV retornará o valor de erro #VALUE!.

Se deg_freedom não for um inteiro, ele será arredondado.

Se deg_freedom < 1 ou deg_freedom > 10¹⁰, CHISQ.INV retornará o valor de erro #VALUE!.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=CHISQ.INV(0.93,1)	Inverso da probabilidade de cauda esquerda da distribuição qui-quadrada para 0,93, usando 1 grau de liberdade.	5,318520074
=CHISQ.INV(0.6,2)	Inverso da probabilidade de cauda esquerda da distribuição qui-quadrada para 0,6, usando 2 graus de liberdade.	1,832581464

CHISQ.INV.RT

08/05/2020 • 2 minutes to read

Retorna o inverso da probabilidade de cauda direita da distribuição qui-quadrada.

Se $\text{probability} = \text{CHISQ.DIST.RT}(x, \dots)$, $\text{CHISQ.INV.RT}(\text{probability}, \dots) = x$. Use essa função para comparar os resultados observados com os esperados para decidir se a hipótese original é válida.

Sintaxe

```
CHISQ.INV.RT(probability,deg_freedom)
```

Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	Uma probabilidade associada à distribuição qui-quadrada.
Deg_freedom	O número de graus de liberdade.

Valor retornado

Retorna o inverso da probabilidade de cauda direita da distribuição qui-quadrada.

Comentários

Se um dos argumentos for não numérico, CHISQ.INV.RT retornará o valor de erro #VALUE!.

Se probabilidade < 0 ou probabilidade > 1, CHISQ.INV.RT retornará o valor de erro #VALUE!.

Se deg_freedom não for um inteiro, ele será arredondado.

Se deg_freedom < 1, CHISQ.INV.RT retornará o valor de erro #VALUE!.

Dado um valor para probabilidade, CHISQ.INV.RT busca esse valor x como $\text{CHISQ.DIST.RT}(x, \text{deg_freedom}) = \text{probabilidade}$. Portanto, a precisão de CHISQ.INV.RT depende da precisão de CHISQ.DIST.RT. CHISQ.INV.RT usa uma técnica de pesquisa iterativa. Se a pesquisa não tiver convergido após 64 iterações, a função retornará o valor de erro #N/A.

CONFIDENCE.NORM

08/05/2020 • 3 minutes to read

O intervalo de confiança é um intervalo de valores. A média amostral, x , está no centro desse intervalo, que é $x \pm \text{CONFIDENCE.NORM}$. Por exemplo, se x é a média amostral dos tempos de entrega para os produtos solicitados pelo correio, $x \pm \text{CONFIDENCE.NORM}$ é um intervalo de médias populacionais. Para qualquer média populacional, μ_0 , nesse intervalo, a probabilidade de obter uma média amostral mais distante de μ_0 que x é maior que α ; para qualquer média populacional, μ_0 , que não esteja nesse intervalo, a probabilidade de obter uma média amostral mais distante de μ_0 que x é menor que α . Em outras palavras, suponha que usemos x , standard_dev e size para construir um teste bicaudal no nível de significância α da hipótese de que a média populacional é μ_0 . Em seguida, não rejeitaremos essa hipótese se μ_0 estiver no intervalo de confiança e rejeitaremos essa hipótese se μ_0 não estiver no intervalo de confiança. O intervalo de confiança não nos permite inferir que há a probabilidade $1 - \alpha$ de que nosso próximo pacote levará um tempo de entrega que esteja no intervalo de confiança.

Sintaxe

```
CONFIDENCE.NORM(alpha, standard_dev, size)
```

Parâmetros

TERMO	DEFINIÇÃO
alpha	O nível de significância usado para calcular o nível de confiança. O nível de confiança é igual a $100 \times (1 - \alpha)\%$ ou, em outras palavras, um α igual a 0,05 indica um nível de confiança de 95%.
standard_dev	O desvio padrão populacional do intervalo de dados. Pressupõe-se que ele seja conhecido.
standard_dev,size	O tamanho da amostra.

Valor retornado

Um intervalo de valores

Comentários

Se qualquer argumento não for numérico, CONFIDENCE.NORM retornará o valor de erro #VALUE!.

Se $\alpha \leq 0$ ou $\alpha \geq 1$, CONFIDENCE.NORM retornará o valor de erro #VALUE!.

Se $\text{standard_dev} \leq 0$, CONFIDENCE.NORM retornará o valor de erro #VALUE!.

Se size não for um inteiro, ele será arredondado.

Se $\text{size} < 1$, CONFIDENCE.NORM retornará o valor de erro #VALUE!.

Se pressupormos que α seja igual a 0,05, precisaremos calcular a área sob a curva normal padrão que é igual a $(1 - \alpha)$ ou 95%. Esse valor é $\pm 1,96$. Portanto, o intervalo de confiança é:

$$\bar{x} \pm 1.96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

CONFIDENCE.T

08/05/2020 • 2 minutes to read

Retorna o intervalo de confiança para uma média populacional, usando uma distribuição t de Student.

Sintaxe

```
CONFIDENCE.T(alpha,standard_dev,size)
```

Parâmetros

TERMO	DEFINIÇÃO
alpha	O nível de significância usado para calcular o nível de confiança. O nível de confiança é igual a $100 \times (1 - \alpha)\%$ ou, em outras palavras, um alfa igual a 0,05 indica um nível de confiança de 95%.
standard_dev	O desvio padrão populacional do intervalo de dados. Pressupõe-se que ele seja conhecido.
size	O tamanho da amostra.

Valor retornado

Retorna o intervalo de confiança para uma média populacional, usando uma distribuição t de Student.

Comentários

Se qualquer argumento for não numérico, CONFIDENCE.T retornará o valor de erro #VALUE!.

Se $\alpha \leq 0$ ou $\alpha \geq 1$, CONFIDENCE.T retornará o valor de erro #VALUE!.

Se $\text{standard_dev} \leq 0$, CONFIDENCE.T retornará o valor de erro #VALUE!.

Se size não for um inteiro, ele será arredondado.

Se size for igual a 1, CONFIDENCE.T retornará o valor de erro #DIV/0!.

Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=CONFIDENCE.T(0,05,1,50)	Intervalo de confiança para a média de uma população com base em um tamanho de amostra igual a 50, com um nível de significância de 5% e um desvio padrão igual a 1. Isso se baseia em uma distribuição t de Student.	0,284196855

COT

15/05/2020 • 2 minutes to read

Retorna a cotangente de um ângulo especificado em radianos.

Sintaxe

```
COT (<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O ângulo em radianos para o qual você quer a cotangente.

Valor retornado

A cotangente do ângulo determinado.

Comentários

O valor absoluto do número deve ser menor que 2^{27} e não pode ser 0.

Se o número estiver fora das restrições, um erro será retornado.

Se o número for um valor não numérico, um erro será retornado.

Exemplo

A seguinte consulta DAX:

```
EVALUATE { COT(30) }
```

Retorna

[VALOR]

-0.156119952161659

COTH

15/05/2020 • 2 minutes to read

Retorna a cotangente hiperbólica de um ângulo hiperbólico.

Sintaxe

```
COTH (<number>)
```

Parâmetros

TERMO	DEFINIÇÃO
número	O ângulo hiperbólico em radianos para o qual você deseja a cotangente hiperbólica.

Valor retornado

A cotangente hiperbólica do ângulo fornecido.

Comentários

A cotangente hiperbólica é uma analogia da cotangente comum (circular).

O valor absoluto do número deve ser menor que 2^27 e não pode ser 0.

Se o número estiver fora das restrições, um erro será retornado

Se o número for um valor não numérico, um erro será retornado.

A equação a seguir é usada:

$$\coth(N) = \frac{1}{\tanh(N)} = \frac{\cosh(N)}{\sinh(N)} = \frac{e^N + e^{-N}}{e^N - e^{-N}}$$

Exemplo

A seguinte consulta DAX:

```
EVALUATE { COTH(2) }
```

Retorna

[VALOR]

1.03731472072755

CONTAGEM

08/05/2020 • 2 minutes to read

A função COUNT conta o número de células de uma coluna que contém valores que não estão em branco.

Sintaxe

```
COUNT(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os valores a serem contados.

Valor retornado

Um número inteiro.

Comentários

O único argumento permitido para essa função é uma coluna. A função COUNT conta as linhas que contêm os seguintes tipos de valores:

- Números
- Datas
- Cadeias de caracteres

Quando a função não encontra nenhuma linha para contagem, ela retorna um valor em branco.

Os valores em branco são ignorados. Não há suporte para valores TRUE/FALSE.

Caso você deseje avaliar uma coluna de valores TRUE/FALSE, use a função COUNTA.

Exemplo

O exemplo a seguir mostra como contar o número de valores na coluna ShipDate.

```
=COUNT([ShipDate])
```

Para contar valores lógicos ou texto, use as funções COUNTA ou COUNTAX.

Consulte também

[Função COUNTA \(DAX\)](#)

[Função COUNTAX \(DAX\)](#)

[Função COUNTX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

COUNTA

09/05/2020 • 2 minutes to read

A função COUNTA conta o número de células de uma coluna que não estão vazias.

Sintaxe

```
COUNTA(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os valores a serem contados

Valor retornado

Um número inteiro.

Comentários

Quando a função não encontra nenhuma linha para contagem, a função retorna um valor em branco.

Exemplo

O exemplo a seguir retorna todas as linhas da tabela `Reseller` que têm qualquer tipo de valor na coluna que armazena números de telefone. Como o nome da tabela não contém espaços, as aspas são opcionais.

```
=COUNTA('Reseller'[Phone])
```

Consulte também

[Função COUNT \(DAX\)](#)

[Função COUNTAX \(DAX\)](#)

[Função COUNTX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

COUNTAX

08/05/2020 • 2 minutes to read

A função COUNTAX conta os resultados sem valor em branco ao avaliar o resultado de uma expressão em uma tabela. Ou seja, ela funciona exatamente como a função COUNTA, mas é usada para iterar nas linhas de uma tabela e contar as linhas em que as expressões especificadas geram um resultado sem valor em branco.

Sintaxe

```
COUNTAX(<table>,<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

Valor retornado

Um número inteiro.

Comentários

Assim como a função COUNTA, a função COUNTAX conta as células que contêm qualquer tipo de informação, incluindo outras expressões.

Por exemplo, se a coluna contiver uma expressão que seja avaliada como uma cadeia de caracteres vazia, a função COUNTAX tratará o resultado como sem valor em branco. Normalmente, a função COUNTAX não conta células vazias, mas, nesse caso, a célula contém uma fórmula, portanto, ela é contada.

Sempre que a função não encontrar nenhuma linha para agregação, ela retornará um valor em branco.

Exemplo

O exemplo a seguir conta o número de linhas que não estão em branco na coluna Phone usando a tabela resultante da filtragem da tabela Reseller em [Status] = **Active**.

```
=COUNTAX(FILTER('Reseller',[Status]="Active"),[Phone])
```

Consulte também

[Função COUNT \(DAX\)](#)

[Função COUNTA \(DAX\)](#)

[Função COUNTX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

COUNTBLANK

08/05/2020 • 2 minutes to read

Conta o número de células em branco em uma coluna.

Sintaxe

```
COUNTBLANK(<column>)
```

Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém as células em branco a serem contadas.

Valor retornado

Um número inteiro. Se não for encontrada nenhuma linha que atenda à condição, serão retornados valores em branco.

Comentários

O único argumento permitido para essa função é uma coluna. Você pode usar colunas que contêm qualquer tipo de dados, mas apenas células em branco são contadas. As células que têm o valor zero (0) não são contadas, pois zero é considerado um valor numérico e não um valor em branco.

Sempre que não houver linhas para agregação, a função retornará um valor em branco. No entanto, se houver linhas, mas nenhuma delas atender aos critérios especificados, a função retornará 0. O Microsoft Excel também retornará um zero se não for encontrada nenhuma linha que atenda às condições.

Em outras palavras, se a função COUNTBLANK não encontrar nenhum valor em branco, o resultado será zero, mas se não houver linhas a serem verificadas, o resultado será um valor em branco.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

Exemplo

O exemplo a seguir mostra como contar o número de linhas da tabela Reseller que têm valores em branco para BankName.

```
=COUNTBLANK(Reseller[BankName])
```

Para contar valores lógicos ou texto, use as funções COUNTA ou COUNTAX.

Consulte também

[Função COUNT \(DAX\)](#)

[Função COUNTA \(DAX\)](#)

Função COUNTAX (DAX)
Função COUNTX (DAX)
Funções estatísticas (DAX)

COUNTROWS

08/05/2020 • 2 minutes to read

A função COUNTROWS conta o número de linhas na tabela especificada ou em uma tabela definida por uma expressão.

Sintaxe

```
COUNTROWS(<table>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	O nome da tabela que contém as linhas a serem contadas ou uma expressão que retorna uma tabela.

Valor retornado

Um número inteiro.

Comentários

Essa função pode ser usada para contar o número de linhas em uma tabela base, mas com mais frequência é usada para contar o número de linhas que resultam da filtragem de uma tabela ou da aplicação de contexto a uma tabela.

Sempre que não houver linhas para agregação, a função retornará um valor em branco. No entanto, se houver linhas, mas nenhuma delas atender aos critérios especificados, a função retornará 0. O Microsoft Excel também retornará um zero se não for encontrada nenhuma linha que atenda às condições.

Exemplo

O exemplo a seguir mostra como contar o número de linhas da tabela Orders. O resultado esperado é 52761.

```
=COUNTROWS('Orders')
```

Exemplo

O exemplo a seguir demonstra como usar COUNTROWS com um contexto de linha. Nesse cenário, há dois conjuntos de dados relacionados por número de pedido. A tabela Reseller contém uma linha para cada revendedor; a tabela ResellerSales contém várias linhas para cada pedido, cada linha contendo um pedido para um revendedor específico. As tabelas são conectadas por uma relação na coluna ResellerKey.

A fórmula obtém o valor de ResellerKey e, em seguida, conta o número de linhas da tabela relacionada que têm a mesma ID de revendedor. O resultado é a saída na coluna, **CalculatedColumn1**.

```
=COUNTROWS(RELATEDTABLE(ResellerSales))
```

A seguinte tabela mostra uma parte dos resultados esperados:

RESELLERKEY	CALCULATEDCOLUMN1
1	73
2	70
3	394

Consulte também

- [Função COUNT \(DAX\)](#)
- [Função COUNTA \(DAX\)](#)
- [Função COUNTAX \(DAX\)](#)
- [Função COUNTX \(DAX\)](#)
- [Funções estatísticas \(DAX\)](#)

COUNTX

08/05/2020 • 2 minutes to read

Conta o número de linhas que contêm um valor que não esteja em branco ou uma expressão que é avaliada como um valor que não esteja em branco, ao avaliar uma expressão em uma tabela.

Sintaxe

```
COUNTX(<table>,<expression>)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas a serem contadas.
expressão	Uma expressão que retorna o conjunto de valores que contém os valores que você deseja contar.

Valor retornado

Um inteiro.

Comentários

A função COUNTX usa dois argumentos. O primeiro argumento sempre precisa ser uma tabela ou qualquer expressão que retorne uma tabela. O segundo argumento é a coluna ou a expressão pesquisada por COUNTX.

A função COUNTX conta apenas valores, datas ou cadeias de caracteres. Se a função não encontrar nenhuma linha para contagem, ela retornará um valor em branco.

Se você desejar contar valores lógicos, use a função COUNTAX.

Exemplo 1

A fórmula a seguir retorna uma contagem de todas as linhas da tabela Product que têm um preço de lista.

```
=COUNTX(Product,[ListPrice])
```

Exemplo 2

A fórmula a seguir ilustra como passar uma tabela filtrada para COUNTX para o primeiro argumento. A fórmula usa uma expressão de filtro para obter apenas as linhas da tabela Product que atendem à condição, ProductSubCategory = "Caps" e, em seguida, conta as linhas na tabela resultante que têm um preço de lista. A expressão FILTER se aplica à tabela Products, mas usa um valor que você pesquisa na tabela relacionada, ProductSubCategory.

```
=COUNTX(FILTER(Product,RELATED(ProductSubcategory[EnglishProductSubcategoryName])="Caps", Product[ListPrice])
```


Consulte também

[Função COUNT \(DAX\)](#)

[Função COUNTA \(DAX\)](#)

[Função COUNTAX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

CROSSJOIN

08/05/2020 • 2 minutes to read

Retorna uma tabela que contém o produto cartesiano de todas as linhas de todas as tabelas nos argumentos. As colunas na nova tabela são todas as colunas em todas as tabelas de argumentos.

Sintaxe

```
CROSSJOIN(<table>, <table>[, <table>]...)
```

Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados

Valor retornado

Uma tabela que contém o produto cartesiano de todas as linhas de todas as tabelas nos argumentos.

Comentários

- Os nomes de coluna dos argumentos de *table* precisam ser diferentes em todas as tabelas ou um erro é retornado.
- O número total de linhas retornadas por CROSSJOIN() é igual ao produto do número de linhas de todas as tabelas nos argumentos; além disso, o número total de colunas na tabela de resultados é a soma do número de colunas em todas as tabelas. Por exemplo, se **TableA** tiver as linhas **rA** e as colunas **cA**, **TableB** tiver as linhas **rB** e as colunas **cB** e **TableC** tiver as linhas **rC** e a coluna **cC**, a tabela resultante terá as linhas **rA × rB × rC** e as colunas **cA + cB + cC**.

Exemplo

O seguinte exemplo mostra os resultados da aplicação de CROSSJOIN() a duas tabelas: **Cores** e **Papel de carta**.

A tabela **Cores** contém cores e padrões:

COR	PADRÃO
Vermelho	Faixa horizontal
Verde	Faixa vertical
Azul	Hachura

A tabela **Papel de carta** contém fontes e apresentação:

FONTE	APRESENTAÇÃO
com serifa	alto-relevo
sem serifa	baixo-relevo

A expressão usada para gerar a união cruzada é apresentada abaixo:

```
CROSSJOIN( Colors, Stationery)
```

Quando a expressão acima é usada sempre que uma expressão de tabela é esperada, os resultados da expressão são os seguintes:

COR	PADRÃO	FONTE	APRESENTAÇÃO
Vermelho	Faixa horizontal	com serifa	alto-relevo
Verde	Faixa vertical	com serifa	alto-relevo
Azul	Hachura	com serifa	alto-relevo
Vermelho	Faixa horizontal	sem serifa	baixo-relevo
Verde	Faixa vertical	sem serifa	baixo-relevo
Azul	Hachura	sem serifa	baixo-relevo

DISTINCTCOUNT

08/05/2020 • 2 minutes to read

Conta o número de valores distintos de uma coluna.

Sintaxe

```
DISTINCTCOUNT(<column>)
```

Parâmetros

TERMO	DESCRIÇÃO
coluna	A coluna que contém os valores a serem contados

Valor retornado

O número de valores distintos na *coluna*.

Comentários

O único argumento permitido para essa função é uma coluna. Você pode usar colunas que contenham qualquer tipo de dados. Quando a função não encontra nenhuma linha para contagem, ela retorna um espaço em branco; caso contrário, retorna a contagem de valores distintos.

A função DISTINCTCOUNT inclui o valor BLANK. Para ignorar o valor BLANK, use a função [DISTINCTCOUNTNOBLANK](#).

Exemplo

O exemplo a seguir mostra como contar o número de pedidos de vendas distintos na coluna ResellerSales_USD[SalesOrderNumber].

```
=DISTINCTCOUNT(ResellerSales_USD[SalesOrderNumber])
```

```
```dax
```

Using the above measure in a table with calendar year in the side and product category on top gives the following results:

```

|**Distinct Reseller Orders count**|**Column Labels**| | | | | |
|---|---|---|---|---|---|---|
|**Row Labels**|**Accessories**|**Bikes**|**Clothing**|**Components**|**Grand Total**|
|2005|135|345|242|205||366|
|2006|356|850|644|702||1015|
|2007|531|1234|963|1138||1521|
|2008|293|724|561|601||894|
|1|1|1|1|1|1|
|**Grand Total**|**1315**|**3153**|**2410**|**2646**|**1**|**3797**|
```

In the above example, note that the rows Grand Total numbers do not add up, this happens because the same order might contain line items, in the same order, from different product categories.

## See also

[COUNT function &#40;DAX&#41;](count-function-dax.md)

[COUNTA function &#40;DAX&#41;](counta-function-dax.md)

[COUNTAX function &#40;DAX&#41;](countax-function-dax.md)

[COUNTX function &#40;DAX&#41;](countx-function-dax.md)

[Statistical functions &#40;DAX&#41;](statistical-functions-dax.md)

# EXPON.DIST

09/05/2020 • 2 minutes to read

Retorna a distribuição exponencial. Use EXPON.DIST para modelar o tempo entre os eventos, como quanto tempo um caixa eletrônico automatizado leva para entregar o dinheiro. Por exemplo, você pode usar EXPON.DIST para determinar a probabilidade de o processo levar no máximo um minuto.

## Sintaxe

EXPON.DIST(x,lambda,cumulative)

### Parâmetros

TERMO	DEFINIÇÃO
x	Obrigatório. O valor da função.
lambda	Obrigatório. O valor do parâmetro.
cumulative	Obrigatório. Um valor lógico que indica qual forma da função exponencial deve ser fornecida. Se o cumulativo for TRUE, EXPON.DIST retornará a função de distribuição cumulativa; se for FALSE, retornará a função de densidade de probabilidade.

## Valor retornado

Retorna a distribuição exponencial.

## Comentários

Se x ou lambda não for numérico, EXPON.DIST retornará o valor de erro #VALUE!.

Se x ou lambda não for um inteiro, ele será arredondado.

Se  $x < 0$ , EXPON.DIST retornará o valor de erro #VALUE!.

Se  $\lambda \leq 0$ , EXPON.DIST retornará o valor de erro #VALUE!.

A equação para a função de densidade de probabilidade é:

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

A equação para a função de distribuição cumulativa é:

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

# GENERATE

09/05/2020 • 3 minutes to read

Retorna uma tabela com o produto cartesiano entre cada linha em *table1* e a tabela resultante da avaliação de *table2* no contexto da linha atual de *table1*.

## Sintaxe

```
GENERATE(<table1>, <table2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
table1	Qualquer expressão DAX que retorna uma tabela.
table2	Qualquer expressão DAX que retorna uma tabela.

## Valor retornado

Uma tabela com o produto cartesiano entre cada linha em *table1* e a tabela resultante da avaliação *table2* no contexto da linha atual de *table1*

## Comentários

- Se a avaliação de *table2* para a linha atual em *table1* retornar uma tabela vazia, a tabela de resultados não conterá a linha atual de *table1*. Isso é diferente de `GENERATEALL()`, em que a linha atual de *table1* será incluída nos resultados, e as colunas correspondentes a *table2* terão valores nulos para essa linha.
- Todos os nomes de coluna de *table1* e *table2* devem ser diferentes ou um erro é retornado.

## Exemplo

No exemplo a seguir, o usuário deseja uma tabela de resumo das vendas por região e categoria de produto para o canal de revendedores, como a tabela a seguir:

SalesTerritory[SalesTerritoryGroup]	ProductCategory[ProductCategoryName]	[Vendas do Revendedor]
Europa	Acessórios	US\$ 142.227,27
Europa	Bicicletas	US\$ 9.970.200,44
Europa	Vestuário	US\$ 365.847,63
Europa	Componentes	US\$ 2.214.440,19

América do Norte	Acessórios	US\$ 379.305,15
América do Norte	Bicicletas	US\$ 52.403.796,85
América do Norte	Vestuário	US\$ 1.281.193,26
América do Norte	Componentes	US\$ 8.882.848,05
Pacífico	Acessórios	US\$ 12.769,57
Pacífico	Bicicletas	US\$ 710.677,75
Pacífico	Vestuário	US\$ 22.902,38
Pacífico	Componentes	US\$ 108.549,71

O código a seguir produz a tabela acima:

```
GENERATE(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
,SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD])
)
)
```

1. A primeira instrução SUMMARIZE, `SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])`, produz uma tabela de grupos de regiões, em que cada linha é um grupo de regiões, como mostrado abaixo:

SALESTERRITORY[SALESTERRITORYGROUP]
América do Norte
Europa
Pacífico
NA

2. A segunda instrução SUMMARIZE,

```
SUMMARIZE(ProductCategory, [ProductCategoryName], "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD]))
```

, produz uma tabela de grupos de categorias de produtos com as vendas do revendedor para cada grupo, conforme mostrado abaixo:

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
Bicicletas	US\$ 63.084.675,04
Componentes	US\$ 11.205.837,96
Vestuário	US\$ 1.669.943,27



PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
Acessórios	US\$ 534.301,99

3. No entanto, quando você observa a tabela acima e a avalia no contexto de cada linha da tabela de grupos de regiões, obtém resultados diferentes para cada região.

# GENERATEALL

09/05/2020 • 3 minutes to read

Retorna uma tabela com o produto cartesiano entre cada linha em *table1* e a tabela resultante da avaliação de *table2* no contexto da linha atual de *table1*.

## Sintaxe

```
GENERATEALL(<table1>, <table2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
table1	Qualquer expressão DAX que retorna uma tabela.
table2	Qualquer expressão DAX que retorna uma tabela.

## Valor retornado

Uma tabela com o produto cartesiano entre cada linha em *table1* e a tabela resultante da avaliação *table2* no contexto da linha atual de *table1*

## Comentários

- Se a avaliação de *table2* para a linha atual em *table1* retornar uma tabela vazia, a linha atual de *table1* será incluída nos resultados e as colunas correspondentes a *table2* terão valores nulos para essa linha. Isso é diferente de `GENERATE()`, em que a linha atual de *table1* \*\*\**não*\*\*\* será incluída nos resultados.
- Todos os nomes de coluna de *table1* e *table2* devem ser diferentes ou um erro é retornado.

## Exemplo

No exemplo a seguir, o usuário deseja uma tabela de resumo das vendas por região e categoria de produto para o canal de revendedores, como a tabela a seguir:

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
Europa	Acessórios	US\$ 142.227,27
Europa	Bicicletas	US\$ 9.970.200,44
Europa	Vestuário	US\$ 365.847,63
Europa	Componentes	US\$ 2.214.440,19
NA	Acessórios	

SALESTERRITORY[SALESTERRITORYGROUP]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
NA	Bicicletas	
NA	Vestuário	
NA	Componentes	
América do Norte	Acessórios	US\$ 379.305,15
América do Norte	Bicicletas	US\$ 52.403.796,85
América do Norte	Vestuário	US\$ 1.281.193,26
América do Norte	Componentes	US\$ 8.882.848,05
Pacífico	Acessórios	US\$ 12.769,57
Pacífico	Bicicletas	US\$ 710.677,75
Pacífico	Vestuário	US\$ 22.902,38
Pacífico	Componentes	US\$ 108.549,71

O código a seguir produz a tabela acima:

```

GENERATEALL(
SUMMARIZE(SalesTerritory, SalesTerritory[SalesTerritoryGroup])
, SUMMARIZE(ProductCategory
, [ProductCategoryName]
, "Reseller Sales", SUMX(RELATEDTABLE(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD])
)
)

```

1. A primeira instrução SUMMARIZE produz uma tabela de grupos de regiões, em que cada linha é um grupo de regiões, como as mostradas abaixo:

SALESTERRITORY[SALESTERRITORYGROUP]
América do Norte
Europa
Pacífico
NA

2. A segunda instrução SUMMARIZE produz uma tabela de grupos de categorias de produtos com as vendas do revendedor para cada grupo, conforme mostrado abaixo:

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
Bicicletas	US\$ 63.084.675,04

PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VENDAS DO REVENDEDOR]
--------------------------------------	------------------------

Componentes	US\$ 11.205.837,96
Vestuário	US\$ 1.669.943,27
Acessórios	US\$ 534.301,99

3. No entanto, quando você observa a tabela acima e avalia a tabela no contexto de cada linha da tabela de grupos de regiões, obtém resultados diferentes para cada região.

# GEOMEAN

09/05/2020 • 2 minutes to read

Retorna a média geométrica dos números em uma coluna.

Para retornar a média geométrica de uma expressão avaliada para cada linha de uma tabela, use a [função GEOMEANX \(DAX\)](#).

## Sintaxe

```
GEOMEAN(<column>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números para os quais a média geométrica deve ser computada.

## Valor retornado

Um número decimal.

## Comentários

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

GEOMEAN( Table[Column] ) é equivalente a GEOMEANX( Table, Table[Column] )

## Exemplo

O seguinte computa a média geométrica da coluna Retorno na tabela Investimentos:

```
=GEOMEAN(Investment[Return])
```

## Consulte também

[Função GEOMEANX \(DAX\)](#)

# GEOMEANX

08/05/2020 • 2 minutes to read

Retorna a média geométrica de uma expressão avaliada para cada linha de uma tabela.

Para retornar a média geométrica dos números em uma coluna, use [Função GEOMEAN \(DAX\)](#).

## Sintaxe

```
GEOMEANX(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

## Valor retornado

Um número decimal.

## Comentários

A função GEOMEANX leva como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento é uma coluna que contém os números para os quais você deseja calcular a média geométrica ou uma expressão avaliada como uma coluna.

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

## Exemplo

O seguinte computa a média geométrica da coluna ReturnPct na tabela Investimentos:

```
=GEOMEANX(Investments, Investments[ReturnPct] + 1)
```

## Consulte também

[Função GEOMEAN \(DAX\)](#)

# MAX

09/05/2020 • 2 minutes to read

Retorna o maior valor de uma coluna ou entre duas expressões escalares.

## Sintaxe

```
MAX(<column>)
```

```
MAX(<expression1>, <expression2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna na qual você deseja encontrar o maior valor.
expressão	Qualquer expressão DAX que retorna um único valor.

## Valor da propriedade/valor retornado

O maior valor.

## Comentários

Ao comparar duas expressões, o espaço em branco é tratado como 0 durante a comparação. Ou seja, Max (1, Blank()) retorna 1 e Max (-1, Blank()) retorna 0. Se ambos os argumentos estiverem em branco, a função MAX retornará um espaço em branco. Se qualquer uma das expressões retornar um valor que não é permitido, a função MAX retornará um erro.

Não há suporte para valores TRUE/FALSE. Caso você deseje avaliar uma coluna de valores TRUE/FALSE, use a função MAXA.

## Exemplo 1

O exemplo a seguir retorna o maior valor encontrado na coluna ExtendedAmount da tabela InternetSales.

```
=MAX(InternetSales[ExtendedAmount])
```

## Exemplo 2

O exemplo a seguir retorna o maior valor entre o resultado de duas expressões.

```
=Max([TotalSales], [TotalPurchases])
```

## Consulte também

[Função MAXA \(DAX\)](#)

[Função MAXX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)



# MAXA

08/05/2020 • 2 minutes to read

Retorna o maior valor de uma coluna.

## Sintaxe

```
MAXA(<column>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna na qual você deseja encontrar o maior valor.

## Valor retornado

O maior valor.

## Comentários

A função MAXA usa como argumento uma coluna e procura o maior valor entre os seguintes tipos de valores:

- Números
- Datas
- Valores lógicos, como TRUE e FALSE. Linhas avaliadas como TRUE contam como 1; linhas avaliadas como FALSE contam como 0 (zero).

Células vazias são ignoradas. Se a coluna não contiver nenhum valor que possa ser usado, MAXA retornará 0 (zero).

Se quiser comparar valores de texto, use a função MAX.

## Exemplo 1

O exemplo a seguir retorna o valor máximo de uma coluna calculada, chamada **ResellerMargin**, que calcula a diferença entre o preço da lista e o preço do revendedor.

```
=MAXA([ResellerMargin])
```

## Exemplo 2

O exemplo a seguir retorna o maior valor de uma coluna que contém datas e horas. Portanto, essa fórmula obtém a data da transação mais recente.

```
=MAXA([TransactionDate])
```

## Consulte também

[Função MAX \(DAX\)](#)

[Função MAXX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

# MAXX

09/05/2020 • 2 minutes to read

Avalia uma expressão para cada linha de uma tabela e retorna o maior valor.

## Sintaxe

```
MAXX(<table>,<expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

## Valor retornado

O maior valor.

## Comentários

O argumento **table** para a função MAXX pode ser um nome de tabela ou uma expressão que é avaliada para uma tabela. O segundo argumento indica a expressão a ser avaliada para cada linha da tabela.

Dos valores a serem avaliados, somente os seguintes são contados:

- Números
- Textos
- Datas

Os valores em branco são ignorados. Não há suporte para valores TRUE/FALSE.

## Exemplo 1

A fórmula a seguir usa uma expressão como o segundo argumento para calcular a quantidade total de impostos e a entrega de cada pedido na tabela, InternetSales. O resultado esperado é 375,7184.

```
=MAXX(InternetSales, InternetSales[TaxAmt]+ InternetSales[Freight])
```

## Exemplo 2

A fórmula a seguir primeiro filtra a tabela InternetSales, usando uma expressão de filtro, para retornar um subconjunto de pedidos para uma região de vendas específica, definida como [SalesTerritory] = 5. Em seguida, a função MAXX avalia a expressão usada como o segundo argumento para cada linha da tabela filtrada e retorna o valor mais alto para impostos e entregas, apenas para esses pedidos. O resultado esperado é 250.3724.

```
=MAXX(FILTER(InternetSales,[SalesTerritoryCode]="5"), InternetSales[TaxAmt]+ InternetSales[Freight])
```

## Consulte também

[Função MAX \(DAX\)](#)

[Função MAXA \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

# MEDIAN

09/05/2020 • 2 minutes to read

Retorna a mediana dos números de uma coluna.

Para retornar o valor mediano de uma expressão avaliada para cada linha em uma tabela, use a [função MEDIANX \(DAX\)](#).

## Sintaxe

```
MEDIAN(<column>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna que contém os números para os quais o valor mediano deve ser computado.

## Valor retornado

Um número decimal.

## Comentários

Somente os números na coluna são contados. Espaços em branco, valores lógicos e texto são ignorados.

MEDIAN( Table[Column] ) é equivalente a MEDIANX( Table, Table[Column] ).

## Exemplo

O demonstrado a seguir computa o valor mediano de uma coluna chamada Age em uma tabela chamada Customers:

```
=MEDIAN(Customers[Age])
```

## Consulte também

[Função MEDIANX \(DAX\)](#)

# MEDIANX)

09/05/2020 • 2 minutes to read

Retorna o número mediano de uma expressão avaliada para cada linha de uma tabela.

Para retornar a mediana dos números em uma coluna, use a [função MEDIAN \(DAX\)](#).

## Sintaxe

```
MEDIANX(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

## Valor retornado

Um número decimal.

## Comentários

A função MEDIANX leva como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento é uma coluna que contém os números para os quais você deseja calcular a mediana ou uma expressão avaliada como uma coluna.

Somente os números na coluna são contados.

Valores lógicos e de texto são ignorados.

MEDIANX não ignora espaços em branco; no entanto, a MEDIAN ignora

## Exemplo

O seguinte calcula a idade mediana dos clientes que moram nos EUA.

```
=MEDIANX(FILTER(Customers, RELATED(Geography[Country]="USA")), Customers[Age])
```

## Consulte também

[Função MEDIAN \(DAX\)](#)

# MIN

08/05/2020 • 2 minutes to read

Retorna o menor valor de uma coluna ou entre duas expressões escalares.

## Sintaxe

```
MIN(<column>)
```

```
MIN(<expression1>, <expression2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna na qual você deseja encontrar o menor valor.
expressão	Qualquer expressão DAX que retorna um único valor.

## Valor retornado

O menor valor.

## Comentários

A função MIN usa uma coluna ou duas expressões como argumento e retorna o menor valor. Os seguintes tipos de valores nas colunas são contados:

- Números
- Textos
- Datas
- Em branco

Ao comparar expressões, o espaço em branco é tratado como 0 durante a comparação. Ou seja, `Min(1,Blank() )` retorna 0 e `Min( -1, Blank() )` retorna -1. Se ambos os argumentos estiverem em branco, a função MIN retornará um espaço em branco. Se qualquer uma das expressões retornar um valor que não é permitido, a função MIN retornará um erro.

Não há suporte para valores TRUE/FALSE. Caso você deseje avaliar uma coluna de valores TRUE/FALSE, use a função MINA.

## Exemplo 1

O exemplo a seguir retorna o menor valor da coluna calculada, `ResellerMargin`.

```
=MIN([ResellerMargin])
```

## Exemplo 2

O exemplo a seguir retorna o menor valor de uma coluna que contém datas e horas, TransactionDate. Portanto, essa fórmula retorna a data mais antiga.

```
=MIN([TransactionDate])
```

## Exemplo 3

O exemplo a seguir retorna o menor valor do resultado de duas expressões escalares.

```
=Min([TotalSales], [TotalPurchases])
```

## Consulte também

[Função MINA \(DAX\)](#)

[Função MINX \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)



# MINA

09/05/2020 • 2 minutes to read

Retorna o menor valor em uma coluna.

## Sintaxe

```
MINA(<column>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	A coluna cujo valor mínimo você deseja encontrar.

## Valor retornado

O menor valor.

## Comentários

A função MINA usa como argumento uma coluna que contém números e determina o menor valor da seguinte maneira:

- Se a coluna não contiver nenhum valor, MINA retornará 0 (zero).
- As linhas na coluna que são avaliadas como valores lógicos, como TRUE e FALSE, são tratadas como 1 se TRUE e 0 (zero) se FALSE.
- Células vazias são ignoradas.

Se você quiser comparar valores de texto, use a função MIN.

## Exemplo 1

A expressão a seguir retorna o encargo de frete mínimo da tabela, InternetSales.

```
=MINA(InternetSales[Freight])
```

## Exemplo 2

A expressão a seguir retorna o valor mínimo na coluna, PostalCode. Já que o tipo de dados da coluna é texto, a função não encontra nenhum valor e a fórmula retorna zero (0).

```
=MINA([PostalCode])
```

## Consulte também

[Função MIN \(DAX\)](#)  
[Função MINX \(DAX\)](#)  
[Funções estatísticas \(DAX\)](#)

# MINX

09/05/2020 • 2 minutes to read

Retorna o menor valor resultante da avaliação de uma expressão para cada linha de uma tabela.

## Sintaxe

```
MINX(<table>, < expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.

## Valor retornado

Um menor valor.

## Comentários

A função MINX leva como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento contém a expressão que é avaliada para cada linha da tabela.

Os valores em branco são ignorados. Não há suporte para valores TRUE/FALSE.

## Exemplo 1

O exemplo a seguir filtra a tabela, InternetSales, e retorna somente as linhas de uma região de vendas específica. Em seguida, a fórmula localiza o valor mínimo na coluna, Frete.

```
=MINX(FILTER(InternetSales, [SalesTerritoryKey] = 5),[Freight])
```

## Exemplo 2

O exemplo a seguir usa a mesma tabela filtrada que no exemplo anterior, mas em vez de apenas procurar valores na coluna para cada linha da tabela filtrada, a função calcula a soma de duas colunas, Frete e TaxAmt, e retorna o menor valor resultante desse cálculo.

```
=MINX(FILTER(InternetSales, InternetSales[SalesTerritoryKey] = 5), InternetSales[Freight] +
InternetSales[TaxAmt])
```

No primeiro exemplo, os nomes das colunas são não qualificados. No segundo exemplo, os nomes de coluna são totalmente qualificados.

## Consulte também

[Função MIN \(DAX\)](#)

[Função MINA \(DAX\)](#)

[Funções estatísticas \(DAX\)](#)

# NORM.DIST

08/05/2020 • 2 minutes to read

Retorna a distribuição normal para a média especificada e o desvio padrão.

## Sintaxe

```
NORM.DIST(X, Mean, Standard_dev, Cumulative)
```

### Parâmetros

TERMO	DEFINIÇÃO
X	O valor para o qual você deseja avaliar a distribuição.
Média	A média aritmética da distribuição.
Standard_dev	O desvio padrão da distribuição.
Cumulative*	Um valor lógico que determina a forma da função. Se cumulative for TRUE, NORM.DIST retornará a função de distribuição cumulativa; se for FALSE, retornará a função de densidade de probabilidade.

## Valor retornado

A distribuição normal para a média especificada e o desvio padrão.

## Exemplo

```
EVALUATE { NORM.DIST(42, 40, 1.5, TRUE) }
```

Retorna

[VALOR]

0,908788780274132

## Consulte também

[Função NORM.S.DIST](#)

[Função NORM.INV](#)

[NORM.S.INV](#)

# NORM.INV

09/05/2020 • 2 minutes to read

O inverso da distribuição cumulativa normal para a média especificada e o desvio padrão.

## Sintaxe

```
NORM.INV(Probability, Mean, Standard_dev)
```

### Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	Uma probabilidade correspondente à distribuição normal.
Média	A média aritmética da distribuição.
Standard_dev	O desvio padrão da distribuição.

## Valor retornado

Retorna o inverso da distribuição cumulativa normal para a média especificada e o desvio padrão.

## Exemplo

```
EVALUATE { NORM.INV(0.908789, 40, 1.5) }
```

Retorna

[VALOR]

42,00000200956628780274132

## Consulte também

[NORM.S.INV](#)

[Função NORM.S.DIST](#)

[Função NORM.DIST](#)

# NORM.S.DIST

09/05/2020 • 2 minutes to read

Retorna a distribuição normal padrão (tem uma média igual a zero e um desvio padrão de um).

## Sintaxe

```
NORM.S.DIST(Z, Cumulative)
```

### Parâmetros

TERMO	DEFINIÇÃO
Z	O valor para o qual você deseja avaliar a distribuição.
Cumulative	Cumulative é um valor lógico que determina a forma da função. Se cumulative for TRUE, NORM.S.DIST retornará a função de distribuição cumulativa; se for FALSE, retornará a função de densidade de probabilidade.

## Valor retornado

A distribuição normal padrão (tem uma média igual a zero e um desvio padrão de um).

## Exemplo

```
EVALUATE { NORM.S.DIST(1.333333, TRUE) }
```

Retorna

[VALOR]

0,908788725604095

## Consulte também

[Função NORM.INV](#)

[Função NORM.DIST](#)

[NORM.S.INV](#)

# NORM.S.INV

09/05/2020 • 2 minutes to read

Retorna o inverso da distribuição cumulativa normal padrão. A distribuição tem uma média igual a zero e um desvio padrão de um.

## Sintaxe

```
NORM.S.INV(Probability)
```

### Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	Uma probabilidade correspondente à distribuição normal.

## Valor retornado

O inverso da distribuição cumulativa normal padrão. A distribuição tem uma média igual a zero e um desvio padrão de um.

## Exemplo

```
EVALUATE { NORM.S.INV(0.908789) }
```

Retorna

[VALOR]

1,33333467304411

## Consulte também

[NORM.INV](#)

[Função NORM.S.DIST](#)

[Função NORM.DIST](#)



# PERCENTILE.EXC

09/05/2020 • 2 minutes to read

Retorna o k-ésimo percentil de valores em um intervalo, em que k está no intervalo de 0..1, exclusivo.

Para retornar o número percentil de uma expressão avaliada para cada linha de uma tabela, use a [função PERCENTILEX.EXC \(DAX\)](#).

## Sintaxe

```
PERCENTILE.EXC(<column>, <k>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma coluna que contém os valores que definem a posição relativa.
k	O valor do percentil no intervalo de 0.. 1, exclusivo.

## Valor retornado

O k-ésimo percentil de valores em um intervalo, em que k está no intervalo 0.. 1, exclusivo.

## Comentários

Se a coluna estiver vazia, BLANK() será retornado.

Se k for zero ou estiver em branco, a classificação percentil de  $1/(n+1)$  retornará o menor valor. Se for zero, ele estará fora do intervalo e um erro será retornado.

Se k não for numérico ou estiver fora do intervalo de 0 a 1, um erro será retornado.

Se k não for um múltiplo de  $1/(n+1)$ , PERCENTILE.EXC será interpolado para determinar o valor no k-ésimo percentil.

PERCENTILE.EXC será interpolado quando o valor do percentil especificado estiver entre dois valores na matriz. Se não for possível interpolar para o percentil de k especificado, um erro será retornado.

## Consulte também

[Função PERCENTILEX.EXC \(DAX\)](#)

# PERCENTILE.INC

09/05/2020 • 2 minutes to read

Retorna o k-ésimo percentil de valores em um intervalo, em que k está no intervalo de 0..1, inclusivo.

Para retornar o número percentil de uma expressão avaliada para cada linha de uma tabela, use a [função PERCENTILEX.INC \(DAX\)](#).

## Sintaxe

```
PERCENTILE.INC(<column>, <k>)
```

### Parâmetros

TERMO	DEFINIÇÃO
coluna	Uma coluna que contém os valores que definem a posição relativa.
k	O valor percentual deve estar no intervalo de 0 a 1, inclusive.

## Valor retornado

O k-ésimo percentil de valores em um intervalo, em que k está no intervalo de 0 a 1, inclusive.

## Comentários

Se a coluna estiver vazia, BLANK() será retornado.

Se k for zero ou estiver em branco, a classificação percentil de  $1/(n+1)$  retornará o menor valor. Se for zero, ele estará fora do intervalo e um erro será retornado.

Se k não for numérico ou estiver fora do intervalo de 0 a 1, um erro será retornado.

Se k não for um múltiplo de  $1/(n+1)$ , PERCENTILE.INC será interpolado para determinar o valor no k-ésimo percentil.

PERCENTILE.INC interpolará quando o valor do percentil especificado estiver entre dois valores na matriz. Se não for possível interpolar para o percentil de k especificado, um erro será retornado.

## Consulte também

[Função PERCENTILEX.INC \(DAX\)](#)

# PERCENTILEX.EXC

09/05/2020 • 2 minutes to read

Retorna o número de percentil de uma expressão avaliada para cada linha de uma tabela.

Para retornar o percentil dos números em uma coluna, use [Função PERCENTILE.EXC \(DAX\)](#).

## Sintaxe

```
PERCENTILEX.EXC(<table>, <expression>, k)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.
k	O valor do percentil desejado no intervalo de 0 a 1, exclusivo.

## Valor retornado

O número percentil de uma expressão avaliada para cada linha de uma tabela.

## Comentários

Se k for zero ou estiver em branco, a classificação percentil de  $1/(n+1)$  retornará o menor valor. Se for zero, ele estará fora do intervalo e um erro será retornado.

Se k não for numérico ou estiver fora do intervalo de 0 a 1, um erro será retornado.

Se k não for um múltiplo de  $1/(n+1)$ , PERCENTILEX.EXC será interpolado para determinar o valor no k-ésimo percentil.

PERCENTILEX.EXC será interpolado quando o valor do percentil especificado estiver entre dois valores na matriz. Se não for possível interpolar para o percentil de k especificado, um erro será retornado.

## Consulte também

[Função PERCENTILE.EXC \(DAX\)](#)

# PERCENTILEX.INC

09/05/2020 • 2 minutes to read

Retorna o número de percentil de uma expressão avaliada para cada linha de uma tabela.

Para retornar o percentual de números em uma coluna, use [Função PERCENTILE.INC \(DAX\)](#).

## Sintaxe

```
PERCENTILEX.INC(<table>, <expression>;, k)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.
k	O valor percentual desejado no intervalo de 0 a 1, inclusive.

## Valor retornado

O número percentil de uma expressão avaliada para cada linha de uma tabela.

## Comentários

Se k for zero ou em branco, a classificação percentual de  $1/(n-1)$  retornará o menor valor. Se for zero, ele estará fora do intervalo e um erro será retornado.

Se k não for numérico ou estiver fora do intervalo de 0 a 1, um erro será retornado.

Se k não for um múltiplo de  $1/(n-1)$ , PERCENTILEX.EXC interpolará para determinar o valor no k-ésimo percentil.

PERCENTILEX.INC interpolará quando o valor do percentil especificado estiver entre dois valores na matriz. Se não for possível interpolar para o percentil de k especificado, um erro será retornado.

## Consulte também

[Função PERCENTILE.INC \(DAX\)](#)

# PERMUT

08/05/2020 • 2 minutes to read

Retorna o número de permutações para um determinado número de objetos que podem ser selecionados entre objetos numéricos. Uma permutação é qualquer conjunto ou subconjunto de objetos ou eventos em que a ordem interna é significativa. Permutas são diferentes de combinações, para as quais a ordem interna não é significativa. Use essa função para cálculos de probabilidade no estilo de loteria.

## Sintaxe

PERMUT(number, number\_chosen)

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. Um inteiro que descreve um número de objetos.
number_chosen	Obrigatório. Um inteiro que descreve um número de objetos em cada permuta.

## Valor retornado

Retorna o número de permutações para um determinado número de objetos que podem ser selecionados entre objetos numéricos

## Comentários

Os dois argumentos numéricos são truncados para inteiros.

Se number ou number\_chosen não for um valor numérico, PERMUT retornará o valor de erro #VALUE!.

Se number  $\leq 0$  ou number\_chosen  $< 0$ , PERMUT retornará o valor de erro #VALUE!.

Se number  $<$  number\_chosen, PERMUT retornará o valor de erro #VALUE!.

A equação para o número de permutas é:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=PERMUT(3,2)	Permutas possíveis para um grupo de 3 objetos em que 2 são escolhidos.	6

# POISSON.DIST

08/05/2020 • 2 minutes to read

Retorna a distribuição de Poisson. Uma aplicação comum dessa distribuição é prever o número de eventos em um determinado tempo, como o número de carros que chegam a um pedágio em 1 minuto.

## Sintaxe

POISSON.DIST(x,mean,cumulative)

### Parâmetros

TERMO	DEFINIÇÃO
x	Obrigatório. O número de eventos.
mean	Obrigatório. O valor numérico esperado.
cumulative	Obrigatório. Um valor lógico que determina a forma da distribuição de probabilidade retornada. Se cumulative for TRUE, POISSON.DIST retornará a probabilidade de Poisson cumulativa de o número de eventos aleatórios ocorrendo estar entre zero e x, inclusive. Se for FALSE, retornará a função de probabilidade de Poisson de massa de o número de eventos ocorrendo ser exatamente x.

## Valor retornado

Retorna a distribuição de Poisson.

## Comentários

Se x não for um inteiro, ele será arredondado.

Se x ou mean não for numérico, POISSON.DIST retornará o valor de erro #VALUE!.

Se x < 0, POISSON.DIST retornará o valor de erro #NUM!.

Se mean < 0, POISSON.DIST retornará o valor de erro #VALUE!.

POISSON.DIST é calculado da seguinte maneira.

Para cumulative = FALSE:

$$POISSON = \frac{e^{-\lambda} \lambda^x}{x!}$$

Para cumulative = TRUE:

$$CUMPOISSON = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$

# RANK.EQ

09/05/2020 • 3 minutes to read

Retorna a classificação de um número em uma lista de números.

## Sintaxe

```
RANK.EQ(<value>, <columnName>[, <order>])
```

### Parâmetros

valor

Qualquer expressão DAX que retorna um único valor escalar cuja classificação precisa ser encontrada. A expressão deve ser avaliada exatamente uma vez, antes que a função seja avaliada, e é o valor passado para a lista de argumentos.

columnName

O nome de uma coluna existente na qual as classificações serão determinadas. Ele não pode ser uma expressão ou uma coluna criada usando essas funções: ADDCOLUMNS, ROW ou SUMMARIZE.

order

(Opcional) Um valor que especifica como classificar *number*, de menor para maior ou de maior para menor:

valor	valor alternativo	Descrição
0 (zero)	FALSO	Classifica em ordem decrescente de <i>columnName</i> . Se <i>value</i> for igual ao maior número em <i>columnName</i> , RANK.EQ será 1.
1	VERDADEIRO	Classifica em ordem crescente de <i>columnName</i> . Se <i>value</i> for igual ao menor número em <i>columnName</i> , RANK.EQ será 1.

## Valor retornado

Um número que indica a classificação de *value* entre os números em *columnName*.

## Exceções

## Comentários

- columnName* não pode fazer referência a nenhuma coluna criada usando essas funções: ADDCOLUMNS, ROW ou SUMMARIZE.
- Se *value* não estiver em *columnName* ou o valor estiver em branco, RANK.EQ retornará um valor em branco.
- Valores duplicados de *value* recebem o mesmo valor de classificação. O valor de classificação atribuído

seguinte será o valor de classificação mais o número de valores duplicados. Por exemplo, se cinco (5) valores estiverem empatados com uma classificação de 11, o valor seguinte receberá uma classificação de 16 (11 + 5).

## Exemplo

O exemplo a seguir cria uma coluna calculada que classifica os valores em SalesAmount\_USD, da tabela *InternetSales\_USD*, em relação a todos os números na mesma coluna.

```
=RANK.EQ(InternetSales_USD[SalesAmount_USD], InternetSales_USD[SalesAmount_USD])
```

## Exemplo

O exemplo a seguir classifica um subconjunto de valores em relação a uma determinada amostra. Suponha que você tenha uma tabela de alunos locais com seu desempenho em um teste nacional específica e, além disso, tenha todo o conjunto de pontuações nesse teste nacional. A coluna calculada a seguir fornecerá a classificação nacional referente a cada um dos alunos locais.

```
=RANK.EQ(Students[Test_Score], NationalScores[Test_Score])
```



# RANKX

09/05/2020 • 4 minutes to read

Retorna a classificação de um número em uma lista de números para cada linha no argumento *table*.

## Sintaxe

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

### Parâmetros

tabela

Qualquer expressão DAX que retorna uma tabela de dados sobre a qual a expressão é avaliada.

expressão

Qualquer expressão DAX que retorna um único valor escalar. A expressão é avaliada para cada linha de *table*, para gerar todos os valores possíveis para classificação. Confira a seção comentários para entender o comportamento da função quando *expression* é avaliado como BLANK.

valor

(Opcional) Qualquer expressão DAX que retorna um valor escalar único cuja classificação deve ser encontrada. Confira a seção de comentários para entender o comportamento da função quando *value* não é encontrado na expressão.

Quando o parâmetro *value* é omitido, o valor da expressão na linha atual é usado em vez disso.

order

(Opcional) Um valor que especifica como classificar *value*, do menor para o maior ou vice-versa:

valor	valor alternativo	Descrição
0 (zero)	FALSO	Classifica em ordem decrescente de valores de expression. Se value for igual ao número mais alto na expressão, RANKX retornará 1.  É o valor padrão quando o parâmetro order é omitido.
1	VERDADEIRO	Classifica em ordem crescente de expressão. Se value for igual ao número mais baixo na expressão, RANKX retornará 1.

empates

(Opcional) Uma enumeração que define como determinar a classificação quando há empates.

enumeração	Descrição

Ignorar	<p>O próximo valor de classificação, após um empate, é o valor de classificação do empate mais a contagem de valores empatados. Por exemplo, se cinco (5) valores estiverem empatados com uma classificação de 11, o valor seguinte receberá uma classificação de 16 (11 + 5).</p> <p>É o valor padrão quando o parâmetro <i>ties</i> é omitido.</p>
Denso	<p>O próximo valor de classificação após um empate é o valor da próxima classificação. Por exemplo, se cinco (5) valores estiverem empatados com uma classificação de 11, o próximo valor receberá uma classificação de 12.</p>

## Valor retornado

O número de classificação de *value* entre todos os valores possíveis de *expression* avaliados para todas as linhas dos números de *table*.

## Exceções

## Comentários

- Se *expression* ou *value* for avaliado como BLANK, ele será tratado como um 0 (zero) para todas as expressões que resultam em um número ou como um texto vazio para todas as expressões de texto.
- Se *value* não estiver entre todos os valores possíveis de *expression*, o RANKX adicionará temporariamente *value* aos valores de *expression* e reavaliará RANKX para determinar a classificação adequada de *value*.
- Argumentos opcionais podem ser ignorados colocando-se uma vírgula vazia (,) na lista de argumentos, ou seja, RANKX(Inventory, [InventoryCost],,"Dense")

## Exemplo

A coluna calculada a seguir na tabela Produtos calcula a classificação de vendas para cada produto no canal da Internet.

```
=RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))
```

# Função ROW

09/05/2020 • 2 minutes to read

Retorna uma tabela com uma única linha contendo valores que resultam das expressões fornecidas para cada coluna.

## Sintaxe

```
ROW(<name>, <expression>[[,<name>, <expression>]...])
```

### Parâmetros

TERMO	DEFINIÇÃO
Nome	O nome dado à coluna, entre aspas duplas.
expressão	Qualquer expressão DAX que retorne um único valor escalar para preenchimento. <i>name</i> .

## Valor retornado

Uma tabela com uma única linha

## Comentários

Os argumentos sempre devem vir em pares de *name* e *expression*.

## Exemplo

O exemplo a seguir retorna uma tabela com uma única linha que contém as vendas totais pela Internet e pelos canais de revendedores.

```
ROW("Internet Total Sales (USD)", SUM(InternetSales_USD[SalesAmount_USD]),
 "Resellers Total Sales (USD)", SUM(ResellerSales_USD[SalesAmount_USD]))
```

O código é dividido em duas linhas para fins de legibilidade

# SAMPLE

09/05/2020 • 2 minutes to read

Retorna uma amostra de N linhas da tabela especificada.

## Sintaxe

```
SAMPLE(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

### Parâmetros

**n\_value**

Número de linhas a ser retornado. É qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto). Se um valor não inteiro (ou uma expressão) for inserido(a), o resultado será convertido em um inteiro.

**tabela**

Qualquer expressão DAX que retorna uma tabela de dados de onde extrair as 'n' linhas de amostra.

**orderBy\_expression**

(Opcional) Qualquer expressão DAX escalar em que o valor de resultado seja avaliado para cada linha de *table*.

**order**

(Opcional) Um valor que especifica como classificar valores *orderBy\_expression* em ordem crescente ou decrescente:

valor	valor alternativo	Descrição
0 (zero)	FALSO	Classifica em ordem decrescente de valores de <i>order_by</i> .  É o valor padrão quando o parâmetro <i>order</i> é omitido.
1	VERDADEIRO	Classifica em ordem crescente de <i>order_by</i> .

## Valor retornado

Uma tabela que consiste em uma amostra de N linhas de *table* ou uma tabela vazia se *n\_value* é menor ou igual a 0 (zero). Se forem fornecidos argumentos de OrderBy, a amostra será estável e determinística, retornando a primeira linha, a última linha e as linhas igualmente distribuídas entre elas. Se nenhuma ordem for especificada, a amostra será aleatória, não estável e não determinística.

## Comentários

- Se *n\_value* for menor ou igual a 0 (zero), a função SAMPLE retornará uma tabela vazia.
- Para evitar valores duplicados na amostra, a tabela fornecida como segundo argumento deverá ser agrupada pela coluna usada para classificação.

# SELECTCOLUMNS

09/05/2020 • 2 minutes to read

Adiciona colunas calculadas à tabela ou à expressão de tabela fornecida.

## Sintaxe

```
SELECTCOLUMNS(<table>, <name>, <scalar_expression> [, <name>, <scalar_expression>]...)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela.
Nome	O nome dado à coluna, entre aspas duplas.
expressão	Qualquer expressão que retorna um valor escalar, como uma referência de coluna, um inteiro ou uma cadeia de caracteres.

## Valor retornado

Uma tabela com o mesmo número de linhas que a tabela especificada como o primeiro argumento. A tabela retornada tem uma coluna para cada par de argumentos <name>, <scalar\_expression> e cada expressão é avaliada no contexto de uma linha do argumento <table> especificado.

## Comentários

SELECTCOLUMNS tem a mesma assinatura de ADDCOLUMNS e tem o mesmo comportamento, exceto que, em vez de iniciar com a <table> especificada, SELECTCOLUMNS começa com uma tabela vazia, antes da adição de colunas.

## Exemplo

Para a tabela a seguir, chamada **Info**:

PAÍS	ESTADO	CONTAGEM	TOTAL
IND	JK	20	800
IND	MH	25	1000
IND	WB	10	900
EUA	AC	5	500
EUA	WA	10	900

```
SELECT COLUMN$ (Info, "StateCountry", [State]&"", "&[Country])
```

Retorna:

STATECOUNTRY
IND, JK
IND, MH
IND, WB
USA, CA
USA, WA

# SIN

09/05/2020 • 2 minutes to read

Retorna o seno do ângulo determinado.

## Sintaxe

`SIN(number)`

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. O ângulo em radianos para o qual você quer o seno.

## Valor retornado

Retorna o seno do ângulo determinado.

## Comentários

Se o argumento estiver em graus, multiplique-o por  $\text{PI}()/180$  ou use a função `RADIANS` para convertê-lo em radianos.

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
<code>=SIN(PI())</code>	Seno de Pi radianos (aproximadamente 0).	0,0
<code>=SIN(PI()/2)</code>	Seno de $\text{Pi}/2$ radianos.	1.0
<code>=SIN(30*PI()/180)</code>	Seno de 30 graus.	0,5
<code>=SIN(RADIANS(30))</code>	Seno de 30 graus.	0,5

# SINH

09/05/2020 • 2 minutes to read

Retorna o seno hiperbólico de um número.

## Sintaxe

SINH(number)

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. Qualquer número real.

## Valor retornado

Retorna o seno hiperbólico de um número.

## Comentários

A fórmula do cosseno hiperbólico é:

$$\text{SINH}(z) = \frac{e^z - e^{-z}}{2}$$

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
= 2,868 * SINH (0,0342*1,03)	Probabilidade de obter um resultado de menor de 1,03 segundos.	0,1010491



# STDEV.S

09/05/2020 • 2 minutes to read

Retorna o desvio padrão de uma amostra de população.

## Sintaxe

```
STDEV.S(<ColumnName>)
```

### Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.

## Valor retornado

Um número que representa o desvio padrão de uma amostra de população.

## Exceções

## Comentários

1. STDEV.S supõe que a coluna se refere a uma amostra da população. Se os dados representarem a população inteira, calcule o desvio padrão usando STDEV.P.

2. STDEV.S usa a seguinte fórmula:

$$\sqrt{\sum (x - \bar{x})^2 / (n-1)}$$

em que  $\bar{x}$  é o valor médio de x para a amostra da população

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma medida que calcula o desvio padrão da coluna, SalesAmount\_USD, quando a tabela InternetSales\_USD é a amostra da população.

```
=STDEV.S(InternetSales_USD[SalesAmount_USD])
```

# STDEV.P

09/05/2020 • 2 minutes to read

Retorna o desvio padrão da população inteira.

## Sintaxe

```
STDEV.P(<ColumnName>)
```

### Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.

## Valor retornado

Um número que representa o desvio padrão da população inteira.

## Exceções

## Comentários

1. STDEV.P presume que a coluna faz referência a toda a população. Se os dados representarem uma amostra da população, então calcule o desvio padrão usando a função STDEV.S.

2. STDEV.P usa a seguinte fórmula:

$$\sqrt{\sum (x - \bar{x})^2 / n}$$

em que  $\bar{x}$  é o valor médio de x para a população inteira

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma medida que calcula o desvio padrão da coluna SalesAmount\_USD quando a tabela InternetSales\_USD é a população inteira.

```
=STDEV.P(InternetSales_USD[SalesAmount_USD])
```

# STDEVX.S

09/05/2020 • 2 minutes to read

Retorna o desvio padrão de uma amostra de população.

## Sintaxe

```
STDEVX.S(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

## Valor retornado

Um número com o desvio padrão de uma amostra de população.

## Exceções

## Comentários

1. STDEVX.S avalia a *expressão* em cada linha da *tabela* e retorna o desvio padrão da *expressão*, supondo que a *tabela* faça referência a uma amostra da população. Se a *tabela* representar a população inteira, calcule o desvio padrão usando STDEVX.P.

2. STDEVX.S usa a seguinte fórmula:

$$\sqrt{\sum (x - \tilde{x})^2 / (n-1)}$$

em que  $\tilde{x}$  é o valor médio de x para a população inteira

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma coluna calculada que estima o desvio padrão do preço unitário por

produto para uma amostra de população quando a fórmula é usada na tabela Product.

```
=STDEVX.S(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# STDEVX.P

09/05/2020 • 2 minutes to read

Retorna o desvio padrão da população inteira.

## Sintaxe

```
STDEVX.P(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

## Valor retornado

Um número que representa o desvio padrão de uma população inteira.

## Comentários

1. STDEVX.P avalia *expression* para cada linha de *table* e retorna o desvio padrão de *expression*, assumindo que *table* se refira à população inteira. Se os dados em *table* representarem uma amostra da população, então você deverá calcular o desvio padrão usando a função STDEVX.S.

2. STDEVX.P usa a seguinte fórmula:

$$\sqrt{\sum (x - \tilde{x})^2 / n}$$

em que  $\tilde{x}$  é o valor médio de *x* para a população inteira

e *n* é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma coluna calculada que calcula o desvio padrão do preço unitário por produto, quando a fórmula é usada na tabela *Product*.

```
=STDEVX.P(RELATEDTABLE(InternetSales_USD), InternetSales_USD[UnitPrice_USD] -
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# SQRTPI

09/05/2020 • 2 minutes to read

Retorna a raiz quadrada de (número \* pi).

## Sintaxe

SQRTPI(number)

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. O número pelo qual pi é multiplicado.

## Valor retornado

Retorna a raiz quadrada de (número \* pi).

## Comentários

XXXXX

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=SQRTPI(1)	Raiz quadrada de pi.	1,772454
=SQRTPI(2)	Raiz quadrada de 2 * pi.	2,506628

# SUMMARIZE

09/05/2020 • 10 minutes to read

Retorna uma tabela de resumo para os totais solicitados sobre um conjunto de grupos.

## Sintaxe

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, <name>, <expression>]...)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
groupBy_columnName	(Opcional) O nome qualificado de uma coluna existente a ser usada para criar grupos de resumo com base nos valores encontrados nela. Esse parâmetro não pode ser uma expressão.
Nome	O nome fornecido a uma coluna total ou de resumo, entre aspas duplas.
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

## Valor retornado

Uma tabela com as colunas selecionadas para os argumentos *groupBy\_columnName* e as colunas resumidas criadas pelos argumentos de nome.

## Comentários

1. Cada coluna para a qual você define um nome deve ter uma expressão correspondente; caso contrário, um erro será retornado. O primeiro argumento, *name*, define o nome da coluna nos resultados. O segundo argumento, *expression*, define o cálculo executado para obter o valor de cada linha nessa coluna.
2. *groupBy\_columnName* deve estar em *table* ou em uma tabela relacionada a *table*.
3. Cada nome deve ser colocado entre aspas duplas.
4. A função agrupa um conjunto selecionado de linhas em um conjunto de linhas de resumo pelos valores de uma ou mais colunas de *groupBy\_columnName*. Uma linha é retornada para cada grupo.

## Exemplo

O exemplo a seguir retorna um resumo das vendas do revendedor agrupadas ao longo do ano civil e do nome da categoria do produto. Essa tabela de resultados permite analisar as vendas do revendedor por ano e categoria de produto.



```
SUMMARIZE(ResellerSales_USD
 , DateTime[CalendarYear]
 , ProductCategory[ProductCategoryName]
 , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
 , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

A tabela a seguir mostra uma visualização dos dados conforme eles seriam recebidos por qualquer função que espera receber uma tabela:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Bicicletas	12968255,42	36167,6592
2005	Bicicletas	6958251,043	4231,1621
2006	Bicicletas	18901351,08	178175,8399
2007	Bicicletas	24256817,5	276065,992
2008	Componentes	2008052,706	39,9266
2005	Componentes	574256,9865	0
2006	Componentes	3428213,05	948,7674
2007	Componentes	5195315,216	4226,0444
2008	Vestuário	366507,844	4151,1235
2005	Vestuário	31851,1628	90,9593
2006	Vestuário	455730,9729	4233,039
2007	Vestuário	815853,2868	12489,3835
2008	Acessórios	153299,924	865,5945
2005	Acessórios	18594,4782	4,293
2006	Acessórios	86612,7463	1061,4872
2007	Acessórios	275794,8403	4756,6546

## Opções avançadas de SUMMARIZE

### SUMMARIZE com ROLLUP

A adição da sintaxe ROLLUP() modifica o comportamento da função SUMMARIZE ao acrescentar linhas acumuladas ao resultado nas colunas groupBy\_columnName.

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, <expression>]...)
```

#### Parâmetros ROLLUP

groupBy\_columnName

O nome qualificado de uma coluna existente a ser usada para criar grupos de resumo com base nos valores encontrados nela. Esse parâmetro não pode ser uma expressão.

**Observação:** todos os outros parâmetros SUMMARIZE são explicados antes e não são repetidos aqui para fins de brevidade.

#### Comentários

- As colunas mencionadas na expressão ROLLUP não podem ser referenciadas como parte de colunas *groupBy\_columnName*.

#### Exemplo

O exemplo a seguir adiciona linhas acumuladas às colunas Group-By da chamada de função SUMMARIZE.

```
SUMMARIZE(ResellerSales_USD
 , ROLLUP(DateTime[CalendarYear], ProductCategory[ProductCategoryName])
 , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
 , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

A tabela a seguir mostra uma visualização dos dados conforme eles seriam recebidos por qualquer função que espera receber uma tabela:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Bicicletas	12968255,42	36167,6592
2005	Bicicletas	6958251,043	4231,1621
2006	Bicicletas	18901351,08	178175,8399
2007	Bicicletas	24256817,5	276065,992
2008	Componentes	2008052,706	39,9266
2005	Componentes	574256,9865	0
2006	Componentes	3428213,05	948,7674
2007	Componentes	5195315,216	4226,0444
2008	Vestuário	366507,844	4151,1235
2005	Vestuário	31851,1628	90,9593
2006	Vestuário	455730,9729	4233,039
2007	Vestuário	815853,2868	12489,3835

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Acessórios	153299,924	865,5945
2005	Acessórios	18594,4782	4,293
2006	Acessórios	86612,7463	1061,4872
2007	Acessórios	275794,8403	4756,6546
2008		15496115,89	41224,3038
2005		7582953,67	4326,4144
2006		22871907,85	184419,1335
2007		30543780,84	297538,0745
		76494758,25	527507,9262

## ROLLUPGROUP

ROLLUPGROUP() pode ser usado para calcular grupos de subtotais. Se usado no lugar de ROLLUP, ROLLUPGROUP produzirá o mesmo resultado adicionando linhas acumuladas ao resultado nas colunas groupBy\_columnName. No entanto, a adição de ROLLUPGROUP() dentro de uma sintaxe ROLLUP pode ser usada para impedir subtotais parciais em linhas acumuladas.

O exemplo a seguir mostra apenas o total geral de todos os anos e categorias sem o subtotal de cada ano com todas as categorias:

```
SUMMARIZE(ResellerSales_USD
, ROLLUP(ROLLUPGROUP(DateTime[CalendarYear], ProductCategory[ProductCategoryName]))
, "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
, "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
)
```

A tabela a seguir mostra uma visualização dos dados conforme eles seriam recebidos por qualquer função que espera receber uma tabela:

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2008	Bicicletas	12968255,42	36167,6592
2005	Bicicletas	6958251,043	4231,1621
2006	Bicicletas	18901351,08	178175,8399
2007	Bicicletas	24256817,5	276065,992
2008	Componentes	2008052,706	39,9266
2005	Componentes	574256,9865	0

DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
2006	Componentes	3428213,05	948,7674
2007	Componentes	5195315,216	4226,0444
2008	Vestuário	366507,844	4151,1235
2005	Vestuário	31851,1628	90,9593
2006	Vestuário	455730,9729	4233,039
2007	Vestuário	815853,2868	12489,3835
2008	Acessórios	153299,924	865,5945
2005	Acessórios	18594,4782	4,293
2006	Acessórios	86612,7463	1061,4872
2007	Acessórios	275794,8403	4756,6546
		76494758,25	527507,9262

## SUMMARIZE com ISSUBTOTAL

Permite que o usuário crie outra coluna, na função Summarize, que retornará True se a linha contiver valores de subtotal para a coluna fornecida como argumento para ISSUBTOTAL, caso contrário retornará False.

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]...[, ROLLUP(<groupBy_columnName>[, <groupBy_columnName>...])][, <name>, {<expression>|ISSUBTOTAL(<columnName>)}]...)
```

### Parâmetros ISSUBTOTAL

columnName

O nome de qualquer coluna na tabela da função SUMMARIZE ou qualquer coluna em uma tabela relacionada à tabela.

### Valor retornado

Um valor **True** se a linha contiver um valor subtotal para a coluna fornecida como argumento; caso contrário, retornará **False**

### Comentários

- ISSUBTOTAL só pode ser usado na parte da expressão de uma função SUMMARIZE.
- ISSUBTOTAL deve ser precedido por um *nome* de coluna correspondente.

### Exemplo

O exemplo a seguir gera uma coluna ISSUBTOTAL() para cada uma das colunas ROLLUP() na chamada de função SUMMARIZE() fornecida.

```

SUMMARIZE(ResellerSales_USD
 , ROLLUP(DateTime[CalendarYear], ProductCategory[ProductCategoryName])
 , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
 , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
 , "Is Sub Total for DateTimeCalendarYear", ISSUBTOTAL(DateTime[CalendarYear])
 , "Is Sub Total for ProductCategoryName", ISSUBTOTAL(ProductCategory[ProductCategoryName])
)

```

A tabela a seguir mostra uma visualização dos dados conforme eles seriam recebidos por qualquer função que espera receber uma tabela:

[É O SUBTOTAL PARA DATETIMECALENDARYEAR]	[É O SUBTOTAL PARA PRODUCTCATEGORYNAME]	DATETIME[CALENDARYEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
FALSO	FALSO				
FALSO	FALSO	2008	Bicicletas	12968255,42	36167,6592
FALSO	FALSO	2005	Bicicletas	6958251,043	4231,1621
FALSO	FALSO	2006	Bicicletas	18901351,08	178175,8399
FALSO	FALSO	2007	Bicicletas	24256817,5	276065,992
FALSO	FALSO	2008	Componentes	2008052,706	39,9266
FALSO	FALSO	2005	Componentes	574256,9865	0
FALSO	FALSO	2006	Componentes	3428213,05	948,7674
FALSO	FALSO	2007	Componentes	5195315,216	4226,0444
FALSO	FALSO	2008	Vestuário	366507,844	4151,1235
FALSO	FALSO	2005	Vestuário	31851,1628	90,9593
FALSO	FALSO	2006	Vestuário	455730,9729	4233,039
FALSO	FALSO	2007	Vestuário	815853,2868	12489,3835
FALSO	FALSO	2008	Acessórios	153299,924	865,5945
FALSO	FALSO	2005	Acessórios	18594,4782	4,293
FALSO	FALSO	2006	Acessórios	86612,7463	1061,4872
FALSO	FALSO	2007	Acessórios	275794,8403	4756,6546
FALSO	VERDADEIRO				
FALSO	VERDADEIRO	2008		15496115,89	41224,3038

[É O SUBTOTAL PARA DATETIMECALENDAR YEAR]	[É O SUBTOTAL PARA PRODUCTCATEGORYNAME]	DATETIME[CALENDAR YEAR]	PRODUCTCATEGORY[PRODUCTCATEGORYNAME]	[VALOR DAS VENDAS (USD)]	[VALOR DE DESCONTO (USD)]
FALSO	VERDADEIRO	2005		7582953,67	4326,4144
FALSO	VERDADEIRO	2006		22871907,85	184419,1335
FALSO	VERDADEIRO	2007		30543780,84	297538,0745
VERDADEIRO	VERDADEIRO			76494758,25	527507,9262

# T.DIST

08/05/2020 • 2 minutes to read

Retorna a distribuição t caudal esquerda do Student.

## Sintaxe

```
T.DIST(X,Deg_freedom,Cumulative)
```

### Parâmetros

TERMO	DEFINIÇÃO
X	O valor numérico no qual você deseja avaliar a distribuição.
Deg_freedom	Um inteiro que indica o número de graus de liberdade.
Cumulative	Um valor lógico que determina a forma da função. Se o cumulativo for TRUE, T.DIST retornará a função de distribuição cumulativa; se for FALSE, retornará a função de densidade de probabilidade.

## Valor retornado

A distribuição t do teste unilateral à esquerda do Student.

## Exemplo

```
EVALUATE { T.DIST(60, 1, TRUE) }
```

Retorna

[VALOR]

0,994695326367377

## Consulte também

[Função T.DIST.2T](#)

[Função T.DIST.RT](#)

[Função T.INV](#)

[Função T.INV.2t](#)

# T.DIST.2T

09/05/2020 • 2 minutes to read

Retorna a distribuição t bicaudal do Student.

## Sintaxe

```
T.DIST.2T(X,Deg_freedom)
```

### Parâmetros

TERMO	DEFINIÇÃO
X	O valor numérico no qual você deseja avaliar a distribuição.
Deg_freedom	Um inteiro que indica o número de graus de liberdade.

## Valor retornado

A distribuição t do Student bicaudal.

## Exemplo

```
EVALUATE { T.DIST.2T(1.959999998, 60) }
```

Retorna

[VALOR]

0,054644929975921

## Consulte também

[Função T.DIST](#)

[Função T.DIST.2T](#)

[Função T.INV](#)

[Função T.INV.2t](#)



# T.DIST.RT

09/05/2020 • 2 minutes to read

Retorna a distribuição t de cauda direita do Student.

## Sintaxe

```
T.DIST.RT(X,Deg_freedom)
```

### Parâmetros

TERMO	DEFINIÇÃO
X	O valor numérico no qual você deseja avaliar a distribuição.
Deg_freedom	Um inteiro que indica o número de graus de liberdade.

## Valor retornado

O teste unilateral à direita da distribuição t do Student.

## Exemplo

```
EVALUATE { T.DIST.RT(1.959999998, 60) }
```

Retorna

[VALOR]

0,0273224649879605

## Consulte também

[Função T.DIST](#)

[Função T.DIST.2T](#)

[Função T.INV](#)

[Função T.INV.2t](#)

# T.INV

09/05/2020 • 2 minutes to read

Retorna o inverso da distribuição t de cauda esquerda do Student.

## Sintaxe

```
T.INV(Probability,Deg_freedom)
```

### Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	A probabilidade associada à distribuição t do Student.
Deg_freedom	O número de graus de liberdade com o qual caracterizar a distribuição.

## Valor retornado

O inverso do teste unilateral à esquerda da distribuição t do Student.

## Exemplo

```
EVALUATE { T.INV(0.75, 2) }
```

Retorna

[VALOR]

0,816496580927726

## Consulte também

[Função T.INV.2T](#)

[Função T.DIST](#)

[Função T.DIST.2T](#)

[Função T.DIST.RT](#)

# T.INV.2T

09/05/2020 • 2 minutes to read

Retorna o inverso bicaudal da distribuição t do Student.

## Sintaxe

```
T.INV.2T(Probability,Deg_freedom)
```

### Parâmetros

TERMO	DEFINIÇÃO
Probabilidade	A probabilidade associada à distribuição t do Student.
Deg_freedom	O número de graus de liberdade com o qual caracterizar a distribuição.

## Valor retornado

O inverso bicaudal da distribuição t do Student.

## Exemplo

```
EVALUATE { T.INV.2T(0.546449, 60) }
```

Retorna

[VALOR]

0,606533075825759

## Consulte também

[Função T.INV](#)

[Função T.DIST](#)

[Função T.DIST.2T](#)

[Função T.DIST.2T](#)

# TAN

09/05/2020 • 2 minutes to read

Retorna a tangente do ângulo determinado.

## Sintaxe

TAN(number)

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. O ângulo em radianos para o qual você quer a tangente.

## Valor retornado

Retorna a tangente do ângulo determinado.

## Comentários

Se o argumento estiver em graus, multiplique-o por  $\text{PI()}/180$  ou use a função RADIANS para convertê-lo em radianos.

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=TAN(0,785)	Tangente de 0,785 radianos (0,99920)	0,99920
=TAN(45*PI()/180)	Tangente de 45 graus (1)	1
=TAN(RADIANS(45))	Tangente de 45 graus (1)	1

# TANH

08/05/2020 • 2 minutes to read

Retorna a tangente hiperbólica de um número.

## Sintaxe

TANH(number)

### Parâmetros

TERMO	DEFINIÇÃO
número	Obrigatório. Qualquer número real.

## Valor retornado

Retorna a tangente hiperbólica de um número.

## Comentários

A fórmula da tangente hiperbólica é:

$$\text{TANH}(z) = \frac{\text{SINH}(z)}{\text{COSH}(z)}$$

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=TANH(-2)	Tangente hiperbólica de -2 (-0,96403)	-0,964028
=TANH(0)	Tangente hiperbólica de 0 (0)	0
=TANH(0,5)	Tangente hiperbólica de 0,5 (0,462117)	0,462117

# TOPN

09/05/2020 • 2 minutes to read

Retorna as N linhas superiores da tabela especificada.

## Sintaxe

```
TOPN(<n_value>, <table>, <orderBy_expression>, [<order>[, <orderBy_expression>, [<order>]]...])
```

### Parâmetros

**n\_value**

Número de linhas a ser retornado. É qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

Confira a seção de comentários para entender quando o número de linhas retornado pode ser maior que *n\_value*.

Confira a seção de comentários para entender quando uma tabela vazia é retornada.

**table**

Qualquer expressão DAX que retorna uma tabela de dados da qual extrair as primeiras 'n' linhas.

**orderBy\_expression**

Qualquer expressão DAX em que o valor de resultado é usado para classificar a tabela e é avaliada para cada linha de *table*.

**order**

(Opcional) Um valor que especifica como classificar valores *orderBy\_expression* em ordem crescente ou decrescente:

valor	valor alternativo	Descrição
0 (zero)	FALSO	Classifica em ordem decrescente de valores de <i>order_by</i> .  É o valor padrão quando o parâmetro <i>order</i> é omitido.
1	VERDADEIRO	Classifica em ordem crescente de <i>order_by</i> .

## Valor retornado

Uma tabela com as primeiras N linhas de *table* ou uma tabela vazia se *n\_value* for 0 (zero) ou menos. As linhas não são necessariamente classificadas em nenhuma ordem específica.

## Comentários

- Se houver um empate nos valores de *order\_by* na *n*-ésima linha da tabela, todas as linhas vinculadas serão retornadas. Então, quando houver empates na *N*-ésima linha, a função poderá retornar mais de *n* linhas.

- Se n\_value for 0 (zero) ou menos, TOPN retornará uma tabela vazia.
- TOPN não garante nenhuma ordem de classificação para os resultados.

## Exemplo

O exemplo a seguir cria uma medida com as vendas dos dez produtos mais vendidos.

```
=SUMX(TOPN(10, SUMMARIZE(Product, [ProductKey], "TotalSales",
SUMX(RELATED(InternetSales_USD[SalesAmount_USD]), InternetSales_USD[SalesAmount_USD]) +
SUMX(RELATED(ResellerSales_USD[SalesAmount_USD]), ResellerSales_USD[SalesAmount_USD])))
```

# VAR.S

09/05/2020 • 2 minutes to read

Retorna a variância de uma amostra da população.

## Sintaxe

```
VAR.S(<columnName>)
```

### Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.

## Valor retornado

Um número com a variância de uma amostra da população.

## Exceções

## Comentários

1. A função VAR.S supõe que a coluna se refere a uma amostra da população. Se os dados representarem a população inteira, calcule a variância usando a função VAR.P.

2. A VAR.S usa a seguinte fórmula:

$$\sum (x - \tilde{x})^2 / (n-1)$$

em que  $\tilde{x}$  é o valor médio de x para a amostra da população

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma medida que calcula a variância da coluna SalesAmount\_USD da tabela InternetSales\_USD para uma amostra da população.

```
=VAR.S(InternetSales_USD[SalesAmount_USD])
```



# VAR.P

08/05/2020 • 2 minutes to read

Retorna a variância da população inteira.

## Sintaxe

```
VAR.P(<columnName>)
```

### Parâmetros

TERMO	DEFINIÇÃO
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.

## Valor retornado

Um número com a variância da população inteira.

## Comentários

1. VAR.P presume que a coluna faz referência a toda a população. Se os dados representarem uma amostra da população, calcule a variância usando VARX.S.

2. VAR.P usa a seguinte fórmula:

$$\sum (x - \tilde{x})^2 / n$$

em que  $\tilde{x}$  é o valor médio de x para a população inteira

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma medida que estima a variância da coluna SalesAmount\_USD da tabela InternetSales\_USD, para toda a população.

```
=VAR.P(InternetSales_USD[SalesAmount_USD])
```

# VARX.S

09/05/2020 • 2 minutes to read

Retorna a variância de uma amostra da população.

## Sintaxe

```
VARX.S(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

## Valor retornado

Um número que representa a variância de uma amostra da população.

## Exceções

## Comentários

1. A VARX.S avalia *expression* para cada linha de *table* e retorna a variância de *expression* – supondo que *table* faça referência a uma amostra da população. Se *table* representar uma população inteira, você deverá calcular a variância usando a função VARX.P.

2. A VAR.S usa a seguinte fórmula:

$$\sum (x - \tilde{x})^2 / (n - 1)$$

em que  $\tilde{x}$  é o valor médio de *x* para a amostra da população

e *n* é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma coluna calculada que estima a variância do preço unitário por produto para uma amostra da população quando a fórmula é usada na tabela Product.

```
=VARX.S(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# VARX.P

09/05/2020 • 2 minutes to read

Retorna a variância da população inteira.

## Sintaxe

```
VARX.P(<table>, <expression>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).

## Valor retornado

Um número com a variância da população inteira.

## Exceções

## Comentários

1. VARX.P avalia a <expression> para cada linha de <table> e retorna a variância de <expression>, supondo que <table> se refira à população inteira. Se <table> representar uma amostra da população, calcule a variância usando VARX.S.

2. VARX.P usa a seguinte fórmula:

$$\sum (x - \tilde{x})^2 / n$$

em que  $\tilde{x}$  é o valor médio de x para a população inteira

e n é o tamanho da população

3. As linhas em branco são filtradas de *columnName* e não são consideradas nos cálculos.

4. Um erro será retornado se *columnName* contiver menos de duas linhas que não estejam em branco

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir mostra a fórmula para uma coluna calculada que calcula a variância do preço unitário por produto, quando a fórmula é usada na tabela Product

```
=VARX.P(InternetSales_USD, InternetSales_USD[UnitPrice_USD] -
(InternetSales_USD[DiscountAmount_USD]/InternetSales_USD[OrderQuantity]))
```

# XIRR

08/05/2020 • 2 minutes to read

Retorna a taxa interna de retorno de um agendamento de fluxos de caixa que não é necessariamente periódico.

## Sintaxe

```
XIRR(<table>, <values>, <dates>, [guess])
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Uma tabela para a qual as expressões de valores e datas devem ser calculadas.
valores	Uma expressão que retorna o valor do fluxo de caixa para cada linha da tabela.
datas	Uma expressão que retorna a data do fluxo de caixa para cada linha da tabela.
guess	(Opcional) Uma estimativa inicial para a taxa interna de retorno. Se ele for omitido, a estimativa padrão de 0,1 será usada.

## Valor retornado

Taxa interna de retorno para as entradas especificadas. Se o cálculo não retornar um resultado válido, um erro será retornado.

## Comentários

O valor é calculado como a taxa que atende à seguinte função:

$$\sum_{j=1}^N \frac{P_j}{(1+rate)^{\frac{d_j-d_1}{365}}}$$

Where  $P_j$  is the j-th payment,  $d_j$  is the j-th payment date, and  $d_1$  is the first payment date

A série de valores do fluxo de caixa precisa conter pelo menos um número positivo e um número negativo.

## Exemplo

O seguinte exemplo calcula a taxa interna de retorno da tabela CashFlows:

```
Rate of return := XIRR(CashFlows, [Payment], [Date])
```

DATA	PAGAMENTO
01/01/2014	-10.000
01/03/2014	2.750
30/10/2014	4.250
15/2/2015	3.250
4/1/2015	2.750

Taxa de retorno = 37,49%

# XNPV

09/05/2020 • 2 minutes to read

Retorna o valor atual de um agendamento de fluxos de caixa que não é necessariamente periódico.

## Sintaxe

```
XNPV(<table>, <values>, <dates>, <rate>)
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Uma tabela para a qual as expressões de valores e datas devem ser calculadas.
valores	Uma expressão que retorna o valor do fluxo de caixa para cada linha da tabela.
datas	Uma expressão que retorna a data do fluxo de caixa para cada linha da tabela.
rate	A taxa de desconto a ser aplicada ao fluxo de caixa para cada linha da tabela.

## Valor retornado

Valor líquido atual.

## Comentários

O valor é calculado como a soma a seguir:

$$\sum_{j=1}^N \frac{P_j}{(1+rate)^{\frac{d_j-d_1}{365}}}$$

Where  $P_j$  is the  $j$ -th payment,  $d_j$  is the  $j$ -th payment date, and  $d_1$  is the first payment date

A série de valores do fluxo de caixa precisa conter pelo menos um número positivo e um número negativo.

## Exemplo

Confira a seguir o cálculo do valor atual da tabela CashFlows:

```
Present value := XNPV(CashFlows, [Payment], [Date], 0.09)
```



DATA	PAGAMENTO
01/01/2014	-10.000
01/03/2014	2.750
30/10/2014	4.250
15/2/2015	3.250
4/1/2015	2.750

Valor atual = 2.086,65

# Funções de texto

21/05/2020 • 3 minutes to read

O DAX (Data Analysis Expressions) inclui um conjunto de funções de texto com base na biblioteca de funções de cadeia de caracteres no Excel, mas que foram modificadas para funcionar com tabelas e colunas em modelos de tabela. Esta seção descreve as funções de texto disponíveis na linguagem DAX.

## Nesta categoria

FUNÇÃO	DESCRIÇÃO
<a href="#">BLANK</a>	Retorna um espaço em branco.
<a href="#">CODE</a>	Retorna um código numérico para o primeiro caractere em uma cadeia de texto.
<a href="#">CONCATENATE</a>	Une duas cadeias de texto em uma.
<a href="#">CONCATENATEX</a>	Concatena o resultado de uma expressão avaliada para cada linha de uma tabela.
<a href="#">CONTAINSSTRING</a>	Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.
<a href="#">CONTAINSSTRINGEXACT</a>	Retorna TRUE ou FALSE, indicando se uma cadeia de caracteres contém outra cadeia de caracteres.
<a href="#">EXACT</a>	Compara duas cadeias de texto e retorna TRUE se elas são exatamente iguais; caso contrário, FALSE.
<a href="#">FIND</a>	Retorna a posição inicial de uma cadeia de texto em outra cadeia de texto.
<a href="#">FIXED</a>	Arredonda um número para o número especificado de decimais e retorna o resultado como texto.
<a href="#">FORMAT</a>	Converte um valor em texto de acordo com o formato especificado.
<a href="#">LEFT</a>	Retorna o número especificado de caracteres do início de uma cadeia de texto.
<a href="#">LEN</a>	Retorna o número de caracteres em uma cadeia de texto.
<a href="#">LOWER</a>	Converte todas as letras de uma cadeia de texto em minúsculas.
<a href="#">MID</a>	Retorna uma cadeia de caracteres do meio de uma cadeia de texto, dados um ponto inicial e um comprimento.

FUNÇÃO	DESCRIÇÃO
REPLACE	REPLACE substitui parte de uma cadeia de texto, com base no número de caracteres que você especificar, com uma cadeia de texto diferente.
REPT	Repete um texto um determinado número de vezes.
RIGHT	RIGHT retorna o último caractere ou caracteres em uma cadeia de texto, com base no número de caracteres que você especificar.
SEARCH	Retorna o número do caractere no qual um caractere específico ou uma cadeia de texto é encontrada pela primeira vez, lendo da esquerda para a direita.
SUBSTITUTE	Substitui o texto original pelo novo texto em uma cadeia de texto.
TRIM	Remove todos os espaços de um valor de texto, exceto espaços simples entre palavras.
UNICHAR	Retorna o caractere Unicode referenciado pelo valor numérico.
UNICODE	Retorna o código numérico correspondente ao primeiro caractere da cadeia de caracteres de texto.
UPPER	Converte uma cadeia de texto em letras maiúsculas.
VALUE	Converte em número uma cadeia de texto que representa um número.

# BLANK

09/05/2020 • 2 minutes to read

Retorna um espaço em branco.

## Sintaxe

```
BLANK()
```

## Valor retornado

Um espaço em branco.

## Comentários

Os espaços em branco não são equivalentes a nulos. O DAX usa espaços em branco para os nulos do banco de dados e para células em branco no Excel.

Algumas funções DAX tratam as células em branco de um modo um pouco diferente do Microsoft Excel. Espaços em branco e cadeias de caracteres vazias ("" ) nem sempre são equivalentes, mas algumas operações podem tratá-los se fossem.

## Exemplo

O exemplo a seguir ilustra como você pode trabalhar com espaços em branco em fórmulas. A fórmula calcula a taxa de vendas entre os revendedores e os canais da Internet. No entanto, antes de tentar calcular a proporção, o denominador deve ser verificado em busca de valores zero. Se o denominador for zero, um valor em branco deverá ser retornado; caso contrário, a taxa será calculada.

```
=IF(SUM(InternetSales_USD[SalesAmount_USD])= 0 , BLANK() ,
SUM(ResellerSales_USD[SalesAmount_USD])/SUM(InternetSales_USD[SalesAmount_USD]))
```

A tabela mostra os resultados esperados quando esta fórmula é usada para criar uma Tabela Dinâmica.

PROPORÇÃO DE VENDAS DO REVENDEDOR PARA A INTERNET	RÓTULOS DE COLUNA			
Rótulos de linha	Acessórios	Bicicletas	Vestuário	Total Geral
2005		2,65		2,89
2006		3,33		4,03
2007	1,04	2,92	6,63	3,51
2008	0,41	1,53	2,00	1,71

PROPORÇÃO DE VENDAS DO REVENDEDOR PARA A INTERNET	RÓTULOS DE COLUNA			
Total Geral	0,83	2,51	5,45	2,94

Na fonte de dados original, a coluna avaliada pela função BLANK poderia ter incluído texto, cadeias de caracteres vazias ou nulos. Se a fonte de dados original era um banco de dados SQL Server, nulos e cadeias de caracteres vazias são tipos diferentes de dados. No entanto, para essa operação, uma conversão implícita de tipo é executada e o DAX as trata como a mesma.

## Consulte também

[Funções de texto \(DAX\)](#)

[Função ISBLANK \(DAX\)](#)

# CODE

09/05/2020 • 2 minutes to read

Retorna um código numérico para o primeiro caractere em uma cadeia de texto. O código retornado corresponde ao conjunto de caracteres usado pelo seu computador.

AMBIENTE OPERACIONAL	CONJUNTO DE CARACTERES
Macintosh	Conjunto de caracteres Macintosh
Windows	ANSI

## Sintaxe

CODE(text)

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto para o qual você deseja o código do primeiro caractere.

## Valor retornado

Um código numérico para o primeiro caractere em uma cadeia de texto.

## Exemplo

FÓRMULA	DESCRIÇÃO	RESULTADO
=CODE("A")	Exibe o código numérico para A	65
=CODE("!")	Exibe o código numérico para !	33

# COMBINEVALUES

09/05/2020 • 4 minutes to read

A função COMBINEVALUES une duas ou mais cadeias de caracteres de texto em uma cadeia de caracteres de texto. A principal finalidade dessa função é oferecer compatibilidade com relacionamentos multicoluna em modelos DirectQuery. Consulte [Comentários](#) para obter detalhes.

## Sintaxe

```
COMBINEVALUES(<delimiter>, <expression>, <expression>[, <expression>]...)
```

### Parâmetros

TERMO	DEFINIÇÃO
delimitador	Um separador a ser usado durante a concatenação. Deve ser um valor constante.
expressão	Uma expressão DAX cujo valor será unido em uma única cadeia de texto.

## Valor retornado

A cadeia de caracteres concatenada.

## Comentários

A função COMBINEVALUES pressupõe, mas não valida, que, quando os valores de entrada são diferentes, as cadeias de caracteres de saída também são diferentes. Com base nessa suposição, quando COMBINEVALUES é usado para criar colunas calculadas para criar uma relação que une várias colunas de duas tabelas DirectQuery, uma condição de junção otimizada é gerada no momento da consulta. Por exemplo, se os usuários desejarem criar uma relação entre Table1 (Column1, Column2) e Table2 (Column1, Column2), eles poderão criar duas colunas calculadas, uma em cada tabela, como:

```
Table1[CalcColumn] = COMBINEVALUES(",", Table1[Column1], Table1[Column2])
```

e

```
Table2[CalcColumn] = COMBINEVALUES(",", Table2[Column1], Table2[Column2]) ,
```

Em seguida, crie uma relação entre `Table1[CalcColumn]` e `Table2[CalcColumn]`. Ao contrário de outras funções e operadores DAX, que são traduzidos literalmente para as funções e operadores SQL correspondentes, a relação acima gera um predicado de junção SQL como:

```
(Table1.Column1 = Table2.Column1 OR Table1.Column1 IS NULL AND Table2.Column1 IS NULL)
```

e

```
(Table1.Column2 = Table2.Column2 OR Table1.Column2 IS NULL AND Table2.Column2 IS NULL) .
```

O predicado de junção pode potencialmente proporcionar um desempenho de consulta muito melhor que um que envolva operadores e funções SQL complexos.

A função COMBINEVALUES baseia-se em usuários para escolher o delimitador apropriado para garantir que combinações exclusivas de valores de entrada produzam cadeias de caracteres de saída distintas, mas não validam que a suposição é verdadeira. Por exemplo, se os usuários escolherem " | " como o delimitador, mas uma linha em Table1 tiver Table1[Column1] = " | " e Table2 [Column2] = " " , enquanto uma linha em Table2 tiver Table2[Column1] = " " e Table2[Column2] = " | " , as duas saídas concatenadas serão a mesma " | | " , o que parece indicar que há uma correspondência das duas linhas na operação de junção. As duas linhas não serão unidas se ambas as tabelas forem da mesma origem do DirectQuery, embora sejam unidas em conjunto se ambas as tabelas forem importadas.

## Exemplo

A seguinte consulta DAX:

```
EVALUATE DISTINCT(SELECTCOLUMNS(DimDate, "Month", COMBINEVALUES(", ", [MonthName], [CalendarYear])))
```

Retorna a seguinte tabela de coluna única:

[MÊS]
Janeiro de 2007
Fevereiro de 2007
Março de 2007
Abril de 2007
Mai de 2007
Junho de 2007
Julho de 2007
Agosto de 2007
Setembro de 2007
Outubro de 2007
Novembro de 2007
Dezembro de 2007
Janeiro de 2008
Janeiro de 2008
Fevereiro de 2008
Março de 2008
Abril de 2008



[MÊS]
Maio de 2008
Junho de 2008
Julho de 2008
Agosto de 2008
Setembro de 2008
Outubro de 2008
Novembro de 2008
Dezembro de 2008

# CONCATENATE

08/05/2020 • 5 minutes to read

Une duas cadeias de texto em uma.

## Sintaxe

```
CONCATENATE(<text1>, <text2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
text1, text2	As cadeias de caracteres de texto a serem unidas em uma única cadeia de caracteres de texto. As cadeias de caracteres podem incluir texto ou números.  Você também pode usar referências de coluna.

## Valor retornado

A cadeia de caracteres concatenada.

## Comentários

A função CONCATENATE une duas cadeias de caracteres de texto em uma cadeia de caracteres de texto. Os itens unidos podem ser texto, números ou valores booleanos representados como texto ou uma combinação desses itens. Você também poderá usar uma referência de coluna se a coluna contiver valores apropriados.

A função CONCATENATE no DAX aceita apenas dois argumentos, enquanto a função CONCATENATE do Excel aceita até 255 argumentos. Se você precisar concatenar várias colunas, poderá criar uma série de cálculos ou, ainda melhor, usar o operador de concatenação ( & ) para unir todas elas em uma expressão mais simples.

Se você quiser usar cadeias de caracteres de texto diretamente, em vez de usar uma referência de coluna, deverá colocar cada cadeia de caracteres entre aspas duplas.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo: Concatenação de literais

### Descrição

A fórmula de exemplo cria um novo valor de cadeia de caracteres combinando dois valores de cadeia de caracteres que você fornece como argumentos.

### Código

```
=CONCATENATE("Hello ", "World")
```

## Exemplo: Concatenação de cadeias de caracteres em colunas

## Descrição

A fórmula de exemplo retorna o nome completo do cliente, conforme listado em um catálogo de telefones. Observe como uma função aninhada é usada como o segundo argumento. Essa é uma maneira de concatenar várias cadeias de caracteres quando você tem mais de dois valores que deseja usar como argumentos.

## Código

```
=CONCATENATE(Customer[LastName], CONCATENATE(", ", Customer[FirstName]))
```

## Exemplo: Concatenação condicional de cadeias de caracteres em colunas

### Descrição

A fórmula de exemplo cria uma nova coluna calculada na tabela Customer com o nome completo do cliente como uma combinação de primeiro nome, primeira letra do segundo nome e sobrenome. Se não houver um segundo nome, o sobrenome virá diretamente após o primeiro nome. Se houver um segundo nome, somente a primeira letra do segundo nome será usada e será seguida por um ponto.

## Código

```
=CONCATENATE([FirstName]&" ", CONCATENATE(IF(LEN([MiddleName])>1, LEFT([MiddleName],1)&" ", ""), [LastName]))
```

### Comentários

Essa fórmula usa funções aninhadas CONCATENATE e IF, juntamente com o operador de e comercial (&) para concatenar condicionalmente três valores de cadeia de caracteres e adicionar espaços como separadores.

## Exemplo: Concatenação de colunas com tipos de dados diferentes

O exemplo a seguir demonstra como concatenar valores em colunas que têm tipos de dados diferentes. Se o valor que você está concatenando for numérico, ele será convertido implicitamente em texto. Se ambos os valores forem numéricos, eles serão convertidos em texto e concatenados como se fossem cadeias de caracteres.

DESCRIÇÃO DO PRODUTO	ABREVIÇÃO DO PRODUTO (COLUNA 1 DA CHAVE COMPOSTA)	NÚMERO DO PRODUTO (COLUNA 2 DA CHAVE COMPOSTA)	NOVA COLUNA DE CHAVE GERADA
Mountain bike	MTN	40	MTN40
Mountain bike	MTN	42	MTN42

## Código

```
=CONCATENATE('Products'[Product abbreviation], 'Products'[Product number])
```

### Comentários

A função CONCATENATE no DAX aceita apenas dois argumentos, enquanto a função CONCATENATE do Excel aceita até 255 argumentos. Se você precisar adicionar mais argumentos, poderá usar o operador de e comercial (&). Por exemplo, a fórmula a seguir produz os resultados, MTN-40 e MTN-42.

```
=[Product abbreviation] & "-" & [Product number]
```

## Consulte também



# CONCATENATEX

08/05/2020 • 2 minutes to read

Concatena o resultado de uma expressão avaliada para cada linha de uma tabela.

## Sintaxe

```
CONCATENATEX(<table>, <expression>, [delimiter])
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	A tabela que contém as linhas para as quais a expressão será avaliada.
expressão	A expressão a ser avaliada para cada linha da tabela.
delimitador	(Opcional) Um separador a ser usado durante a concatenação.

## Valor retornado

Uma cadeia de texto.

## Comentários

Essa função usa como seu primeiro argumento uma tabela ou uma expressão que retorna uma tabela. O segundo argumento é uma coluna que contém os valores que você deseja concatenar ou uma expressão que retorna um valor.

## Exemplo

Tabela Funcionários

FIRSTNAME	LASTNAME
Alan	Brewer
Michael	Blythe

```
CONCATENATEX(Employees, [FirstName] & " " & [LastName], ",")
```

Retorna "Alan Brewer, Michael Blythe"

# EXACT

09/05/2020 • 2 minutes to read

Compara duas cadeias de texto e retorna TRUE se elas são exatamente iguais; caso contrário, FALSE. EXACT diferencia maiúsculas de minúsculas, mas ignora as diferenças de formatação. Você pode usar EXACT para testar o texto que está sendo inserido em um documento.

## Sintaxe

```
EXACT(<text1>,<text2>)
```

### Parâmetros

TERMO	DEFINIÇÃO
text1	A primeira cadeia de texto ou coluna que contém texto.
text2	A segunda cadeia de texto ou coluna que contém texto.

## Valor da propriedade/valor retornado

True ou false. (Booleano)

## Exemplo

A fórmula a seguir verifica o valor de Coluna1 para a linha atual em relação ao valor de Coluna2 para a linha atual e retornará TRUE se elas forem iguais, mas retornará FALSE se forem diferentes.

```
=EXACT([Column1],[Column2])
```

## Consulte também

[Funções de texto \(DAX\)](#)

# FIND

09/05/2020 • 2 minutes to read

Retorna a posição inicial de uma cadeia de texto em outra cadeia de texto. FIND diferencia maiúsculas de minúsculas.

## Sintaxe

```
FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

### Parâmetros

TERMO	DEFINIÇÃO
find_text	O texto que você deseja encontrar. Use aspas duplas (texto vazio) para corresponder ao primeiro caractere em <b>within_text</b> .
within_text	O texto que contém o texto que você deseja encontrar.
start_num	(opcional) O caractere no qual iniciar a pesquisa; se omitido, <b>start_num</b> = 1. O primeiro caractere em <b>within_text</b> é o número de caracteres 1.
NotFoundValue	(opcional) O valor que deve ser retornado quando a operação não encontrar uma substring correspondente, normalmente 0, -1 ou BLANK().

## Valor da propriedade/valor retornado

Número que mostra o ponto inicial da cadeia de texto que você deseja localizar.

## Comentários

Enquanto o Microsoft Excel tem várias versões da função FIND para acomodar as linguagens SBCS (conjunto de caracteres de um byte) e DBCS (conjunto de caracteres de byte duplo), o DAX usa Unicode e conta cada caractere da mesma forma; portanto, você não precisa usar uma versão diferente dependendo do tipo de caractere.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre as diferenças semânticas do modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

FIND não dá suporte a curingas. Para usar curingas, use [SEARCH](#).

## Exemplo

A fórmula a seguir localiza a posição da primeira letra da designação do produto, BMX, na cadeia de caracteres que contém a descrição do produto.

```
=FIND("BMX","line of BMX racing goods")
```

## Consulte também

[Funções de texto \(DAX\)](#)



# FIXED

09/05/2020 • 2 minutes to read

Arredonda um número para o número especificado de decimais e retorna o resultado como texto. Você pode especificar que o resultado seja retornado com ou sem vírgulas.

## Sintaxe

```
FIXED(<number>, <decimals>, <no_commas>)
```

### Parâmetros

TERMO	DEFINIÇÃO
número	O número que você deseja arredondar e converter em texto ou uma coluna que contém um número.
decimals	(opcional) O número de dígitos à direita do ponto decimal; se omitido, 2.
no_commas	(opcional) Um valor lógico: se for 1, não exibir vírgulas no texto retornado; se for 0 ou omitido, exibir vírgulas no texto retornado.

## Valor da propriedade/valor retornado

Um número representado como texto.

## Comentários

Se o valor usado para o parâmetro **decimals** for negativo, **number** será arredondado à esquerda do ponto decimal.

Se você omitir **decimals**, será considerado 2.

Se **no\_commas** for 0 ou for omitido, o texto retornado incluirá vírgulas como de costume.

A principal diferença entre a formatação de uma célula que contém um número usando um comando e a formatação de um número diretamente com a função FIXED é que esta função converte seu resultado em texto. Um número formatado com um comando no menu de formatação ainda é um número.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

## Exemplo

O exemplo a seguir obtém o valor numérico da linha atual na coluna, PctCost, e a retorna como texto com quatro casas decimais e sem vírgulas.

```
=FIXED([PctCost],3,1)
```

Os números nunca podem ter mais de 15 dígitos significativos, mas os decimais podem ter até 127.

## Consulte também

[Funções de texto \(DAX\)](#)

[Funções matemáticas e trigonométricas \(DAX\)](#)

# FORMAT

08/05/2020 • 3 minutes to read

Converte um valor em texto de acordo com o formato especificado.

## Sintaxe

```
FORMAT(<value>, <format_string>)
```

### Parâmetros

TERMO	DEFINIÇÃO
valor	Um valor ou uma expressão que é avaliada como um único valor.
format_string	Uma cadeia de caracteres com o modelo de formatação.

## Valor retornado

Uma cadeia de caracteres contendo **valor** formatada conforme definido por **format\_string**.

### IMPORTANT

Se *value* for BLANK(), a função retornará uma cadeia de caracteres vazia.

Se *format\_string* for BLANK(), o valor será formatado com um formato "Número Geral" ou "Data Geral" (de acordo com tipo **valor**).

## Comentários

Para obter informações sobre como usar o parâmetro **format\_string**, confira o tópico apropriado listado abaixo:

PARA FORMATAR	SIGA ESTAS INSTRUÇÕES
Números	Use formatos numéricos predefinidos ou crie formatos numéricos definidos pelo usuário.
Datas e horas	Use formatos de data/hora predefinidos ou crie formatos de data/hora definidos pelo usuário.

Todas as cadeias de caracteres de formatação predefinidas usam a localidade do usuário atual ao formatar o resultado.

### Caution

As cadeias de caracteres de formato com suporte como um argumento para a função DAX FORMAT baseiam-se nas cadeias de caracteres de formato usadas pelo Visual Basic (Automação OLE), não nas cadeias de caracteres de formato usadas pelo .NET Framework. Portanto, você poderá obter resultados inesperados ou um erro se o argumento não corresponder a nenhuma cadeia de caracteres de formato definida. Por exemplo, não há suporte para "p" como uma abreviação de "Porcentagem". As cadeias de caracteres fornecidas como um argumento para a

função FORMAT que não estão incluídas na lista de cadeias de caracteres de formato predefinidas são manipuladas como parte de uma cadeia de caracteres de formato personalizado ou como um literal de cadeia de caracteres.

Não há suporte para essa função DAX para uso no modo DirectQuery. Para obter mais informações sobre limitações nos modelos do DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219172>.

## Consulte também

[Formatos numéricos predefinidos para a função FORMAT](#)

[Formatos numéricos personalizados para a função FORMAT](#)

[Formatos de data e hora predefinidos para a função FORMAT](#)

[Formatos de data e hora personalizados para a função FORMAT](#)

[Função VALUE \(DAX\)](#)

# Formatos numéricos predefinidos para a função FORMAT

09/05/2020 • 4 minutes to read

A tabela a seguir identifica os nomes de formatos numéricos predefinidos. Eles podem ser usados pelo nome como o argumento de estilo para a função Format.

ESPECIFICAÇÃO DE FORMATO	DESCRIÇÃO
"General Number"	Exibe o número sem separadores de milhar.
"Currency"	Exibe o número com separadores de milhar, se apropriado; exibe dois dígitos à direita do separador decimal. A saída se baseia nas configurações de localidade do sistema.
"Fixed"	Exibe pelo menos um dígito à esquerda e dois dígitos à direita do separador decimal.
"Standard"	Exibe o número com separador de milhar, pelo menos um dígito à esquerda e dois dígitos à direita do separador decimal.
"Percent"	Exibe o número multiplicado por 100 com um sinal de porcentagem (%) imediatamente à direita; sempre exibe dois dígitos à direita do separador decimal.
"Scientific"	Usa notação científica padrão, fornecendo dois dígitos significativos.
"Yes/No"	Exibe Não se o número é 0; caso contrário, exibe Sim.
"True/False"	Exibe False se o número é 0; caso contrário, exibe True.
"On/Off"	Exibe Desligado o número é 0; caso contrário, exibe Ligado.

## Comentários

Observe que as cadeias de caracteres de formato se baseiam em Visual Basic (Automação OLE) e, portanto, podem ter um comportamento ligeiramente diferente das cadeias de caracteres de formato usadas pelo .NET Framework. O uso de abreviações como "P" e "x" não é compatível. Qualquer outra cadeia de caracteres que você fornecer como um argumento para a função FORMAT será interpretada como definindo um formato personalizado.

### IMPORTANT

Se *value* for BLANK(), a função retornará uma cadeia de caracteres vazia.

Se *format\_string* for BLANK(), o valor será formatado com um formato de "Número Geral".

## Exemplo

Os exemplos a seguir mostram o uso de diferentes cadeias de caracteres de formatação predefinidas para formatar um valor numérico.

```
FORMAT(12345.67, "General Number")
FORMAT(12345.67, "Currency")
FORMAT(12345.67, "Fixed")
FORMAT(12345.67, "Standard")
FORMAT(12345.67, "Percent")
FORMAT(12345.67, "Scientific")
```

As expressões acima retornam os seguintes resultados:

**12345.67** "Número Geral" exibe o número sem formatação.

**\$12345.67** "Moeda" exibe o número com a formatação de moeda da sua localidade. O exemplo aqui mostra a formatação padrão de moeda dos Estados Unidos.

**12345.67** "Fixo" exibe o número com separador de milhar, pelo menos um dígito à esquerda do separador decimal e dois dígitos à direita do separador decimal.

**12,345.67** "Standard" exibe o número com separador de milhar, pelo menos um dígito à esquerda do separador decimal e dois dígitos à direita do separador decimal e inclui separadores de milhar. O exemplo aqui mostra a formatação padrão de número dos Estados Unidos.

**1,234,567.00 %** "Percentual" exibe o número como um percentual (multiplicado por 100) com formatação e o sinal de porcentagem à direita do número separado por um único espaço.

**1.23E+04** "Científico" exibe o número em notação científica com dois dígitos decimais.

## Consulte também

[Função FORMAT \(DAX\)](#)

[Formatos de data e hora predefinidos para a função FORMAT](#)

[Formatos numéricos personalizados para a função FORMAT](#)

# Formatos numéricos personalizados para a função FORMAT

09/05/2020 • 11 minutes to read

Este artigo descreve os formatos personalizados e definidos pelo usuário para valores numéricos em uma expressão FORMAT.

## Formatos diferentes para valores numéricos diferentes

Uma expressão de formato definida pelo usuário para números pode ter de uma a quatro seções separadas por ponto e vírgula. Se o argumento de formato contiver um dos formatos numéricos nomeados, apenas uma seção será permitida.

SE VOCÊ USAR	O RESULTADO É
Apenas uma seção	A expressão de formato se aplica a todos os valores.
Duas seções	A primeira seção é aplicada a valores positivos e zeros; a segunda a valores negativos.
Três seções	A primeira seção é aplicada a valores positivos, a segunda a valores negativos e a terceira a zeros.
Quatro seções	A primeira seção é aplicada a valores positivos, a segunda a valores negativos, a terceira a zeros e a quarta a valores nulos.

"\$#,##0;(\$#,##0)"

Caso inclua ponto e vírgula sem nada entre ele, a seção ausente será definida usando o formato do valor positivo. Por exemplo, o formato a seguir exibe valores positivos e negativos usando o formato da primeira seção e exibe "Zero" se o valor for zero.

"\$#,##0

## Formatos numéricos definidos pelo usuário

A tabela a seguir identifica os caracteres que você pode usar para criar formatos de número definidos pelo usuário.

CARACTERE	DESCRIÇÃO
Nenhum	Exibe o número sem formatação.

CARACTERE	DESCRIÇÃO
(0)	Espaço reservado de dígito. Exibe um dígito ou um zero. Se a expressão tiver um dígito na posição em que 0 aparece na cadeia de caracteres de formato, exiba-o. Caso contrário, exiba um zero nessa posição. Se o número tiver menos dígitos do que zeros (em ambos os lados do decimal) na expressão de formato, exiba zeros à esquerda ou à direita. Se o número tiver mais dígitos à direita do separador decimal do que zeros à direita do separador decimal na expressão do formato, arredonde o número para a mesma quantidade de casas decimais que há de zeros. Se o número tiver mais dígitos à esquerda do separador decimal do que zeros à esquerda do separador decimal na expressão do formato, exibe os dígitos extras sem modificação.
( # )	Espaço reservado de dígito. Exibe um dígito ou nada. Se a expressão tiver um dígito na posição em que # aparece na cadeia de caracteres de formato, exibe-o; caso contrário, não exibe nada nessa posição. Esse símbolo funciona como o espaço reservado para o dígito 0, exceto pelo fato de que os zeros à esquerda e à direita não serão exibidos se o número tiver menos ou a mesma quantidade de dígitos do que os caracteres # em ambos os lados do separador decimal na expressão de formato.
( . )	Espaço reservado de decimal. Em algumas localidades, uma vírgula é usada como o separador decimal. O espaço reservado decimal determina quantos dígitos são exibidos à esquerda e à direita do separador decimal. Se a expressão do formato contiver apenas sinais de número à esquerda desse símbolo, números menores que 1 começarão com um separador decimal. Para mostrar um zero à esquerda exibido com números fracionários, use 0 como o espaço reservado do primeiro dígito à esquerda do separador decimal. O caractere real usado como espaço reservado decimal na saída formatada depende do Formato de Número reconhecido pelo seu sistema.
( % )	Espaço reservado de porcentagem. A expressão é multiplicada por 100. O caractere de porcentagem ( % ) é inserido na posição em que ele aparece na cadeia de caracteres de formato.



CARACTERE	DESCRIÇÃO
(, )	Separador de milhar. Em algumas localidades, um ponto é usado como um separador de milhar. O separador de milhar separa milhares de centenas em um número que tem quatro ou mais casas à esquerda do separador decimal. O uso padrão do separador de milhar é especificado se o formato contiver um separador de milhar entre os espaços reservados para dígitos (0 ou # ). Dois separadores de milhar adjacentes ou um separador de milhar imediatamente à esquerda do separador decimal (se um decimal for ou não especificado) significa "dimensionar o número dividindo-o por 1.000, arredondando conforme necessário". Por exemplo, você pode usar a cadeia de caracteres de formato "##0,," para representar 100 milhões como 100. Os números inferiores a 1 milhão são exibidos como 0. Dois separadores de milhar adjacentes em qualquer posição que não seja imediatamente à esquerda do separador decimal são tratados simplesmente como especificação do uso de um separador de milhar. O caractere real usado como o separador de milhar na saída formatada depende do Formato de Número reconhecido pelo seu sistema.
(: )	Separador de hora. Em algumas localidades, outros caracteres podem ser usados para representar o separador de hora. O separador de horas separa hora, minutos e segundos quando os valores de hora são formatados. O caractere real usado como o separador de hora na saída formatada é determinado pelas configurações do sistema.
(/ )	Separador de data. Em algumas localidades, outros caracteres podem ser usados para representar o separador de data. O separador de data separa o dia, o mês e o ano quando os valores de data são formatados. O caractere real usado como o separador de data na saída formatada é determinado pelas configurações do sistema.
(E- E+ e- e+ )	Formato científico. Se a expressão do formato contiver pelo menos um espaço reservado para dígito (0 ou # ) à direita de E-, E+, e- ou e+, o número será exibido em formato científico e E ou e será inserido entre o número e seu expoente. O número de espaços reservados de dígito à direita determina o número de dígitos no expoente. Use E- ou e- para colocar um sinal de menos ao lado de expoentes negativos. Use E+ ou e+ para colocar um sinal de menos ao lado de expoentes negativos e um sinal de mais ao lado de expoentes positivos.
- + \$ ( )	Exibe um caractere literal. Para exibir um caractere diferente dos listados, preceda-o com uma barra invertida ( \ ) ou coloque-o entre aspas duplas ( " ").

CARACTERE	DESCRIÇÃO
( \ )	Exibe o próximo caractere na cadeia de caracteres de formato. Para exibir um caractere que tenha um significado especial como um caractere literal, preceda-o com uma barra invertida ( \ ). A barra invertida em si não é exibida. Usar uma barra invertida é como colocar o próximo caractere entre aspas duplas. Para exibir uma barra invertida, use duas barras invertidas ( \ \ ). Exemplos de caracteres que não podem ser exibidos como caracteres literais são os caracteres de formatação de data e hora (a, c, d, h, m, n, p, q, s, t, w, y, / e :), os caracteres de formatação numérica (#, 0, %, E, e, vírgula e ponto) e os caracteres de formatação de cadeia de caracteres (@, &, <, > e !).
("ABC")	Exibe a cadeia de caracteres entre aspas duplas (" ").

## Comentários

Se você incluir ponto e vírgula sem nada entre ele, a seção ausente será impressa usando o formato do valor positivo.

## Consulte também

[Função FORMAT \(DAX\)](#)

[Formatos numéricos predefinidos para a função FORMAT](#)

[Formatos de data e hora personalizados para a função FORMAT](#)

# Formatos de data e hora predefinidos para a função FORMAT

08/05/2020 • 2 minutes to read

A tabela a seguir identifica os nomes dos formatos de data e hora predefinidos. Se você usar cadeias de caracteres diferentes dessas predefinidas, elas serão interpretadas como um formato de data e hora personalizado.

ESPECIFICAÇÃO DE FORMATO	DESCRIÇÃO
"General Date"	Exibe uma data e/ou hora. Por exemplo, 3/12/2008 11:07:31 AM. A exibição de data é determinada pelo valor de cultura atual de seu aplicativo.
"Long Date" OU "Medium Date"	Exibe uma data de acordo com o formato de data longa de sua cultura atual. Por exemplo, quarta-feira, 12 de março de 2008.
"Short Date"	Exibe uma data usando o formato de data curta de sua cultura atual. Por exemplo, 3/12/2008.
"Long Time" OU	Exibe uma hora usando o formato de hora longo da cultura atual. Normalmente, inclui horas, minutos, segundos. Por exemplo, 11:07:31 AM.
"Medium Time"	Exibe uma hora no formato de 12 horas. Por exemplo, 11:07 AM.
"Short Time"	Exibe uma hora no formato de 24 horas. Por exemplo, 11:07.

## Comentários

As cadeias de caracteres de formatação se baseiam no Visual Basic (Automação OLE) e não nas cadeias de caracteres de formatação do .NET Framework. Portanto, os resultados podem ser um pouco diferentes do esperado das cadeias de caracteres de formato do .NET. Observe que não há suporte para abreviações como "D" para datas longas e "t" para horas curtas.

### IMPORTANT

Se *value* for BLANK(), a função retornará uma cadeia de caracteres vazia.

Se *format\_string* for BLANK(), o valor será formatado com um formato de "Data Geral".

## Consulte também

[Formatos de data e hora personalizados para a função FORMAT](#)

# Formatos de data e hora personalizados para a função FORMAT

08/05/2020 • 8 minutes to read

As tabelas a seguir descrevem os caracteres usados para criar formatos de data/hora definidos pelo usuário.

## Formatos de data/hora definidos pelo usuário

CARACTERE	DESCRIÇÃO
(:)	Separador de hora. Em algumas localidades, outros caracteres podem ser usados para representar o separador de hora. O separador de horas separa hora, minutos e segundos quando os valores de hora são formatados. O caractere real usado como o separador de hora na saída formatada é determinado pelas configurações do sistema.
(/)	Separador de data. Em algumas localidades, outros caracteres podem ser usados para representar o separador de data. O separador de data separa o dia, o mês e o ano quando os valores de data são formatados. O caractere real usado como o separador de data na saída formatada é determinado pelas configurações do sistema.
c	Exibe a data como <code>ddddd</code> e a hora como <code>ttttt</code> , nessa ordem. Exibe apenas as informações de data se não houver nenhuma parte fracionária para o número de série de data. Exibe apenas as informações de hora se não houver nenhuma parte inteira.
d	Exibe o dia como um número sem zero à esquerda (1–31).
dd	Exibe o dia como um número com um zero à esquerda (01–31).
ddd	Exibe o dia como uma abreviação (dom–sáb). Localizado.
dddd	Exibe o dia como um nome completo (domingo–sábado). Localizado.
ddddd	Exibe a data como uma data completa (incluindo dia, mês e ano), formatada de acordo com a configuração de formato de data abreviada do seu sistema. O formato de data abreviada padrão é <code>m/d/yy</code> .
dddddd	Exibe um número de série de data como uma data completa (incluindo dia, mês e ano) formatado de acordo com a configuração de data completa reconhecida pelo seu sistema. O formato de data completa padrão é <code>mmm dd, yyyy</code> .
w	Exibe o dia da semana como um número (de 1 para domingo a 7 para sábado).

CARACTERE	DESCRIÇÃO
ww	Exibe a semana do ano como um número (1–54).
m	Exibe o mês como um número sem um zero à esquerda (1–12). Se <code>m</code> vier imediatamente em seguida de <code>h</code> ou <code>hh</code> , o minuto, em vez do mês, será exibido.
mm	Exibe o mês como um número com um zero à esquerda (01–12). Se <code>m</code> vier imediatamente em seguida de <code>h</code> ou <code>hh</code> , o minuto, em vez do mês, será exibido.
mmm	Exibe o mês como uma abreviação (jan–dez). Localizado.
mmm	Exibe o mês como um nome completo (janeiro–dezembro). Localizado.
q	Exibe o trimestre do ano como um número (1–4).
a	Exibe o dia do ano como um número (1–366).
aa	Exibe o ano como um número de 2 dígitos (00–99).
aaaa	Exibe o ano como um número de 4 dígitos (100–9999).
h	Exibe a hora como um número sem zero à esquerda (0–23).
hh	Exibe a hora como um número com um zero à esquerda (00–23).
n	Exibe o minuto como um número sem zero à esquerda (0–59).
nn	Exibe o minuto como um número com um zero à esquerda (00–59).
s	Exibe o segundo como um número sem zero à esquerda (0–59).
ss	Exibe o segundo como um número com um zero à esquerda (00–59).
tttt	Exibe a hora como uma hora completa (incluindo hora, minuto e segundo), formatada usando o separador de hora definido pelo formato de hora reconhecido pelo seu sistema. Um zero à esquerda será exibido se a opção de zero à esquerda for selecionada e a hora for anterior às 10:00 AM. ou P.M. O formato de hora padrão é <code>h:mm:ss</code> .
AM/PM	Usa o relógio de 12 horas e exibe AM em letras maiúsculas em qualquer hora antes do meio-dia; exibe PM em letras maiúsculas em qualquer hora entre meio-dia e 11:59 PM.

CARACTERE	DESCRIÇÃO
am/pm	Usa o relógio de 12 horas e exibe AM em letras minúsculas com qualquer hora antes do meio-dia; exibe PM em letras minúsculas com qualquer hora entre o meio-dia e 11:59 PM.
A/P	Usa o relógio de 12 horas e exibe um A maiúsculo com qualquer hora antes do meio-dia; exibe um P maiúsculo com qualquer hora entre o meio-dia e 11:59 PM.
a/p	Usa o relógio de 12 horas e exibe um A minúsculo com qualquer hora antes do meio-dia; exibe um P minúsculo com qualquer hora entre o meio-dia e 11:59 PM.
AMPM	Usa o relógio de 12 horas e exibe o literal de cadeia de caracteres AM, conforme definido pelo sistema, com qualquer hora antes do meio-dia; exibe o literal de cadeia de caracteres PM, conforme definido pelo sistema, com qualquer hora entre o meio-dia e 11:59 PM. AMPM pode ser indicado em letras maiúsculas ou minúsculas, porém, as maiúsculas e as minúsculas da cadeia de caracteres exibidas correspondem à cadeia de caracteres, conforme definido pelas configurações do sistema. O formato padrão é AM/PM. Caso o sistema esteja definido como um relógio de 24 horas, a cadeia de caracteres será definida como uma cadeia de caracteres de comprimento zero.

## Comentários

A formatação de data/hora usa a localidade atual do usuário para determinar o formato final da cadeia de caracteres. Por exemplo, para formatar a data 18 de março de 1995, com a seguinte cadeia de caracteres de formato "m/d/yyyy", caso a localidade do usuário esteja definida como Estados Unidos da América (en-us), o resultado será "3/18/1995", no entanto, caso a localidade do usuário esteja definida como Alemanha (de-de), o resultado será "18.03.1995".

## Consulte também

[Função FORMAT \(DAX\)](#)

[Formatos numéricos personalizados para a função FORMAT](#)

[Formatos de data e hora predefinidos para a função FORMAT](#)

# LEFT

09/05/2020 • 2 minutes to read

Retorna o número especificado de caracteres do início de uma cadeia de texto.

## Sintaxe

```
LEFT(<text>, <num_chars>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	A cadeia de texto que contém os caracteres que você deseja extrair ou uma referência a uma coluna que contém texto.
num_chars	(Opcional) Número de caracteres que a função LEFT deverá extrair. Se omitido, assumirá o valor 1.

## Valor da propriedade/valor retornado

Uma cadeia de texto.

## Comentários

Enquanto o Microsoft Excel contém diferentes funções para trabalhar com texto em linguagens de caractere de byte único e byte duplo, o DAX funciona com Unicode e armazena todos os caracteres com o mesmo comprimento. Portanto, uma única função é suficiente.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

O exemplo a seguir retorna os cinco primeiros caracteres do nome da empresa da coluna [ResellerName] e as cinco primeiras letras do código geográfico da coluna [GeographyKey] e os concatena para criar um identificador.

```
=CONCATENATE(LEFT('Reseller'[ResellerName], LEFT(GeographyKey, 3)))
```

Se o argumento **num\_chars** for um número maior que o número de caracteres disponíveis, a função retornará o máximo de caracteres disponíveis e não resultará em erro. Por exemplo, a coluna [GeographyKey] contém números como 1, 12 e 311. Portanto, o resultado também tem comprimento variável.

## Consulte também

[Funções de texto \(DAX\)](#)

# LEN

09/05/2020 • 2 minutes to read

Retorna o número de caracteres em uma cadeia de texto.

## Sintaxe

```
LEN(<text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto cujo comprimento você deseja localizar ou uma coluna que contém texto. Os espaços contam como caracteres.

## Valor retornado

Um número inteiro que indica o número de caracteres na cadeia de caracteres de texto.

## Comentários

Enquanto o Microsoft Excel contém funções diferentes para trabalhar com linguagens de caractere de byte único e de byte duplo, o DAX usa Unicode e armazena todos os caracteres com o mesmo comprimento.

Portanto, LEN sempre conta cada caractere como 1, independentemente da configuração de idioma padrão.

Se você usar LEN com uma coluna que contém valores que não são texto, como datas ou booleanos, a função converterá implicitamente o valor em texto usando o formato de coluna atual.

## Exemplo

A fórmula a seguir soma os comprimentos de endereços nas colunas, [AddressLine1] e [AddressLine2].

```
=LEN([AddressLine1])+LEN([AddressLin2])
```



# LOWER

09/05/2020 • 2 minutes to read

Converte todas as letras de uma cadeia de texto em minúsculas.

## Sintaxe

```
LOWER(<text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto que você deseja converter em minúsculas ou uma referência a uma coluna que contém texto.

## Valor da propriedade/valor retornado

Texto em minúsculas.

## Comentários

Caracteres que não forem letras não serão alterados. Por exemplo, a fórmula `=LOWER("123ABC")` retorna **123abc**.

## Exemplo

A fórmula a seguir pega cada linha da coluna [ProductCode] e converte o valor em minúsculas. Os números da coluna não serão afetados.

```
=LOWER('New Products'[ProductCode])
```

## Consulte também

[Funções de texto \(DAX\)](#)

# MID

08/05/2020 • 2 minutes to read

Retorna uma cadeia de caracteres do meio de uma cadeia de texto, dados um ponto inicial e um comprimento.

## Sintaxe

```
MID(<text>, <start_num>, <num_chars>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	A cadeia de texto da qual você deseja extrair os caracteres ou uma coluna que contém texto.
start_num	A posição do primeiro caractere que você deseja extrair. As posições começam em 1.
num_chars	O número de caracteres a serem retornados.

## Valor da propriedade/valor retornado

Uma cadeia de texto do comprimento especificado.

## Comentários

Enquanto o Microsoft Excel contém funções diferentes para trabalhar com linguagens de caracteres de byte único e de byte duplo, o DAX usa Unicode e armazena todos os caracteres com o mesmo tamanho.

## Exemplos

A seguinte expressão,

```
MID("abcde",2,3))
```

Retorna **"bcd"**.

A seguinte expressão,

```
MID('Reseller'[ResellerName],1,5))
```

Retorna o mesmo resultado que `LEFT([ResellerName],5)`. As duas expressões retornam as cinco primeiras letras da coluna, `[ResellerName]`.

## Consulte também

[Funções de texto \(DAX\)](#)

# REPLACE

09/05/2020 • 2 minutes to read

REPLACE substitui parte de uma cadeia de texto, com base no número de caracteres que você especificar, com uma cadeia de texto diferente.

## Sintaxe

```
REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
old_text	A cadeia de texto que contém os caracteres que você deseja substituir ou uma referência a uma coluna que contém texto.
start_num	A posição do caractere em <b>old_text</b> que você deseja substituir por <b>new_text</b> .
num_chars	O número de caracteres a serem substituídos. <b>Aviso:</b> Se o argumento <i>num_chars</i> for um espaço em branco ou fizer referência a uma coluna que seja avaliada como um espaço em branco, a cadeia de caracteres para <i>new_text</i> será inserida na posição, <i>start_num</i> , sem que nenhum caractere seja substituído. Esse é o mesmo comportamento do Excel.
new_text	O texto de substituição dos caracteres especificados em <b>old_text</b> .

## Valor da propriedade/valor retornado

Uma cadeia de texto.

## Comentários

Enquanto o Microsoft Excel contém funções diferentes para usar linguagens de caractere de byte único e de byte duplo, o DAX usa Unicode e, portanto, armazena todos os caracteres com o mesmo tamanho.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo

A fórmula a seguir cria uma nova coluna calculada que substitui os dois primeiros caracteres do código do produto na coluna, [ProductCode], por um novo código de duas letras, OB.

```
=REPLACE('New Products'[Product Code],1,2,"OB")
```

## Consulte também

[Funções de texto \(DAX\)](#)

[Função SUBSTITUTE \(DAX\)](#)

# REPT

09/05/2020 • 2 minutes to read

Repete um texto um determinado número de vezes. Use a função REPT para preencher uma célula com um determinado número de instâncias de uma cadeia de texto.

## Sintaxe

```
REPT(<text>, <num_times>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto que você deseja repetir.
num_times	Um número positivo que especifica o número de vezes que o texto deverá ser repetido.

## Valor da propriedade/valor retornado

Uma cadeia de caracteres que contém as alterações.

## Comentários

Se **number\_times** for zero (0), a função REPT retornará um espaço em branco.

Se **number\_times** não for um número inteiro, ele estará truncado.

O resultado da função REPT não poderá ser maior que 32.767 caracteres ou ela retornará um erro.

## Exemplo: Repetição de cadeias de caracteres literais

### Descrição

O exemplo a seguir retorna a cadeia de caracteres 85 repetida três vezes.

### Código

```
=REPT("85",3)
```

## Exemplo: Repetição dos valores da coluna

### Descrição

O exemplo a seguir retorna a cadeia de caracteres da coluna [MyText], repetida o número de vezes especificado na coluna [MyNumber]. Como a fórmula se estende para a coluna inteira, a cadeia de caracteres resultante depende do texto e do valor do número de cada linha.

### Código

=REPT([MyText],[MyNumber])

Comentários

MYTEXT	MYNUMBER	CALCULATEDCOLUMN1
Texto	2	TextText
Número	0	
85	3	858585

Consulte também

[Funções de texto \(DAX\)](#)

# RIGHT

09/05/2020 • 2 minutes to read

RIGHT retorna o último caractere ou caracteres em uma cadeia de texto, com base no número de caracteres que você especificar.

## Sintaxe

```
RIGHT(<text>, <num_chars>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	A cadeia de texto que contém os caracteres que você deseja extrair ou uma referência a uma coluna que contém texto.
num_chars	(Opcional) Número de caracteres que a função RIGHT deverá extrair. Se omitido, assumirá o valor 1. Você também pode usar uma referência a uma coluna que contém números.

Se a referência de coluna não contiver texto, ela será implicitamente convertida para texto.

## Valor da propriedade/valor retornado

Uma cadeia de texto que contém os caracteres à direita especificados.

## Comentários

RIGHT sempre conta cada caractere, seja ele de um byte ou de dois bytes, como 1, não importando a configuração de idioma padrão.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre as diferenças semânticas do modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>

## Exemplo: Retornando um número de caracteres fixo

### Descrição

A fórmula a seguir retorna os dois últimos dígitos do código do produto na tabela Novos Produtos.

### Código

```
=RIGHT('New Products'[ProductCode],2)
```

## Exemplo: Usando uma referência de coluna para especificar a contagem de caracteres

### Descrição

A fórmula a seguir retorna um número variável de dígitos do código do produto na nova tabela produtos, dependendo do número na coluna, MyCount. Se não existir nenhum valor na coluna MyCount ou o valor for um espaço em branco, BLANK também retornará um espaço em branco.

### Código

```
=RIGHT('New Products'[ProductCode],[MyCount])
```

## Consulte também

[Funções de texto \(DAX\)](#)

[Função LEFT \(DAX\)](#)

[Função MID \(DAX\)](#)



# SEARCH

09/05/2020 • 6 minutes to read

Retorna o número do caractere no qual um caractere específico ou uma cadeia de texto é encontrada pela primeira vez, lendo da esquerda para a direita. A pesquisa não diferencia maiúsculas de minúsculas e diferencia acentos.

## Sintaxe

```
SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])
```

### Parâmetros

TERMO	DEFINIÇÃO
find_text	O texto que você deseja encontrar.  Você pode usar caracteres curinga – o ponto de interrogação (?) e o asterisco (*) – em <b>find_text</b> . Um ponto de interrogação corresponde a qualquer caractere único; um asterisco corresponde a qualquer sequência de caracteres. Se você quiser localizar um ponto de interrogação ou um asterisco real, digite um til (~) antes do caractere.
within_text	O texto no qual você deseja pesquisar <b>find_text</b> ou uma coluna contendo texto.
start_num	(opcional) A posição do caractere em <b>within_text</b> em que você deseja iniciar a pesquisa. Se omitido, 1.
NotFoundValue	(opcional) O valor que deve ser retornado quando a operação não encontrar uma substring correspondente, normalmente 0, -1 ou BLANK().

## Valor retornado

O número da posição inicial da primeira cadeia de texto do primeiro caractere da segunda cadeia de texto.

## Comentários

1. A função de pesquisa não diferencia maiúsculas de minúsculas. A pesquisa por "N" encontrará a primeira ocorrência de "N" ou "n".
2. A função de pesquisa diferencia acentos. Pesquisar por "á" encontrará a primeira ocorrência de "á", mas nenhuma ocorrência de "a", "à" ou das versões em maiúsculas, "A" e "Á".
3. Usando essa função, você pode localizar uma cadeia de texto dentro de uma segunda cadeia de texto e retornar a posição em que a primeira cadeia é iniciada.
4. Você pode usar a função de pesquisa para determinar a localização de um caractere ou de uma cadeia de texto dentro de outra cadeia de texto e, em seguida, usar a função MID para retornar o texto ou usar a função REPLACE para alterar o texto.

5. Se **find\_text** não puder ser encontrado em **within\_text**, a fórmula retornará um erro. Esse comportamento é como o do Excel, que retorna #VALUE se a substring não for encontrada. Os valores nulos em **within\_text** serão interpretados como uma cadeia de caracteres vazia neste contexto.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo: Pesquisar dentro de uma cadeia de caracteres

### Descrição

A fórmula a seguir localiza a posição da letra "n" na palavra "impressora".

### Código

```
=SEARCH("n","printer")
```

### Comentários

A fórmula retorna 4 porque "n" é o quarto caractere da palavra "impressora".

## Exemplo: Pesquisar dentro de uma coluna

### Descrição

Você pode usar uma referência de coluna como um argumento para SEARCH. A fórmula a seguir localiza a posição do caractere "-" (hífen) na coluna [PostalCode].

### Código

```
=SEARCH("-",[PostalCode])
```

### Comentários

O resultado retornado é uma coluna de números, indicando a posição do índice do hífen.

## Exemplo: Tratamento de erros com SEARCH

### Descrição

A fórmula no exemplo anterior falhará se a cadeia de caracteres de pesquisa não for encontrada em todas as linhas da coluna de origem. Portanto, o exemplo a seguir demonstra como usar IFERROR com a função SEARCH para garantir que um resultado válido seja retornado para cada linha.

A fórmula a seguir localizará a posição do caractere "-" dentro da coluna e retornará -1 se a cadeia de caracteres não for encontrada.

### Código

```
= IFERROR(SEARCH("-",[PostalCode]),-1)
```

### Comentários

Observe que o tipo de dados do valor que você usa como uma saída de erro deve corresponder ao tipo de dados do tipo de saída que não é de erro. Nesse caso, você fornece um valor numérico a ser apresentado em caso de erro porque a pesquisa retorna um valor inteiro.

No entanto, você também pode retornar um espaço em branco (cadeia de caracteres vazia) usando `BLANK()` como

o segundo argumento para IFERROR.

## Consulte também

[Função MID \(DAX\)](#)

[Função REPLACE \(DAX\)](#)

[Funções de texto \(DAX\)](#)

# SUBSTITUTE

09/05/2020 • 2 minutes to read

Substitui o texto original pelo novo texto em uma cadeia de texto.

## Sintaxe

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto em que você deseja substituir caracteres ou uma referência a uma coluna contendo texto.
old_text	O texto original que você deseja substituir.
new_text	O texto que você deseja inserir no lugar de <b>old_text</b> .
instance_num	(Opcional) A ocorrência de <b>old_text</b> que você deseja substituir. Se omitido, todas as instâncias de <b>old_text</b> serão substituídas

## Valor da propriedade/valor retornado

Uma cadeia de texto.

## Comentários

Use a função SUBSTITUTE quando desejar substituir um texto específico em uma cadeia de texto; e use a função REPLACE quando desejar substituir qualquer texto de comprimento variável que ocorra em uma localização específica de uma cadeia de texto.

A função SUBSTITUTE diferencia maiúsculas de minúsculas. Se não houver uma correspondência exata das configurações de maiúsculas e minúsculas entre **text** e **old\_text**, a função SUBSTITUTE não substituirá o texto.

Essa função DAX pode retornar resultados diferentes quando usada em um modelo implantado e, em seguida, consultado no modo DirectQuery. Para obter mais informações sobre diferenças semânticas no modo DirectQuery, confira <https://go.microsoft.com/fwlink/?LinkId=219171>.

## Exemplo: Substituição em uma cadeia de caracteres

### Descrição

A fórmula a seguir cria uma cópia da coluna [Product Code] que substitui o novo código do produto **NW** pelo código de produto antigo **PA** onde quer que ele ocorra na coluna.

### Código

```
=SUBSTITUTE([Product Code], "NW", "PA")
```

## Consulte também

[Funções de texto \(DAX\)](#)

[Função REPLACE \(DAX\)](#)

# TRIM

08/05/2020 • 2 minutes to read

Remove todos os espaços de um valor de texto, exceto espaços simples entre palavras.

## Sintaxe

```
TRIM(<text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
text	O texto do qual você deseja remover os espaços ou uma coluna que contenha texto.

## Valor da propriedade/valor retornado

A cadeia de caracteres com espaços removidos.

## Comentários

Use TRIM no texto que você recebeu de outro aplicativo que pode ter espaçamento irregular.

A função TRIM foi criada originalmente para cortar o caractere de espaço ASCII de 7 bits (valor 32) do texto. No conjunto de caracteres Unicode, há um caractere de espaço adicional chamado caractere de espaço incondicional que tem um valor decimal de 160. Esse caractere costuma ser usado em páginas da Web como a entidade HTML, &nbsp;. Por si só, a função TRIM não remove esse caractere de espaço incondicional. Para obter um exemplo de como cortar os dois caracteres de espaço do texto, confira Remover espaços e caracteres não imprimíveis do texto.

## Exemplo

A fórmula a seguir cria uma nova cadeia de caracteres que não tem espaço em branco à direita.

```
=TRIM("A column with trailing spaces. ")
```

Quando você cria a fórmula, ela é propagada pela linha assim como você a digitou, para que você veja a cadeia de caracteres original em cada fórmula e para que os resultados não sejam aparentes. No entanto, quando a fórmula é avaliada, a cadeia de caracteres é cortada.

Você pode verificar se a fórmula produz o resultado correto verificando o comprimento da coluna calculada criada pela fórmula anterior, da seguinte maneira:

```
=LEN([Calculated Column 1])
```

## Consulte também

[Funções de texto \(DAX\)](#)

# UNICHAR

09/05/2020 • 2 minutes to read

Retorna o caractere Unicode referenciado pelo valor numérico.

## Sintaxe

```
UNICHAR(number)
```

### Parâmetros

TERMO	DEFINIÇÃO
número	O número Unicode que representa o caractere.

## Valor retornado

Um caractere representado pelo número Unicode

## Comentários

Se os caracteres XML não forem inválidos, UNICHAR retornará um erro.

Se os números Unicode forem substitutos parciais e os tipos de dados não forem válidos, UNICHAR retornará um erro.

Se os números forem valores numéricos que se enquadrarem fora do intervalo permitido, UNICHAR retornará um erro.

Se o número for zero (0), UNICHAR retornará um erro.

O caractere Unicode retornado pode ser uma cadeia de caracteres, por exemplo, em códigos UTF-8 ou UTF-16.

## Exemplo

O exemplo a seguir retorna o caractere representado pelo número Unicode 66 (A maiúscula).

```
=UNICHAR(65)
```

O exemplo a seguir retorna o caractere representado pelo número Unicode 32 (caractere de espaço).

```
=UNICHAR(32)
```

O exemplo a seguir retorna o caractere representado pelo número Unicode 9733 (caractere ★).

```
=UNICHAR(9733)
```

# UNICODE

09/05/2020 • 2 minutes to read

Retorna o número (ponto de código) correspondente ao primeiro caractere do texto.

## Sintaxe

```
UNICODE(<Text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
Texto	Texto é o caractere do qual você deseja o valor Unicode.

## Valor retornado

Um código numérico para o primeiro caractere em uma cadeia de texto.



# UPPER

09/05/2020 • 2 minutes to read

Converte uma cadeia de texto em letras maiúsculas.

## Sintaxe

```
UPPER (<text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto que você deseja converter em letras maiúsculas ou uma referência a uma coluna que contém texto.

## Valor da propriedade/valor retornado

Mesmo texto, em letras maiúsculas.

## Exemplo

A fórmula a seguir converte a cadeia de caracteres da coluna [ProductCode] deixando-a com todas as letras maiúsculas. Os caracteres não alfabéticos não são afetados.

```
=UPPER(['New Products'[Product Code])
```

## Consulte também

[Funções de texto \(DAX\)](#)

[Função LOWER \(DAX\)](#)

# VALUE

09/05/2020 • 2 minutes to read

Converte em número uma cadeia de texto que representa um número.

## Sintaxe

```
VALUE(<text>)
```

### Parâmetros

TERMO	DEFINIÇÃO
texto	O texto a ser convertido.

## Valor retornado

O número convertido em tipo de dados decimal.

## Comentários

O valor passado como parâmetro de **texto** pode estar em qualquer um dos formatos – constante, número, data ou hora – reconhecidos pelo aplicativo ou pelos serviços que você está usando. Se o **texto** não estiver em um desses formatos, um erro será retornado.

Geralmente, você não precisa usar a função VALUE em uma fórmula, pois o mecanismo converte implicitamente o texto em números, conforme necessário.

Você também pode usar referências de coluna. Por exemplo, se você tiver uma coluna que contém diferentes tipos numéricos misturados, a função VALUE poderá ser usada para converter todos os valores em um único tipo de dados numérico. No entanto, se você usar a função VALUE com uma coluna que contém números e texto misturados, a coluna inteira será sinalizada com um erro, pois nem todos os valores de todas as linhas poderão ser convertidos em números.

## Exemplo

A fórmula a seguir converte a cadeia de caracteres digitada, "3", no valor numérico 3.

```
=VALUE("3")
```

## Consulte também

[Funções de texto \(DAX\)](#)

# Instruções

09/05/2020 • 2 minutes to read

## Nesta categoria

DE	DESCRIÇÃO
DEFINE	(Palavra-chave) Define entidades que existem apenas durante uma consulta DAX.
EVALUATE	(Palavra-chave) Uma instrução necessária para executar uma consulta DAX.
ORDER BY	(Palavra-chave) Define uma ou mais expressões usadas para classificar os resultados de uma consulta DAX.
VAR	(Palavra-chave) Armazena o resultado de uma expressão como uma variável nomeada, que pode ser passada como argumento para outras expressões de medida.

# DEFINIR

09/05/2020 • 2 minutes to read

Uma palavra-chave que define entidades que podem ser aplicadas a uma ou mais instruções EVALUATE de uma [consulta DAX](#).

## Sintaxe

```
DEFINE { <entity> [<name>] = <expression> }
```

### Argumentos

TERMO	DEFINIÇÃO
entidade	MEASURE, VAR, TABLE ou COLUMN.
Nome	O nome de uma entidade. Não pode ser uma expressão.
expressão	Qualquer expressão DAX que retorne um único valor escalar. A expressão pode usar qualquer uma das entidades definidas. A expressão deve retornar uma tabela. Se um valor escalar for necessário, encapsule o escalar dentro de uma função ROW() para produzir uma tabela.

## Comentários

As entidades podem ser variáveis, medidas, tabelas e colunas.

As definições normalmente precedem a instrução EVALUATE e são válidas para todas as instruções EVALUATE.

As definições podem fazer referência a outras definições que aparecem antes ou depois da definição atual.

As definições existem apenas durante a consulta.

## Consulte também

[Consultas DAX](#)

[ORDER BY](#)

[VAR](#)

# AVALIAR

09/05/2020 • 2 minutes to read

Uma instrução que contém uma expressão de tabela necessária em uma [consulta DAX](#).

## Sintaxe

```
EVALUATE <table>
```

### Parâmetros

TERMO	DEFINIÇÃO
tabela	Uma expressão de tabela

## Valor retornado

O resultado de uma expressão de tabela.

## Comentários

Uma consulta pode conter várias instruções EVALUATE.

## Exemplo

```
EVALUATE(
 'Internet Sales'
)
```

Retorna todas as linhas e colunas da tabela Vendas pela Internet como uma tabela.

## Consulte também

[Consultas DAX](#)

[DEFINE](#)

[ORDER BY](#)

# ORDER BY

09/05/2020 • 2 minutes to read

Define a ordem de classificação dos resultados da consulta retornados por uma instrução EVALUATE em uma [consulta DAX](#).

## Sintaxe

```
ORDER BY {<expression> [{ASC | DESC}]}
```

### Argumentos

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorne um único valor escalar.
ASC	(padrão) Ordem de classificação crescente.
DESC	Ordem de classificação decrescente.

## Valor retornado

O resultado de uma instrução EVALUATE em ordem crescente (ASC) ou decrescente (DESC).

## Exemplo

```
EVALUATE(
 'Internet Sales'
)
ORDER BY
 'Internet Sales'[Order Date]
```

Retorna todas as linhas e colunas da tabela Vendas pela Internet ordenadas por Data do Pedido, como uma tabela.

## Consulte também

[Consultas DAX](#)

[EVALUATE](#)

# VAR

09/05/2020 • 3 minutes to read

Armazena o resultado de uma expressão como uma variável com nome, que pode ser passada como argumento para outras expressões de medida. Depois que os valores resultantes tiverem sido calculados para uma expressão variável, esses valores não serão alterados, mesmo que a variável seja referenciada em outra expressão.

## Sintaxe

```
VAR <name> = <expression>
```

### Parâmetros

TERMO	DEFINIÇÃO
Nome	<p>O nome da variável (identificador).</p> <p>Não há suporte para delimitadores. Por exemplo, "varName" ou [varName] resultará em erro.</p> <p>Conjunto de caracteres compatíveis: a-z, A-Z, 0-9.</p> <p>Os caracteres 0-9 não são válidos como primeiro caractere.</p> <p>O sublinhado duplo () <b>é permitido como prefixo do nome do identificador</b>.</p> <p>Nenhum outro caractere especial é compatível.</p> <p>Palavras-chave reservadas não são permitidas.</p> <p>Nomes de tabelas existentes não são permitidos.</p> <p>Espaços vazios não são permitidos.</p>
expressão	<p>Uma expressão DAX que retorna um valor escalar ou de tabela.</p>

## Valor retornado

Uma variável com nome que contém o resultado do argumento da expressão.

## Exceções

## Comentários

Uma expressão passada como argumento para a função VAR pode conter outra declaração da função VAR.

Ao referenciar uma variável:

- As medidas não podem se referir a variáveis definidas fora da expressão de medida, mas podem se referir a variáveis de escopo funcional definidas dentro da expressão.
- As variáveis podem se referir a medidas.
- As variáveis podem se referir a variáveis definidas anteriormente.
- As colunas nas variáveis de tabela não podem ser referenciadas por meio da sintaxe TableName [ColumnName].

## Exemplo

Para calcular um percentual de crescimento em relação ao ano anterior sem usar uma variável, você pode criar três medidas separadas. A primeira medida calcula a soma do valor de vendas:

```
Sum of SalesAmount = SUM(SalesTable[SalesAmount])
```

A segunda medida calcula o valor de vendas no ano anterior:

```
SalesAmount PreviousYear =
 CALCULATE([Sum of SalesAmount],
 SAMEPERIODLASTYEAR(Calendar[Date]))
)
```

Em seguida, você pode criar uma terceira medida que combina as outras duas medidas a fim de calcular o percentual de crescimento. Observe que a soma da medida SalesAmount é usada em dois lugares: primeiro, para determinar se há uma venda, depois novamente para calcular o percentual.

```
Sum of SalesAmount YoY%: =
 IF([Sum of SalesAmount] ,
 DIVIDE(([Sum of SalesAmount] - [SalesAmount PreviousYear]), [Sum of SalesAmount])
)
```

Usando uma variável, você poderá criar uma única medida que calcula o mesmo resultado:

```
YoY% = VAR Sales = SUM(SalesTable[SalesAmount])

VAR SalesLastYear =
 CALCULATE (SUM (SalesTable[SalesAmount]), SAMEPERIODLASTYEAR ('Calendar'[Date]))

return if(Sales, DIVIDE(Sales - SalesLastYear, Sales))
```

Usando uma variável, você poderá obter o mesmo resultado, mas de uma maneira mais legível. Além disso, o resultado da expressão é armazenado na variável no momento da declaração. Não será necessário recalculá-lo o resultado cada vez que ele for usado, como ocorre quando não se usa uma variável. Isso pode melhorar o desempenho da medida.



# Sintaxe do DAX

08/05/2020 • 17 minutes to read

Este artigo descreve a sintaxe e os requisitos para a linguagem de expressão de fórmula DAX.

## Requisitos de sintaxe

Uma fórmula DAX sempre começa com um sinal de igual (=). Após o sinal de igual, você pode fornecer qualquer expressão que seja avaliada como um escalar ou uma expressão que possa ser convertida em um escalar. Elas incluem o seguinte:

- Uma constante escalar ou expressão que usa um operador escalar (+, -, \*, /, >=, <, & , ...)
- Referências a colunas ou tabelas. A linguagem DAX sempre usa tabelas e colunas como entradas para funções, nunca uma matriz ou um conjunto arbitrário de valores.
- Operadores, constantes e valores fornecidos como parte de uma expressão.
- O resultado de uma função e seus argumentos necessários. Algumas funções DAX retornam uma tabela, em vez de uma escala, devendo ser encapsuladas em uma função que avalia a tabela e retorna um escalar; a menos que a tabela seja uma única coluna, uma única tabela de linha, ela é tratada como um valor escalar.

A maioria das funções DAX exige um ou mais argumentos, que podem incluir tabelas, colunas, expressões e valores. No entanto, algumas funções, como PI, não exigem argumentos, mas sempre exigem parênteses para indicar o argumento nulo. Por exemplo, você sempre deve digitar PI(), não PI. Você também pode aninhar funções em outras funções.

- Expressões. Uma expressão pode conter qualquer um dos seguintes: operadores, constantes ou referências a colunas.

Por exemplo, as fórmulas a seguir são todas válidas.

FÓRMULA	RESULTADO
=3	3
= "Sales"	Vendas
= 'Sales'[Amount]	Se você usar essa fórmula na tabela Sales, obterá o valor da coluna Amount na tabela Sales da linha atual.
= (0.03 * [Amount]) = 0.03 * [Amount]	Três por cento do valor na coluna Amount da tabela atual. Embora essa fórmula possa ser usada para calcular um percentual, o resultado não é mostrado como um percentual, a menos que você aplique a formatação na tabela.
=PI()	O valor da constante pi.

#### NOTE

As fórmulas podem se comportar de forma diferente, dependendo se são usadas em uma coluna calculada ou em uma medida dentro de uma Tabela Dinâmica. Você sempre deve estar atento ao contexto e a como os dados que você usa na fórmula estão relacionados a outros dados que podem ser usados no cálculo.

## Requisitos de nomenclatura

Um modelo de dados geralmente contém várias tabelas. Juntas, as tabelas e suas colunas compõem um banco de dados armazenado no mecanismo analítico na memória (VertiPaq). Dentro desse banco de dados, todas as tabelas devem ter nomes exclusivos. Os nomes das colunas também devem ser exclusivos em cada tabela. Os nomes de objeto *não diferenciam maiúsculas de minúsculas*; por exemplo, os nomes **SALES** e **Sales** representariam a mesma tabela.

Cada coluna e medida que você adiciona a um modelo de dados existente deve pertencer a uma tabela específica. Você especifica a tabela que contém a coluna implicitamente, ao criar uma coluna calculada dentro de uma tabela, ou explicitamente, ao criar uma medida e especifica o nome da tabela em que a definição da medida deve ser armazenada.

Quando você usa uma tabela ou coluna como uma entrada para uma função, geralmente deve *qualificar* o nome da coluna. O nome *totalmente qualificado* de uma coluna é o nome da tabela, seguido pelo nome da coluna entre colchetes: por exemplo, 'U.S. Sales'[Products]. Um nome totalmente qualificado sempre é necessário quando você faz referência a uma coluna nos seguintes contextos:

- Como um argumento para a função, VALUES
- Como um argumento para as funções, ALL ou ALLEXCEPT
- Em um argumento de filtro para as funções, CALCULATE ou CALCULATETABLE
- Como um argumento para a função, RELATEDTABLE
- Como um argumento para qualquer função de inteligência de tempo

Um nome de coluna *não qualificado* é apenas o nome da coluna entre colchetes: por exemplo, [Sales Amount]. Por exemplo, ao fazer referência a um valor escalar da mesma linha da tabela atual, você pode usar o nome de coluna não qualificado.

Se um nome de tabela contiver espaços, palavras-chave reservadas ou caracteres não permitidos, você precisará colocar o nome da tabela entre aspas simples. Você também deverá colocar os nomes de tabela entre aspas se o nome contiver quaisquer caracteres fora do conjunto de caracteres alfanuméricos ANSI, independentemente de sua localidade dar ou não suporte ao conjunto de caracteres. Por exemplo, se você abrir uma pasta de trabalho contendo nomes de tabela escritos em caracteres cirílicos, como 'Таблица', o nome da tabela deverá ser colocado entre aspas, mesmo que não contenha espaços.

#### NOTE

Para facilitar a inserção dos nomes totalmente qualificados das colunas, use o recurso de Preenchimento Automático no editor de fórmulas.

#### Tabelas

- Os nomes de tabela são necessários sempre que a coluna for de uma tabela diferente da tabela atual. Os nomes de tabela devem ser exclusivos no banco de dados.
- Os nomes de tabela deverão ser colocados entre aspas simples se contiverem espaços, outros caracteres especiais ou quaisquer caracteres alfanuméricos que não estejam em inglês.

## Medidas

- Os nomes de medidas sempre devem estar entre colchetes.
- Os nomes de medidas podem conter espaços.
- Cada nome de medida deve ser exclusivo dentro de um modelo. Portanto, o nome da tabela é opcional na frente de um nome de medida ao referenciar uma medida existente. No entanto, ao criar uma medida, você sempre deve especificar uma tabela na qual a definição da medida será armazenada.

## Colunas

Os nomes de coluna devem ser exclusivos no contexto de uma tabela; no entanto, várias tabelas podem ter colunas com os mesmos nomes (a ambiguidade vem com o nome da tabela).

Em geral, as colunas podem ser referenciadas sem referenciar a tabela base à qual pertencem, exceto quando houver um conflito de nome para resolver ou com determinadas funções que exigem que os nomes de coluna sejam totalmente qualificados.

## Palavras-chave reservadas

Se o nome usado para uma tabela for o mesmo que uma palavra-chave reservada do Analysis Services, um erro será gerado e você deverá renomear a tabela. No entanto, você poderá usar palavras-chave em nomes de objeto se o nome do objeto estiver entre colchetes (para colunas) ou aspas (para tabelas).

### NOTE

As aspas podem ser representadas por vários caracteres diferentes, dependendo do aplicativo. Se você colar fórmulas de um documento externo ou página da Web, verifique o código ASCII do caractere usado para abrir e fechar as aspas para garantir que elas sejam as mesmas. Caso contrário, o DAX poderá não conseguir reconhecer os símbolos como aspas, tornando a referência inválida.

## Caracteres especiais

Os seguintes caracteres e tipos de caracteres não são válidos nos nomes de tabelas, colunas ou medidas:

- Espaços à esquerda ou à direita; a menos que os espaços estejam entre delimitadores de nome, colchetes ou apóstrofos únicos.
- Caracteres de controle
- Os caracteres a seguir não são válidos nos nomes de objetos:

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

## Exemplos de nomes de objeto

A tabela a seguir mostra exemplos de alguns nomes de objeto:

Tipos de objeto	Exemplos	Comentário
Nome da tabela	<b>Vendas</b>	Se o nome da tabela não contiver espaços ou outros caracteres especiais, o nome não precisará ser colocado entre aspas.
Nome da tabela	<b>'Vendas do Canadá'</b>	Se o nome contiver espaços, tabulações ou outros caracteres especiais, coloque o nome entre aspas simples.

Nome da coluna totalmente qualificado	<b>Sales[Amount]</b>	O nome da tabela precede o nome da coluna e o nome da coluna é colocado entre colchetes.
Nome de medida totalmente qualificado	<b>Sales[Profit]</b>	O nome da tabela precede o nome da medida e o nome da medida é colocado entre colchetes. Em determinados contextos, um nome totalmente qualificado sempre é necessário.
Nome de coluna não qualificado	<b>[Amount]</b>	O nome não qualificado é apenas o nome da coluna entre colchetes. Contextos em que você pode usar o nome não qualificado incluem fórmulas em uma coluna calculada dentro da mesma tabela ou em uma função de agregação que está sendo verificada na mesma tabela.
Coluna totalmente qualificada na tabela com espaços	<b>'Vendas do Canadá'[Qtd]</b>	O nome da tabela contém espaços, portanto, deve estar entre aspas simples.

### Outras restrições

A sintaxe necessária para cada função e o tipo de operação que ela pode executar varia muito conforme a função. Porém, as regras a seguir geralmente se aplicam a todas as fórmulas e expressões:

- Fórmulas e expressões DAX não podem modificar nem inserir valores individuais em tabelas.
- Não é possível criar linhas calculadas usando DAX. Você pode criar apenas as colunas e medidas calculadas.
- Ao definir colunas calculadas, você pode aninhar funções em qualquer nível.
- O DAX tem várias funções que retornam uma tabela. Normalmente, você usa os valores retornados por essas funções como entrada para outras funções, que exigem uma tabela como entrada.

## Operadores e constantes DAX

A tabela a seguir lista os operadores que têm suporte no DAX. Para obter mais informações sobre a sintaxe de operadores individuais, confira [Operadores DAX](#).

TIPO DE OPERADOR	SÍMBOLO E USO
Operador de parênteses	() ordem de precedência e agrupamento de argumentos
Operadores aritméticos	+ (adição) - (subtração/ sinal) * (multiplicação) / (divisão) ^ (exponenciação)

TIPO DE OPERADOR	SÍMBOLO E USO
Operadores de comparação	= (igual a) > (maior que) < (menor que) >= (maior ou igual a) <= (inferior ou igual a) <> (não igual a)
Operador de concatenação de texto	& (concatenação)
Operadores lógicos	&& (e)    (ou)

## Tipos de dados

Você não precisa converter nem especificar de outra forma o tipo de dados de uma coluna ou valor que você usa em uma fórmula DAX. Quando você usa dados em uma fórmula DAX, o DAX identifica automaticamente os tipos de dados em colunas referenciadas e os valores que você digita e executa conversões implícitas, quando necessário, para concluir a operação especificada.

Por exemplo, se você tentar adicionar um número a um valor de data, o mecanismo interpretará a operação no contexto da função e converterá os números em um tipo de dados comum e, em seguida, apresentará o resultado no formato pretendido, uma data.

No entanto, há algumas limitações nos valores que podem ser convertidos com êxito. Se um valor ou uma coluna tiver um tipo de dados incompatível com a operação atual, o DAX retornará um erro. Além disso, o DAX não fornece funções que permitem que você altere ou converta explicitamente o tipo de dados existentes que você importou em um modelo de dados.

### IMPORTANT

O DAX não dá suporte ao uso do tipo de dados de variante. Portanto, quando você carrega ou importa dados em um modelo de dados, os dados em cada coluna geralmente devem ser todos de um tipo de dados consistente.

Algumas funções retornam valores escalares, incluindo cadeias de caracteres, enquanto outras funções funcionam com números, tanto inteiros quanto reais, ou datas e horas. O tipo de dados necessário para cada função é descrito na seção [Funções DAX](#).

Você pode usar tabelas que contêm várias colunas e várias linhas de dados como o argumento para uma função. Algumas funções também retornam tabelas, que são armazenadas na memória e podem ser usadas como argumentos para outras funções.

# Operadores DAX

26/05/2020 • 14 minutes to read

A linguagem DAX (Data Analysis Expression) usa operadores para criar expressões que comparam valores, executam cálculos aritméticos ou trabalham com cadeias de caracteres. Esta seção descreve o uso de cada operador.

## Tipos de operadores

Há quatro tipos diferentes de operadores de cálculo: aritmético, comparação, concatenação de texto e lógico.

### Operadores aritméticos

Para executar operações matemáticas básicas, como adição, subtração ou multiplicação; combinar números; e produzir resultados numéricos, use os operadores aritméticos a seguir.

OPERADOR ARITMÉTICO	SIGNIFICADO	EXEMPLO
+ (sinal de adição)	Adição	3+3
– (sinal de subtração)	Subtração ou sinal	3–1–1
* (asterisco)	Multiplicação	3*3
/ (barra)	Divisão	3/3
^ (sinal de interpolação)	Exponenciação	16^4

#### NOTE

O sinal de adição pode funcionar como um *operador binário* e como um *operador unário*. Um operador binário requer números em ambos os lados do operador e executa a adição. Quando você usa valores em uma fórmula DAX em ambos os lados do operador binário, o DAX tentará converter os valores em tipos de dados numéricos, se eles ainda não forem números. Por outro lado, o operador unário pode ser aplicado a qualquer tipo de argumento. O símbolo de adição não afeta o tipo nem o valor e é simplesmente ignorado, enquanto o operador de subtração cria um valor negativo, se aplicado a um valor numérico.

### Operadores de comparação

Você pode comparar dois valores com os operadores a seguir. Quando dois valores são comparados usando estes operadores, o resultado é um valor lógico, TRUE ou FALSE.

OPERADOR DE COMPARAÇÃO	SIGNIFICADO	EXEMPLO
=	Igual a	[Região] = "EUA"
==	Estrito igual a	[Região] == "EUA"
>	Maior que	[Data de vendas] > "Jan 2009"
<	Menor que	[Data de vendas] < "1º Jan 2009"

OPERADOR DE COMPARAÇÃO	SIGNIFICADO	EXEMPLO
>=	Maior ou igual a	[Quantidade] >= 20000
<=	Menor ou igual a	[Quantidade] <= 100
<>	Diferente de	[Região] <> "EUA"

Todos os operadores de comparação, exceto ==, tratam BLANK como igual ao número 0, cadeia de caracteres vazia "", DATE(1899, 12, 30) ou FALSE. Como resultado, [Column] = 0 será true quando o valor de [Column] for 0 ou BLANK. Por outro lado, [Column] == 0 é true somente quando o valor de [Column] é 0.

### Operador de concatenação de texto

Use o E comercial (&) para unir ou concatenar duas ou mais cadeias de caracteres de texto para produzir uma única parte do texto.

OPERADOR DE TEXTO	SIGNIFICADO	EXEMPLO
& (e comercial)	Conecta ou concatena dois valores para produzir um valor de texto contínuo	[Região] & ", " & [Cidade]

### Operadores lógicos

Use operadores lógicos (&&) e (||) para combinar expressões para produzir um único resultado.

OPERADOR DE TEXTO	SIGNIFICADO	EXEMPLOS
&& (e comercial duplo)	Cria uma condição AND entre duas expressões que têm um resultado booleano. Se ambas as expressões retornarem TRUE, a combinação das expressões também retornará TRUE; caso contrário, a combinação retornará FALSE.	(([Region] = "France") && ([BikeBuyer] = "yes"))
(símbolo de pipe duplo)	Cria uma condição OR entre duas expressões lógicas. Se uma das expressões retornar TRUE, o resultado será TRUE; somente quando as duas expressões são FALSE o resultado é FALSE.	(([Region] = "France")    ([BikeBuyer] = "yes"))
IN	Cria uma condição OR lógica entre cada linha sendo comparada a uma tabela. Observação: a sintaxe do construtor de tabela usa chaves.	'Produto'[Cor] IN { "Vermelho", "Azul", "Preto" }

## Operadores e ordem de precedência

Em alguns casos, a ordem na qual o cálculo é executado pode afetar o valor retornado. Portanto, é importante entender como o pedido é determinado e como você pode alterar a ordem para obter os resultados desejados.

### Ordem de cálculo

Uma expressão avalia os operadores e valores em uma ordem específica. Todas as expressões sempre começam com um sinal de igual (=). O sinal de igual indica que os caracteres seguintes constituem uma expressão.

Após o sinal de igual, estão os elementos a serem calculados (os operandos), que são separados por operadores de cálculo. As expressões sempre são lidas da esquerda para a direita, mas a ordem na qual os elementos são agrupados pode ser controlada em algum grau usando parênteses.

### Precedência do operador

Se você combinar vários operadores em uma única fórmula, as operações serão ordenadas conforme a tabela a seguir. Se os operadores tiverem um valor de precedência igual, eles serão ordenados da esquerda para a direita. Por exemplo, se uma expressão contiver tanto um operador de multiplicação quanto um de divisão, eles serão avaliados na ordem em que aparecem na expressão, da esquerda para a direita.

OPERADOR	DESCRIÇÃO
$\wedge$	Exponenciação
$-$	Sinal (como em $-1$ )
$*$ e $/$	Multiplicação e divisão
$!$	NOT (operador unário)
$+$ e $-$	Adição e subtração
$\&$	Conecta duas cadeias de caracteres de texto (concatenação)
$=, <, >, <=, >=, <>$	Comparação

### Usando parênteses para controlar a ordem de cálculo

Para alterar a ordem de avaliação, você deve incluir entre parênteses a parte da fórmula que deve ser calculada primeiro. Por exemplo, a fórmula a seguir produz 11 porque a multiplicação é calculada antes da adição. A fórmula multiplica 2 por 3 e, em seguida, adiciona 5 ao resultado.

`=5+2*3`

Por outro lado, se você usar parênteses para alterar a sintaxe, a ordem será alterada para que 5 e 2 sejam adicionados juntos e o resultado seja multiplicado por 3 para produzir 21.

`=(5+2)*3`

No exemplo a seguir, os parênteses em torno da primeira parte da fórmula forçam o cálculo a avaliar a expressão `(3 + 0.25)` primeiro e, em seguida, dividir o resultado pelo resultado da expressão `(3 - 0.25)`.

`=(3 + 0.25)/(3 - 0.25)`

No exemplo a seguir, o operador de exponenciação é aplicado primeiro, de acordo com as regras de precedência para operadores e, em seguida, o operador de sinalização é aplicado. O resultado dessa expressão é -4.

`=-2^2`

Para garantir que o operador de sinalização seja aplicado primeiro ao valor numérico, você pode usar parênteses para controlar operadores, conforme mostrado no exemplo a seguir. O resultado dessa expressão é 4.



$$= (-2)^2$$

## Compatibilidade

O DAX manipula e compara facilmente vários tipos de dados, assim como o Microsoft Excel. No entanto, o mecanismo de computação subjacente é baseado no SQL Server Analysis Services e fornece recursos avançados adicionais de um armazenamento de dados relacional, incluindo suporte mais avançado para tipos de data e hora. Portanto, em alguns casos, os resultados de cálculos ou o comportamento das funções podem não ser os mesmos do Excel. Além disso, o DAX dá suporte a mais tipos de dados do que o Excel. Esta seção descreve as principais diferenças.

### Coerção de tipos de dados de operandos

Em geral, os dois operandos nos lados esquerdo e direito de qualquer operador devem ser do mesmo tipo de dados. No entanto, se os tipos de dados forem diferentes, o DAX os converterá em um tipo de dados comum para aplicar o operador em alguns casos:

1. Os dois operandos são convertidos no maior tipo de dados comum possível.
2. O operador é aplicado, se possível.

Por exemplo, suponha que você tenha dois números que deseja combinar. Um número resulta de uma fórmula, como `= [Price] * .20`, e o resultado pode conter muitas casas decimais. O outro número é um inteiro que foi fornecido como um valor de cadeia de caracteres.

Nesse caso, o DAX converterá os números em números reais em um formato numérico, usando o maior formato numérico que pode armazenar os dois tipos de números. Em seguida, o DAX aplicará a multiplicação.

Dependendo da combinação de tipo de dados, a coerção de tipo poderá não ser aplicada a operações de comparação. Para ver uma lista de tipos de dados compatíveis com o DAX, confira [Tipos de dados compatíveis em modelos tabulares](#) e [Tipos de dados no Power BI Desktop](#).

Inteiro, Número Real, Moeda, Data/Hora e Em Branco são considerados numéricos para fins de comparação. Em branco é avaliado como zero ao executar uma comparação. As combinações de tipo de dados a seguir têm suporte para operações de comparação.

TIPO DE DADOS DO LADO ESQUERDO	TIPO DE DADOS DO LADO DIREITO
Numérica	Numérica
Booleano	Booleano
Cadeia de caracteres	Cadeia de caracteres

Outras comparações mistas de tipo de dados retornarão um erro. Por exemplo, uma fórmula como `"1" > 0` retorna um erro informando que *operações de comparação DAX não oferecem suporte a valores de comparação do tipo texto com valores do tipo inteiro*.

TIPOS DE DADOS USADOS NO DAX	TIPOS DE DADOS USADOS NO EXCEL
Números (I8, R8)	Números (R8)
Cadeia de caracteres	Cadeia de caracteres
Booleano	Booleano

TIPOS DE DADOS USADOS NO DAX	TIPOS DE DADOS USADOS NO EXCEL
DateTime	Variante
Moeda	Moeda

### Diferenças na ordem de precedência

A ordem de precedência das operações em fórmulas DAX é basicamente a mesma usada pelo Microsoft Excel, mas não há suporte para alguns operadores do Excel, como a porcentagem. Além disso, não há suporte para intervalos.

Portanto, sempre que você copiar e colar fórmulas do Excel, examine-as com cuidado, pois alguns operadores ou elementos nas fórmulas podem não ser válidos. Quando há alguma dúvida sobre a ordem na qual as operações são executadas, recomendamos usar parênteses para controlar a ordem das operações e remover qualquer ambiguidade sobre o resultado.

## Consulte também

[Sintaxe do DAX](#)

[Nomenclatura de parâmetro do DAX](#)

# Consultas do DAX

21/05/2020 • 8 minutes to read

Com consultas DAX, você pode consultar e retornar dados definidos por uma expressão de tabela. Os clientes de relatório constroem consultas DAX sempre que um campo é colocado em uma superfície de relatório ou quando um filtro ou cálculo é aplicado. As consultas DAX também podem ser criadas e executadas no SSMS ([SQL Server Management Studio](#)) e em ferramentas de software livre, como o [DAX Studio](#). As consultas DAX são executadas no SSMS e no DAX Studio retornam resultados como uma tabela.

Antes de aprender sobre as consultas, é importante que você tenha uma compreensão sólida das noções básicas do DAX. Se ainda não fez isto, certifique-se de conferir a [Visão geral do DAX](#).

## Sintaxe

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }
 { VAR <name> = <expression>}]
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
[START AT {<value>|<parameter>} [, ...]]]
```

## Palavras-chave

### EVALUATE (obrigatório)

No nível mais básico, uma consulta DAX é uma instrução **EVALUATE** que contém uma expressão de tabela. No entanto, uma consulta pode conter várias instruções **EVALUATE**.

#### Sintaxe

```
EVALUATE <table>
```

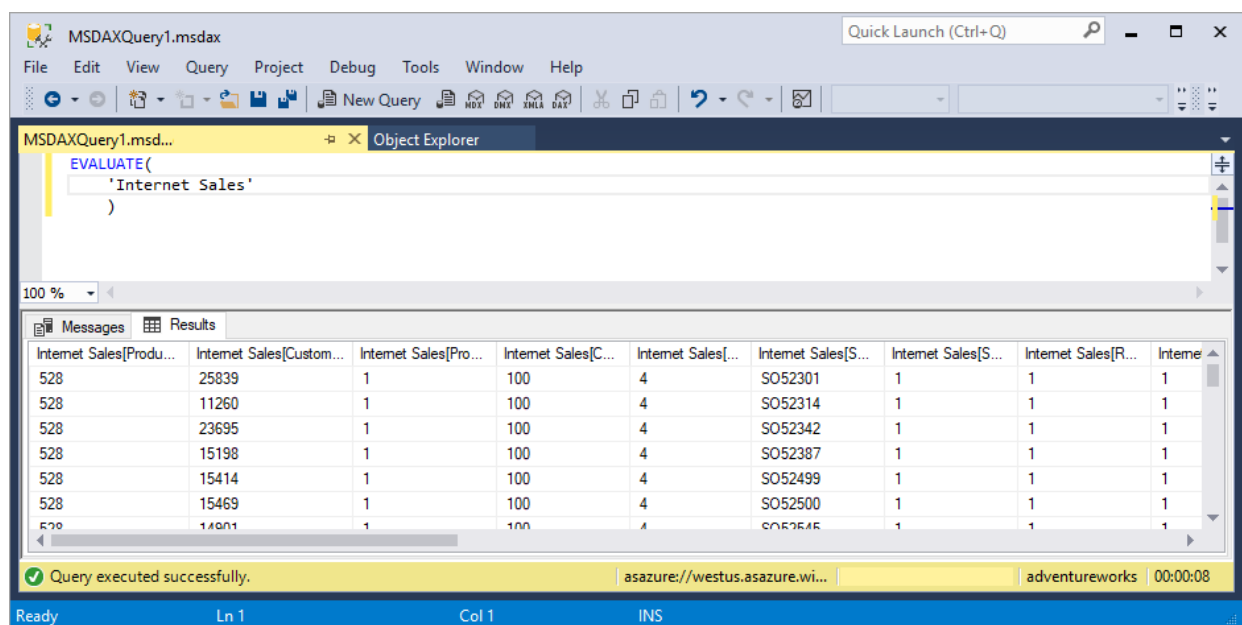
#### Argumentos

TERMO	DEFINIÇÃO
tabela	Uma expressão de tabela.

#### Exemplo

```
EVALUATE(
 'Internet Sales'
)
```

Retorna todas as linhas e colunas da tabela Vendas pela Internet como uma tabela.



## ORDER BY (opcional)

A palavra-chave opcional **ORDER BY** define uma ou mais expressões usadas para classificar os resultados da consulta. Qualquer expressão que possa ser avaliada para cada linha do resultado é válida.

### Sintaxe

```
EVALUATE <table>
[ORDER BY {<expression> [{ASC | DESC}]}[, ...]
```

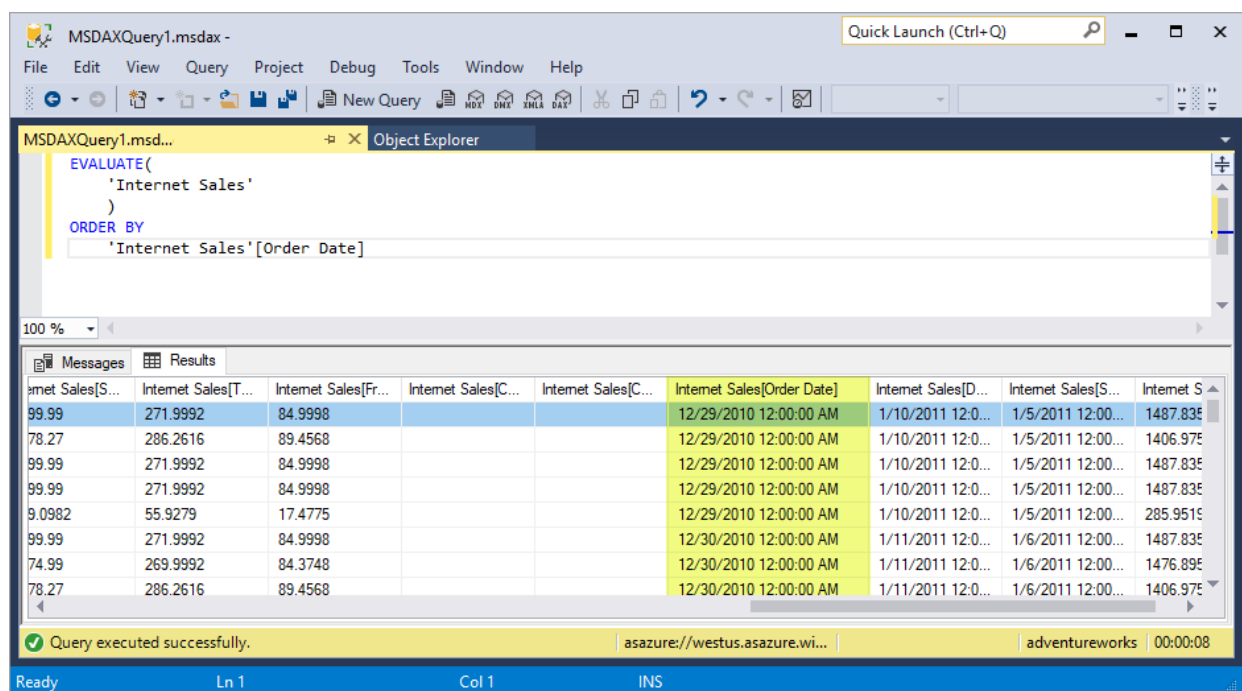
### Argumentos

TERMO	DEFINIÇÃO
expressão	Qualquer expressão DAX que retorne um único valor escalar.
ASC	(padrão) Ordem de classificação crescente.
DESC	Ordem de classificação decrescente.

### Exemplo

```
EVALUATE(
 'Internet Sales'
)
ORDER BY
 'Internet Sales'[Order Date]
```

Retorna todas as linhas e colunas da tabela Vendas pela Internet ordenadas por Data do Pedido, como uma tabela.



## START AT (opcional)

A palavra-chave opcional **START AT** é usada dentro de uma cláusula **ORDER BY**. Ela define o valor no qual os resultados da consulta começam.

### Sintaxe

```
EVALUATE <table>
[ORDER BY <{expression> [{ASC | DESC}]][, ...]
[START AT <{value}<parameter> [, ...]]]
```

### Argumentos

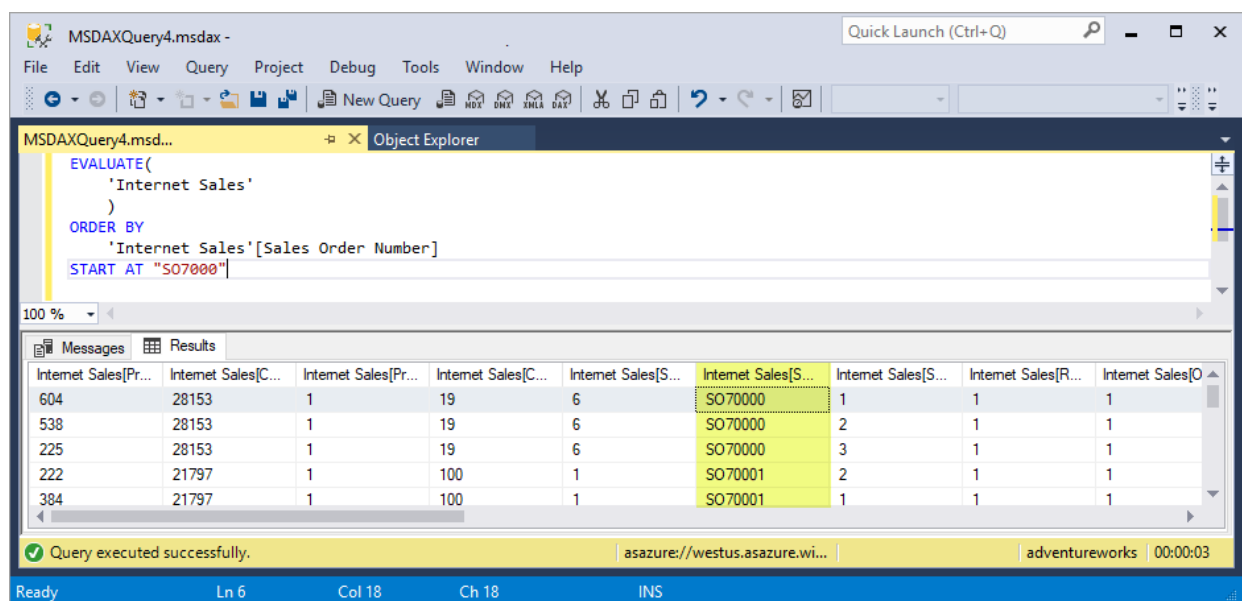
TERMO	DEFINIÇÃO
valor	Um valor constante. Não pode ser uma expressão.
parâmetro	O nome de um parâmetro em uma instrução XMLA prefixada com um caractere @ .

Os argumentos **START AT** têm uma correspondência de um para um com as colunas na cláusula **ORDER BY**. Pode haver tantos argumentos na cláusula **START AT** quanto na cláusula **ORDER BY**, mas não mais. O primeiro argumento no início em define o valor inicial na coluna 1 das colunas **ORDER BY**. O segundo argumento no início define o valor inicial na coluna 2 das colunas **ORDER BY** dentro das linhas que atendem ao primeiro valor para a coluna 1.

### Exemplo

```
EVALUATE(
 'Internet Sales'
)
ORDER BY
 'Internet Sales'[Sales Order Number]
START AT "SO7000"
```

Retorna todas as linhas e colunas da tabela de vendas pela Internet ordenadas por Número de Pedidos de Vendas, começando em SO7000.



Várias cláusulas **EVALUATE**/**ORDER BY**/**START AT** podem ser especificadas nesta consulta.

### DEFINE (opcional)

A palavra-chave **DEFINE** opcional define entidades que existem apenas durante a consulta. As definições são válidas para todas as instruções **EVALUATE**. As entidades podem ser variáveis, medidas, tabelas e colunas. As definições podem fazer referência a outras definições que aparecem antes ou depois da definição atual. As definições normalmente precedem a instrução **EVALUATE**.

#### Sintaxe

```
[DEFINE { MEASURE <tableName>[<name>] = <expression> }
 { VAR <name> = <expression>}]
EVALUATE <table>
```

#### Argumentos

TERMO	DEFINIÇÃO
tableName	O nome de uma tabela existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.
Nome	O nome de uma nova medida. Não pode ser uma expressão.
expressão	Qualquer expressão DAX que retorne um único valor escalar. A expressão pode usar qualquer uma das medidas definidas. A expressão deve retornar uma tabela. Se um valor escalar for necessário, encapsule o escalar dentro de uma função ROW() para produzir uma tabela.
VAR	Uma expressão opcional como uma variável nomeada. Um <b>VAR</b> pode ser passado como um argumento para outras expressões.

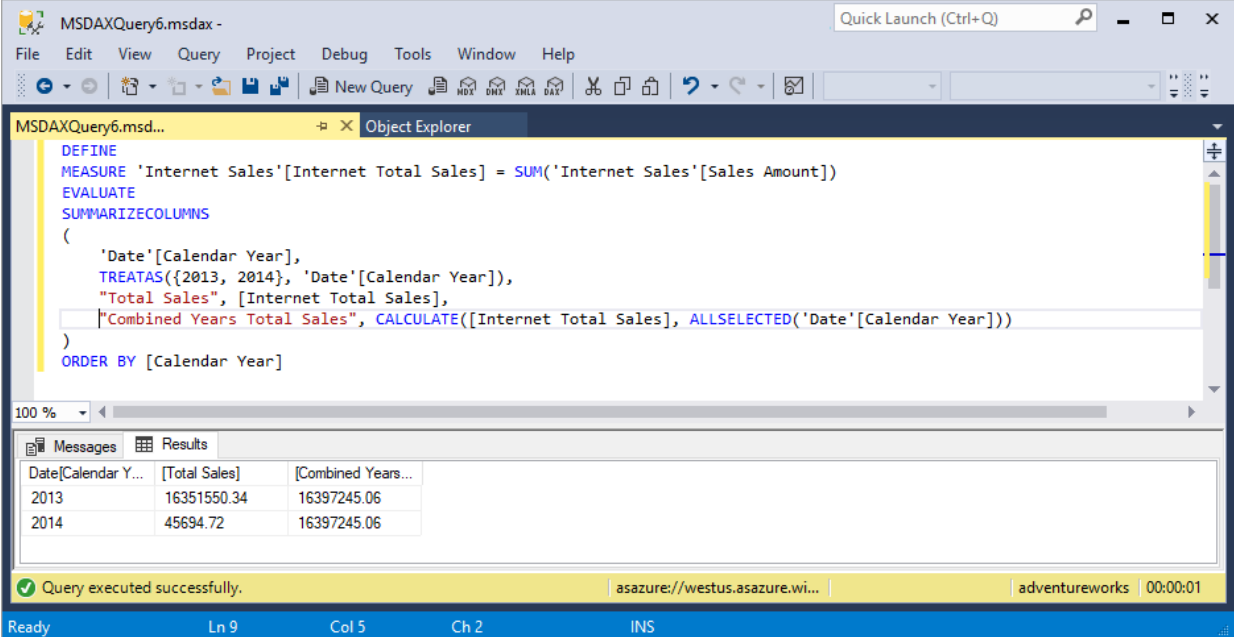
#### Exemplo

```

DEFINE
MEASURE 'Internet Sales'[Internet Total Sales] = SUM('Internet Sales'[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
 'Date'[Calendar Year],
 TREATAS({2013, 2014}, 'Date'[Calendar Year]),
 "Total Sales", [Internet Total Sales],
 "Combined Years Total Sales", CALCULATE([Internet Total Sales], ALLSELECTED('Date'[Calendar Year]))
)
ORDER BY [Calendar Year]

```

Retorna o total de vendas calculadas para os anos 2013 e 2014 e o total de vendas calculadas combinadas para os anos 2013 e 2014, como uma tabela. A medida na instrução DEFINE, Total de Vendas pela Internet, é usada nas expressões Total de Vendas e Total de Vendas de Anos Combinados.



The screenshot shows the SSDT interface with a DAX query in the query editor. The query is as follows:

```

DEFINE
MEASURE 'Internet Sales'[Internet Total Sales] = SUM('Internet Sales'[Sales Amount])
EVALUATE
SUMMARIZECOLUMNS
(
 'Date'[Calendar Year],
 TREATAS({2013, 2014}, 'Date'[Calendar Year]),
 "Total Sales", [Internet Total Sales],
 "Combined Years Total Sales", CALCULATE([Internet Total Sales], ALLSELECTED('Date'[Calendar Year]))
)
ORDER BY [Calendar Year]

```

The Results pane displays the following table:

Date[Calendar Y...	Total Sales	Combined Years...
2013	16351550.34	16397245.06
2014	45694.72	16397245.06

The status bar at the bottom indicates "Query executed successfully." and "adventureworks | 00:00:01".

## Parâmetros em consultas DAX

Uma instrução de consulta DAX bem definida pode ser parametrizada e usada repetidamente apenas com alterações nos valores de parâmetro.

O método [Execute Method \(XMLA\)](#) tem um elemento de coleção [Elemento de Parâmetros \(XMLA\)](#) que permite definir parâmetros e atribuir um valor. Na coleção, cada elemento do [Elemento de Parâmetro \(XMLA\)](#) define o nome do parâmetro e um valor para ele.

Faça referência a parâmetros XMLA prefixando o nome do parâmetro com um caractere `@`. Portanto, em qualquer lugar na sintaxe em que um valor é permitido, ele pode ser substituído por uma chamada de parâmetro. Todos os parâmetros XMLA são digitados como texto.

### IMPORTANT

Os parâmetros definidos na seção de parâmetros e não usados no elemento `<STATEMENT>` geram uma resposta de erro no XMLA.

### IMPORTANT

Os parâmetros usados e não definidos no elemento `<Parameters>` geram uma resposta de erro no XMLA.

# Consulte também

- [FILTER](#)
- [SUMMARIZECOLUMNS](#)
- [TREATAS](#)
- [VAR](#)



# Convenções de nomenclatura de parâmetro DAX

08/05/2020 • 3 minutes to read

Os nomes de parâmetro são padronizados na referência DAX para facilitar o uso e a compreensão das funções.

## Nomes do parâmetro

PARÂMETRO	DESCRIÇÃO
expressão	Qualquer expressão DAX que retorna um único valor escalar, em que a expressão deve ser avaliada várias vezes (para cada linha/contexto).
valor	Qualquer expressão DAX que retorna um único valor escalar em que a expressão deve ser avaliada exatamente uma vez antes de todas as outras operações.
tabela	Qualquer expressão DAX que retorna uma tabela de dados.
tableName	O nome de uma tabela existente, usando a sintaxe DAX padrão. Não pode ser uma expressão.
columnName	O nome de uma coluna existente usando a sintaxe DAX padrão, geralmente totalmente qualificada. Não pode ser uma expressão.
Nome	Uma constante de cadeia de caracteres que será usada para fornecer o nome de um novo objeto.
order	Uma enumeração usada para determinar a ordem de classificação.
empates	Uma enumeração usada para determinar a manipulação de valores de ligação.
type	Uma enumeração usada para determinar o tipo de dados para PathItem e PathItemReverse.

### Como prefixar nomes de parâmetro ou usar apenas o prefixo

PARÂMETRO	DESCRIÇÃO
como prefixar	<p>Os nomes de parâmetro podem ser mais qualificados com um prefixo descritivo de como o argumento é usado e para evitar a leitura ambígua dos parâmetros. Por exemplo:</p> <p>Result_ColumnName – refere-se a uma coluna existente usada para obter os valores de resultado na função LOOKUPVALUE().</p> <p>Search_ColumnName – refere-se a uma coluna existente usada para pesquisar um valor de resultado na função LOOKUPVALUE().</p>

PARÂMETRO	DESCRIÇÃO
como omitir	<p>Os nomes de parâmetro serão omitidos se o prefixo for claro o suficiente para descrever o parâmetro.</p> <p>Por exemplo, em vez de ter a seguinte sintaxe DATE (Year_Value, Month_Value Day_Value), é mais claro para o usuário ler DATE (Year, Month, Day). Repetir três vezes o valor do sufixo não adiciona nada a uma compreensão melhor da função e complica a leitura desnecessariamente.</p> <p>No entanto, se o parâmetro prefixado for Year_columnName, o nome do parâmetro e o prefixo permanecerão para garantir que o usuário entenda que o parâmetro requer uma referência a uma coluna existente de anos.</p>